Lexical Selection for Machine Translation

A thesis submitted to the University of Manchester for the degree of

Doctor of Philosophy

in the Faculty of Engineering and Physical Sciences

2011

Yasser Muhammad Naguib Sabtan

School of Computer Science

Contents

Word Count: 94,065

List of Figures

List of Tables

Abstract

Current research in Natural Language Processing (NLP) tends to exploit corpus resources as a way of overcoming the problem of knowledge acquisition. Statistical analysis of corpora can reveal trends and probabilities of occurrence, which have proved to be helpful in various ways. Machine Translation (MT) is no exception to this trend. Many MT researchers have attempted to extract knowledge from parallel bilingual corpora.

The MT problem is generally decomposed into two sub-problems: lexical selection and reordering of the selected words. This research addresses the problem of lexical selection of open-class lexical items in the framework of MT. The work reported in this thesis investigates different methodologies to handle this problem, using a corpus-based approach. The current framework can be applied to any language pair, but we focus on Arabic and English. This is because Arabic words are hugely ambiguous and thus pose a challenge for the current task of lexical selection. We use a challenging Arabic-English parallel corpus, containing many long passages with no punctuation marks to denote sentence boundaries. This points to the robustness of the adopted approach. In our attempt to extract lexical equivalents from the parallel corpus we focus on the co-occurrence relations between words.

The current framework adopts a lexicon-free approach towards the selection of lexical equivalents. This has the double advantage of investigating the effectiveness of different techniques without being distracted by the properties of the lexicon and at the same time saving much time and effort, since constructing a lexicon is time-consuming and labour-intensive. Thus, we use as little, if any, hand-coded information as possible. The accuracy score could be improved by adding hand-coded information. The point of the work reported here is to see how well one can do without any such manual intervention.

With this goal in mind, we carry out a number of preprocessing steps in our framework. First, we build a lexicon-free Part-of-Speech (POS) tagger for Arabic. This POS tagger uses a combination of rule-based, transformation-based learning (TBL) and probabilistic techniques. Similarly, we use a lexicon-free POS tagger for English. We use the two POS taggers to tag the bi-texts. Second, we develop lexicon-free shallow parsers for Arabic and English. The two parsers are then used to label the parallel corpus with dependency relations (DRs) for some critical constructions. Third, we develop stemmers for Arabic and English, adopting the same knowledge - free approach. These preprocessing steps pave the way for the main system (or proposer) whose task is to extract translational equivalents from the parallel corpus.

The framework starts with automatically extracting a bilingual lexicon using unsupervised statistical techniques which exploit the notion of co-occurrence patterns in the parallel corpus. We then choose the target word that has the highest frequency of occurrence from among a number of translational candidates in the extracted lexicon in order to aid the selection of the contextually correct translational equivalent. These experiments are carried out on either raw or POS-tagged texts. Having labelled the bi-texts with DRs, we use them to extract a number of translation seeds to start a number of bootstrapping techniques to improve the proposer. These seeds are used as anchor points to resegment the parallel corpus and start the selection process once again. The final F-score for the selection process is 0.701. We have also written an algorithm for detecting ambiguous words in a translation lexicon and obtained a precision score of 0.89.

Declaration

I hereby declare that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the “Copyright”) and he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the John Rylands University Library of Manchester. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

iii. The ownership of any patents, designs, trade marks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and exploitation of this thesis, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available from the Head of School of Computer Science (or the Vice-President).

Publication Based on the Thesis

Ramsay, A., and Sabtan, Y. (2009). Bootstrapping a Lexicon-Free Tagger for Arabic. In *Proceedings of the 9th Conference on Language Engineering ESOLEC'2009*, 23-24 December 2009, Cairo, Egypt, pp. 202-215.

Acknowledgments

The road from theoretical linguistics to computational linguistics is long and thorny. It would have been even more difficult and frustrating without the advice and support of my supervisor Prof. Allan Ramsay. I would like to express my deep gratitude and appreciation to him for his priceless support, encouragement, and guidance during this research. Actually, I will miss his weekly meetings when I learned from him new approaches and theories about computational linguistics. I am deeply indebted to him for his patience, numerous helpful discussions, invaluable suggestions, and comments on my work.

I wish also to extend my thanks to my fellow Ph.D. students Maytham Alabbas and Majda Al-Liabi for support, exchanging ideas and fruitful discussions on Arabic Natural Language Processing.

My deepest gratitude and appreciation are due to my wife and children for their help and support in difficult times. Your love, understanding and patience sustained me through to the end of my Ph.D.

I cannot thank my parents enough for their love and sacrifice throughout my whole life. Over the years they have continued to support and encourage me. I would like to thank them for everything.

Last but not least, I would like to thank my beloved country “Egypt” for its sponsorship to do this research and for giving me this opportunity.

Transliteration Table[1]

Arabic letter	Trans.	Name of letter	Arabic letter	Trans.	Name of letter
ء	'	HAMZA	ظ	Z	ZAH
آ	|	ALEF WITH MADDA ABOVE	ع	E	AIN
أ	O	ALEF WITH HAMZA ABOVE	غ	g	GHAIN
ؤ	W	WAW WITH HAMZA ABOVE	ـ	_	ARABIC TATWEEL
إ	I	ALEF WITH HAMZA BELOW	ف	f	FEH
ئ	}	YEH WITH HAMZA ABOVE	ق	q	QAF
ا	A	ALEF	ك	k	KAF
ب	b	BEH	ل	l	LAM
ة	p	TEH MARBUTA	م	m	MEEM
ت	t	TEH	ن	n	NOON
ث	v	THEH	هـ	h	HEH
ج	j	JEEM	و	w	WAW
ح	H	HAH	ي	y	YEH
خ	x	KHAH	ى	Y	ALEF MAQSURA
د	d	DAL	ـً	F	ARABIC FATHATAN
ذ	*	THAL	ـٌ	N	ARABIC DAMMATAN
ر	r	REH	ـٍ	K	ARABIC KASRATAN
ز	z	ZAIN	ـَ	a	ARABIC FATHA
س	s	SEEN	ـُ	u	ARABIC DAMMA
ش	$	SHEEN	ـِ	i	ARABIC KASRA
ص	S	SAD	ـّ	~	ARABIC SHADDA
ض	D	DAD	ـْ	o	ARABIC SUKUN
ط	T	TAH			

[1] We use the standard Buckwalter transliteration for converting Arabic script to the Roman alphabet. The transliteration scheme is available at: http://www.qamus.org/transliteration.htm

List of Abbreviations and Acronyms

Abbreviation	Full Form
1	first person
2	second person
3	third person
acc	accusative
ADJP	adjectival phrase
ADVP	adverbial phrase
ATB	Penn Arabic Treebank
AV	Arabic verses
BNC	British National Corpus
CA	Classical Arabic
CATiB	Columbia Arabic Treebank
CAV	canonical Arabic verses
CBMT	Corpus-based Machine Translation
CEV	canonical English verses
CFG	Context-Free Grammar
CL	Computational Linguistics
COG ACC	cognate accusative
COMPS	complement sentence
CONJ	Conjunction
CWS	Confidence-Weighted Score
def	definite
DG	Dependency Grammar
DOBJ	direct object
DR	dependency relation
DT	dependency tree
EBMT	Example-based Machine Translation
EV	English verses
fem	feminine
freq	frequency
fut	future
gen	genitive
GPSG	Generalized Phrase Structure Grammar
GR	grammatical relation
HMM	Hidden Markov Model
HPSG	Head-driven Phrase Structure Grammar
IDOBJ	indirect object
indef	indefinite
IR	Information Retrieval
LFG	Lexical Functional Grammar
masc	masculine
MLE	Maximum Likelihood Estimation

MOD	modifier
MRBD	Machine-Readable Bilingual Dictionary
MRD	Machine-Readable Dictionary
MSA	Modern Standard Arabic
MT	Machine Translation
MTT	Meaning-Text Theory
MWE	Multi-word expression
neg	negative
NLP	Natural Language Processing
nom	nominative
NP	noun phrase
OBJ	object
OVS	object-verb-subject
PADT	Prague Arabic Dependency Treebank
PASS SUBJ	passive subject
pl	plural
POBJ	object of preposition
POS	part-of-speech
POSS	possessive
PP	prepositional phrase
PRED	predicate
PREP	preposition
pres	present
PSG	Phrase Structure Grammar
PST	phrase structure tree
QADT	The Qur’anic Arabic dependency Treebank
RBMT	Rule-based Machine Translation
RE	regular expression
sing	singular
SL	source language
SMT	Statistical Machine Translation
ST	source text
SUBJ	subject
SVM	Support Vector Machine
SVO	subject-verb-object
TBL	Transformation-Based Learning
TL	target language
TM	Translation memory
TT	target text
VCOP	copula verb
VOS	verb-object-subject
VP	verb phrase
VSO	verb-subject-object
WG	Word Grammar

Chapter 1

Introduction

1.1 Motivation

In recent years there has been a growing interest in exploiting corpus resources for different Natural Language Processing (NLP) tasks. Machine translation (MT), which is defined as the automatic translation of text or speech from a source language (SL) to a target language (TL), is no exception to this trend. Corpora, which are collections of machine-readable texts, are increasingly recognized as an important resource for both teaching and research. Statistical analysis of corpora has proved to be extremely useful in identifying the properties of texts under analysis (Farghaly, 2003).

The increasing number of available bilingual corpora has encouraged research towards corpus-based MT. This move from the older rule-based approach to the new corpus-based approach was motivated by the desire to overcome the ‘knowledge acquisition bottleneck’ which characterizes rule-based MT systems. Such systems require the development of large-scale hand-written rules, which is expensive, labour-intensive and time-consuming. Corpus-based systems, in contrast, are generally more robust than rule-based ones, since they require fewer, if any, linguistic resources and thus are cheaper, easier and quicker to build. Consequently, the MT research community has started to focus on corpus-based rather than rule-based techniques. Additionally, there is an increasing tendency to employ hybrid approaches in building MT systems. Such hybrid approaches attempt to select the best techniques from both rule-based and corpus-based paradigms. This combination of the positive elements of both paradigms has clear advantages: “a combined model has the potential to be highly accurate, robust, cost-effective to build and adaptable” (Hearne, 2005).

It is generally claimed that the MT problem is decomposed into two sub-problems: lexical (or word) selection problem and word reordering problem of the selected words (Lee, 2002; Bangalore et al., 2007). Lexical selection in an MT system is a process that selects an appropriate target word or words which carry the same meaning as the corresponding word in the source text (Wu and Palmer, 1994; Lee et al., 1999; 2003). Word reordering, in contrast, is concerned with rearranging the selected TL words to produce a well-formed TL sentence.

The current research is oriented towards handling the first sub-problem, i.e. the word selection. Handling the second sub-problem, i.e. word reordering, is outside the scope of this research. Solving the word selection problem is a very important step in performing high quality MT, since the quality of translation varies substantially according to the results of translation selection. Yet, it is very difficult to solve the lexical selection problem because of its sensitivity to the local syntax and semantics.

Like other MT problems, knowledge acquisition is crucial for lexical selection. Thus, many researchers have attempted to extract knowledge from existing resources. For instance, corpus-based approaches select a target word using statistical information that is automatically extracted from corpora (Lee et al., 1999). Some of such corpus-based approaches use a bilingual corpus as a knowledge source to extract statistical information (Brown et al., 1990). Such paired corpora provide evidence that a lexicon could be extracted from alignment of texts one of which is a translation of the other (Boguraev and Pustejovsky, 1996).

We see our approach to solving lexical selection problem as closely aligned with corpus-based MT approaches, and as a separate component that could be incorporated into existing systems. Notably, the approach we adopt in this study can be applied to any language pair, but we focus our experiments on Arabic and English for the following reasons.

(i) Since Arabic and English belong to two unrelated families, MT is bound to face many problems in producing meaningful coherent translations between these languages (Izwaini, 2006). Such problems occur on a number of linguistic levels, i.e. lexical, structural, semantic and pragmatic. We are concerned in this study with the lexical level.

(ii) Furthermore, Arabic words are hugely ambiguous due to the lack of short vowels and other diacritic marks, as will be shown below, and thus pose a challenge for the current lexical selection task.

We use a parallel Arabic-English corpus in our endeavour to extract lexical equivalents, paying attention to the co-occurrence relations between words. Statistical analysis of corpora can reveal relevant trends and probabilities of occurrences, which have proved to be helpful in natural language analysis (Allen, 1995; Charniak, 1993). According to Ney (1997), "the principal goal of statistics is to learn from observations and make predictions about new observations." Thus, we make guesses when we wish to make a judgement but have incomplete information or uncertain knowledge.

Our statistical approach to lexical selection seeks to automatically learn lexical and structural preferences from corpora. We recognize that there is a lot of information in the relationships between words, i.e. which words tend to group with each other. These relations are investigated based on the context in which words occur. This context may be on the lexical level, which focuses on which words co-occur in a given sentence, or the structural level, which deals with the co-occurrence of words in a given syntactic relation. Consequently, we evaluate our approach on both raw and linguistically annotated texts. The linguistic information we use to improve the selection process is part-of-speech (POS) tags and dependency relations (DRs) for both Arabic and English.

We can say that the meaning of a linguistic unit is vitally affected by the environment in which it occurs, i.e. which units precede and follow it. Let us make this point clearer by giving the following examples.

1.1 I run races[2].
1.2 The run on the stock market continues.

In the first sentence the linguistic environment in which *run* occurs indicates that it is a verb. Similarly, in the second sentence the linguistic environment in which *run* appears indicates that it is operating as a noun. This is because it is not possible to have a noun phrase consisting of *the* alone. Moreover, it is well-known that nouns follow articles in English. This has important consequences for a probabilistic approach to language, because it means that the probabilities of occurring words are

[2] Throughout the thesis, English examples are written in regular form, whereas Arabic examples in transliteration are written in italic and the English gloss in double quotations. However, when only English words are mentioned inside paragraphs they are written in italic. Notably, the English translation of Qur'anic verses is written between square brackets [ ], because the verse may contain a quotation.

not independent, but that they affect one another. Thus, the probability of the word *run* being a verb may be 50%, but the probability of it being a verb following the definite article *the* may stand at 1% (McEnery, 1992).

1.2 About Arabic

Arabic is one of the most widely spoken languages in the world with over 300 million speakers. It is the official language of all the countries of northern Africa, the Arabian Peninsula and much of the Middle East. In addition, it is the religious language of all Muslims worldwide, regardless of their origin. There are a number of varieties that are spoken across the Arab countries. Two main varieties are widely used among the Arab nations and are understood by all Arabs. The first one is Classical Arabic (CA), which is the language of the Qur'an and prophetic traditions. This variety is used in education in religious schools and in religious sermons. The second variety is Modern Standard Arabic (MSA), which is the contemporary language that is used in newspapers, magazines, text books, academic books, novels and other writing (Parkinson, 1990). Besides these two main varieties, there are other varieties that are classified as colloquial language or dialects. These dialects differ from one country to another, where every country has its own vernacular. These dialects differ even inside one country from one part to another and from one context to another. In Egypt, for instance, different varieties of Arabic are used in different contexts. These varieties are best described by Badawi (1973) who lists five varieties or levels of Arabic on a descending scale, on top of which comes فصحى التراث *fuSoHaY AlturaAv* "Classical Arabic". It is followed by what he calls فصحى العصر *fuSoHaY AlEaSor* "Modern Standard Arabic". Next comes three consecutive levels of colloquial Arabic, viz. عامية المثقفين *EaAm~iy~ap Almuvaq~afiyn* "Colloquial of the Educated", عامية المتنورين *EaAm~iy~ap Almutanaw~iriyn* "Standard Colloquial" and finally عامية الأميين *EaAm~iy~ap AlOum~iyyin* "Colloquial of the Illiterate".

In fact, Arabic exhibits a true diglossic situation (Farghaly and Shaalan, 2009). Diglossia, according to Ferguson (1959), is a phenomenon whereby two or more varieties of the same language are used by a speech community. Each variety is used for a specific purpose and in a distinct situation. This diglossia is vivid in Arabic in the three varieties, where CA is the language of religion and is used by Arabic speakers in their daily prayers while MSA, the more recent variety of CA, is used by educated people in formal settings such as the media, the news, and the classroom. As for the regional dialects, they are used with family and friends. The current study has nothing to do with the colloquial varieties. We are mainly concerned with the first two varieties, namely CA and MSA. Our work is applied to CA with a view to be applied to MSA in the future. Notably, MSA is a simplified form of CA, and follows its grammar. The main differences between CA and MSA are that MSA has a larger (more modern) vocabulary, and does not use some of the more complicated structures of CA (Khoja, 2001a). The same view is expressed by Ryding (2005) who says that differences between CA and MSA are primarily in style and vocabulary. In terms of linguistic structure, CA and MSA are largely similar. Thus, we use an undiacritized version of a CA corpus to mimic the way MSA is written.

As pointed out above, CA is the language of the Qur’an and Sunna (prophetic traditions). CA is written with diacritic marks above the consonants. This was basically done to help people to read such Arabic texts perfectly. The modern form of Arabic (MSA), in contrast, is written without diacritics. This results in a great number of ambiguities, since a certain lemma in MSA can be interpreted in different ways. This represents a challenge for any NLP task (Maamouri et al., 2006). Figure (1.1) below shows an example for a surface form composed of only three letters but with seven different readings.

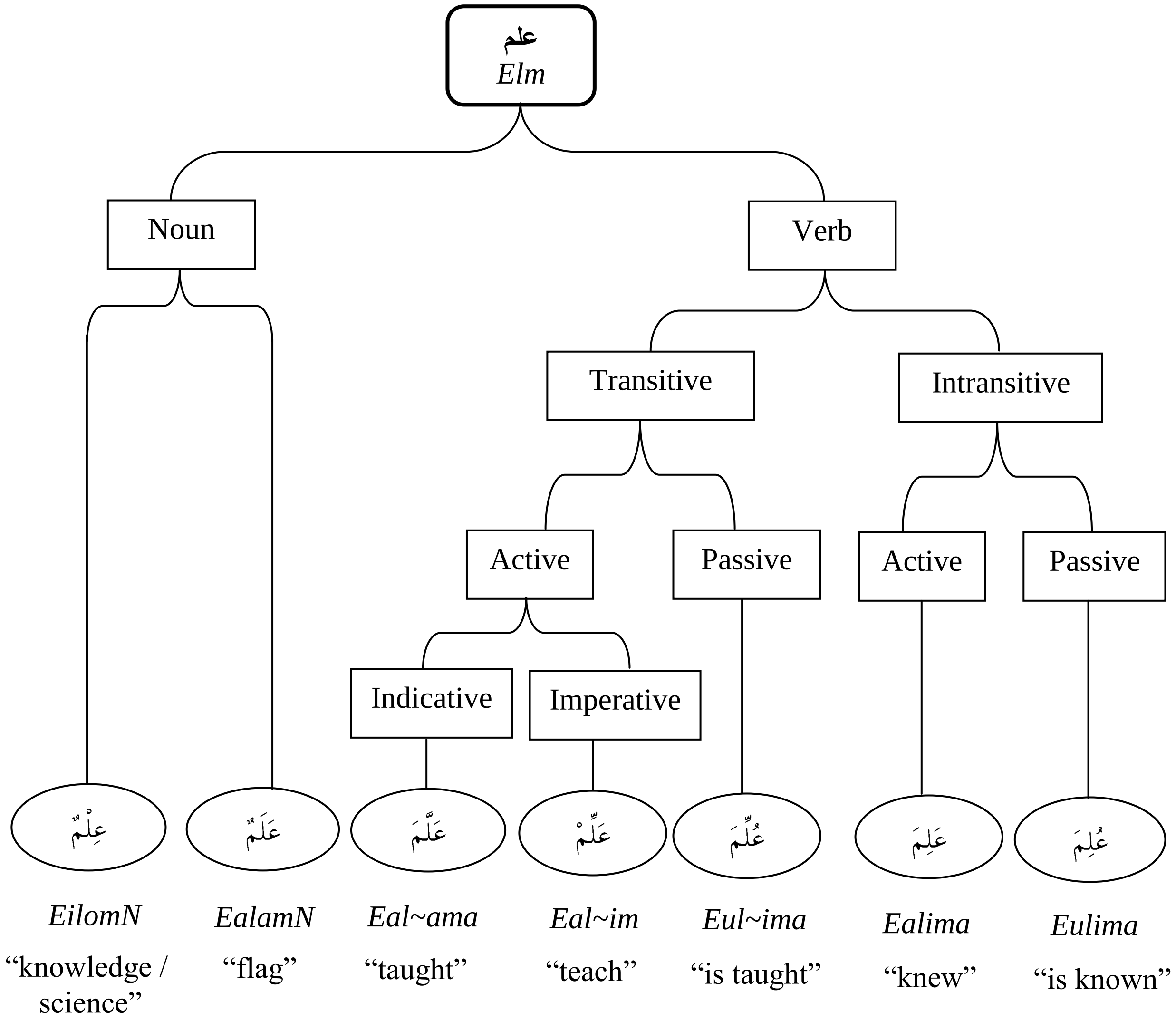


Figure 1.1 Ambiguity caused by the lack of diacritics

Due to this lack of diacritics in MSA, a single word can have different senses. Every sense is largely determined by the context in which the word is used. Habash and Rambow (2005) refer to this potential ambiguity caused by the missing short vowels and other diacritic marks in MSA as follows.

> "Arabic words are often ambiguous in their morphological analysis. This is due to Arabic's rich system of affixation and clitics and the omission of disambiguating short vowels and other orthographic diacritics in standard orthography."

Ali (2003) gives an example that can make an English speaker grasp the complexity caused by dropping Arabic diacritics. Let us suppose that vowels are dropped from an English word and the result is *sm*. The possibilities of the original word are: *some, same, sum, seem, seam* and *semi*. However, the situation is worse in MSA than in

English, since English can be sometimes understood without vowels as in the following example.

1.3 He snt me a txt msg

This lack of diacritization is problematic for computational systems(Nelken and Sieber, 2005). This is because the surface form of a word gives rise to a number of possible underlying forms, as shown in figure (1.1) above. Many efforts have been devoted to reconstruct the missing diacritics in MSA for developing a number of applications such as text-to-speech systems (Ramsay and Mansour, 2004; 2007).

As far as MT is concerned, this undiacritized form of the language poses many challenges in the field. Kübler and Mohamed (2008) and Mohamed and Kübler (2009) point out that this lack of diacritics causes problems for many tasks in Arabic NLP, including MT. To emphasize this point, they cite the example in 1.4 that is translated wrongly by Google.

1.4 اشتريت المسكن من الصيدلية

A$tryt Almskn mn AlSydlyp

I bought the home from the pharmacy (Google Translate)

I bought a painkiller from the pharmacy (correct translation)

As a matter of fact, this error in translation has occurred because the word-form مسكن *mskn* is a highly ambiguous word that has a number of meanings with several pronunciations. Thus, it can be pronounced as *maskan* “home”, *musak~in* “analgesic or painkiller”, *masakn* “they (fem.) have held”, or *musikn* “they (fem.) have been held”. Similarly, Al-Maskari and Sanderson (2006) reported that the term علم النفس *Elm Alnfs* “psychology” was wrongly translated by Systran as “flag of breath”. This is because the system translated each word individually. The MSA lexeme علم *Elm* has different interpretations when diacritics are added, as noted above. Therefore, the system has chosen the sense of “flag” and ignored the correct sense of “science”. The other lexeme النفس *Alnfs* can be interpreted as either *Alnafos* “soul” or *Alnafas* “breath”. However, the Arabic words علم النفس *Elm Alnfs* are used as a compound noun to mean “psychology”.

1.3 Research Aim

As pointed out above, the current study aims at choosing the TL word that most closely conveys the meaning of an SL word, adopting a corpus-based approach. The two languages concerned here are Arabic and English. We attempt to extract the translational equivalents from our parallel bilingual corpus. We are particularly interested in extracting information about co-occurrence patterns of lexical items from the SL (Arabic) corpus and using them for identifying equivalents in the TL (English) corpus.

We hold the view that the meaning of a lexical item is largely determined by its relations with other neighbouring items in a given context. Cruse (1986) refers to this notion that the meaning of a word is fully revealed in its contextual relations. We follow the school of syntax-driven lexical semantics, which is based on syntactic theory. The central role of this approach in the process of deriving the meaning of a text is to decode the nature of dependency relations between heads of phrases and their arguments in a particular language (Nirenburg and Levin, 1992). The following English example makes this point clearer.

1.5 John interviewed Max for a job.

In order to know that this English sentence means that John was considering hiring Max and not that Max was in the position to hire John, it is necessary to know that the interviewer role is expressed as the subject of the sentence, and that the interviewee role is expressed as the object. In addition, it should be known that the subject precedes the verb and that the object follows it (Nirenburg and Levin, ibid). According to MacDonald et al. (1994), this knowledge of words is termed by current syntactic theories as argument structure. They (ibid.) indicate that "the argument structures associated with a word encode the relationships between the word and the phrases that typically occur with it (the word's arguments)". In actual fact, this concept of argument structure is related to the earlier notion of verb subcategorization frames (as expounded by Chomsky, 1965), which refer to the kinds of syntactic phrases that optionally or obligatorily occur with a verb in a sentence. For example, the verb *put* must occur with both a direct object NP and a prepositional phrase (PP). In addition to this information, an argument structure representation actually provides some semantic information about the relationship between a word and each of its associated arguments (MacDonald et al., 1994). For a verb, for instance, the argument structure includes also its subject, which was typically excluded from its subcategorization frames. Thus, the argument structure for the verb *put* includes a subject NP (which takes the role of agent), an object NP (which takes the role of theme), and a PP (which takes the role of location).

Thus, in our research we will study syntax-based co-occurrence patterns, i.e. co-occurrences of words in certain syntactic relations (such as subject-verb, verb-object, adjective-noun, etc.). According to Dagan et al. (1991), these are also called lexical relations. The typical relations we exploit are those between verbs and their subjects, objects and modifying prepositional phrases. As a case in point, Rimon et al. (1991) indicate that the statistics obtained about such relations can help in solving the problem of target word selection. He gives the following example for translation from German into English.

1.6 Es wurde auch die Vorstellung begraben, man könne mit den Ideen und Ideologien des 19. Jahrhunderts die ganz anderen Probleme des 20. Jahrhunderts lösen.

This sentence contains three ambiguous words, namely *Vorstellung*, *begraben* and *lösen*. These words have a number of possible translations into English. Without having information which is the right translation for each of these words in this context, one would get alternative translations for the current sentence, such as:

But also the idea /picture /performance /presentation was abandoned / relinquished / buried / ended that one could solve / resolve / remove / cancel the totally different problems of the 20th Century with the ideas and ideologies of the 19th Century.

According to Rimon et al. (ibid.), the statistical data on the frequency of lexical relations in very large English corpora help in selecting automatically the correct translation for the three cases. The words *idea* and *abandon* were selected because they co-occurred in the 'verb-object' relation significantly more times than all other alternative combinations. Similarly, the verb *solve* was selected since it appeared frequently with the noun *problem* in the 'verb-object' relation. In this way, corpus-based studies make it possible to identify the meaning of words by looking at their occurrences in natural contexts, rather than relying on intuitions about how a word is used (Biber et al., 1998).

We adopt a lexicon-free approach to our task of selecting lexical equivalents. This has been done to achieve the following goals:

- To investigate the effectiveness of different techniques without being distracted by the properties of the lexicon.
- To make the overall work as purely automatic as possible, using as little, if any, hand-coded information as possible.

Concerning the first goal above, it is known that when a lexicon of words is used, it guides the NLP task in question, giving less opportunity for a real test of the employed algorithms and techniques. Thus, the lexicon has a major role to play in the entire process, such as the selection process that we aim for in this research. In other words, we are specifically interested in how effectively we can carry out this task without providing any information about particular lexical items, especially open-class items, since this will make it easier to see the contributions made by particular algorithms. Any practical system for carrying out this task will benefit from the presence of hand-coded information about specific words, but the provision of such information makes it harder to evaluate the effectiveness of more general principles. We have therefore deliberately avoided including a hand-coded lexicon. As for the second goal, the ultimate goal of most NLP systems is to make the computer carry out a given task in a completely automatic way. This idea of automatization has the great advantage of saving time and effort, since constructing a lexicon is time-consuming and labour-intensive. Furthermore, we avoid the need for a large training set of manually annotated data. We thus try to minimize the resources required to achieve our task. This will be made clear when we talk about each of the steps that we have taken to achieve our primary goal, i.e. lexical selection, in the following chapters.

Words in any natural language are normally subdivided into open-class and closed-class. Sometimes these two categories have different names such as content and function words or, according to Palmer (1981), full and form words respectively. Content words carry most of the lexical content in a sentence and are therefore called lexical words. Function words are essential to the grammatical structure of a sentence and are therefore called grammatical or structural words (Stubbs, 2002). In our lexicon-free approach towards lexical selection we deal only with the open-class words. Handling the closed-class words is outside the scope of this thesis.

With this in mind, we have carried out a number of steps that are described as follows:

(i) We have started with building a lexicon-free POS tagger for Arabic.

(ii) We have used a similarly lexicon-free POS tagger for English developed by Prof. Allan Ramsay.

(iii) We have written a lexicon-free shallow dependency parser for Arabic.

(iv) We have also used a lexicon-free shallow dependency parser for English.

(v) We have built a lexicon-free bilingual proposer to propose lexical equivalents.

(vi) Along with the proposer we have written a lexicon-free stemmer for Arabic and English.

(vii) We have applied bootstrapping techniques to the proposer.

(viii) We have automatically detected ambiguous words with the same POS tags in a given translation lexicon.

Most of the steps outlined above are preprocessing steps to be fed into the bilingual proposer. Thus, the taggers are used to POS tag the parallel corpus and then the proposer is applied to the tagged texts. Likewise, the parsers are used to produce the dependency relations (DRs) in the parallel corpus and then this output is fed into the proposer to suggest a number of ‘head-dependent’ pairs to bootstrap the selection process once again. As for the stemmers, they are used to get the canonical forms of similar word-forms in the parallel corpus. The details of these steps will be given in the following lines.

As for step (i), the Arabic POS tagger uses a combination of rule-based, machine learning and probabilistic techniques. As pointed out earlier, MSA is written without diacritics, which makes it hugely ambiguous. To achieve this task of tagging for MSA without using a lexicon is thus extremely hard. Therefore, we have opted for starting with a diacritized text. In addition, we should have a parallel corpus in order to achieve our main task of lexical selection. The available diacritized text that has a parallel English translation is the Qur’an. Consequently, we have used the Arabic text of the Qur’an and its English translation as our parallel corpus. It is worth noting that some Arabic researchers have used both the diacritized and undiacritized texts of the Qur’an as a testing ground for some NLP applications. Hammo et al.

(2007; 2008) is a case in point, where they used the vowelized and unvowelized texts of the Qur'an to test an Arabic information retrieval (IR) search engine. As far as the tagger is concerned, we use the Arabic diacritized text only in the early stages of training the tagger but we remove diacritics from it and apply all the subsequent steps on the undiacritized version of the corpus. This has been done with the belief that the adopted approach would extend to MSA if we had a diacritized parallel MSA corpus. MSA is, of course, generally written without diacritics. So, any parallel corpus is likely to be undiacritized. However, it is possible to automatically diacritize text with reasonable accuracy. It is unclear whether the accuracy of such artificial diacritization is good enough for our technique to work, but in principle it should be possible. This is because CA and MSA are morphologically, syntactically and semantically similar to a large extent, as MSA is a more recent variety of CA (Farghaly and Shaalan, 2009). The details of the used parallel corpus will be discussed in the coming chapter. As regards Arabic POS tagging, our approach to POS tagging can be summarized as follows:

(A) For the diacritized version of the Arabic corpus, we apply two subsequent types of tagging:

  (i) Rule-Based Tagging.

  (ii) Transformation-Based Learning (TBL) Tagging

(B) Then we remove diacritics from the corpus and keep the tags. We apply two subsequent types of tagging to this undiacritized corpus:

  (iii) Bayes + Maximum Likelihood Estimation (MLE) Tagging.

  (iv) TBL Tagging

This phase is the first step towards disambiguating the Arabic undiacritized lexical items. This point can be made clear through giving examples. The English word *book* can be used, among other uses, as a **noun** to mean "a written literary work" or as a **verb** to mean "reserve". When its POS tag is known to be a noun in a given context the other verb possibility is excluded and thus its meaning is disambiguated. Similarly, the Arabic undiacritized word كتب *ktb* can be used, among other uses, to mean either the verb "wrote" or the plural noun "books" according to the context in which it occurs. Another striking example that shows this lexical ambiguity is the Arabic word دخل *dxl* which may be a **noun** meaning *daxol* "income" or a **verb** meaning *daxala* "entered". Thus, if the POS tag is verb, then the other possibility of being a noun is excluded. In this way a lexical item is categorically disambiguated.

Categorical ambiguity, according to Hirst (1987), is a type of lexical ambiguity which includes also homonymous and polysemous ambiguity. A word is categorically ambiguous if it can be used in different syntactic categories. For example, the word *right* can be used as a noun, a verb, an adjective or an adverb. It goes without saying that resolving this type of lexical ambiguity constitutes the main challenge and the ultimate goal of a POS tagger (Alqrainy et al., 2008). It follows then that tagging text with parts of speech is very useful for machine translation, since a word in one language could mean two or more different words in another language depending on the word's grammatical category, i.e. POS tag. For example, the Arabic word حملته *Hmlth* could be either a verb meaning "she carried him" or a noun meaning "his campaign" (Khoja, 2003). With respect to step (ii) above, we only use the developed lexicon-free tagger for English. We have not contributed to the English tagger. So, we will describe only the used tagset when we describe POS tagging in chapter 4.

As for steps (iii) and (iv), we have written the shallow dependency parsers in Arabic and English using regular expressions (REs). The advantage of using REs for this task is that they can be applied extremely quickly. Both parsers output dependency relations (DRs) for certain lexical categories. According to Ide and Véronis (1998), researchers have recently avoided complex processing by using shallow or partial parsing. For example, in her approach towards disambiguation of nouns, Hearst (1991) segments text into simple noun and prepositional phrases and verb groups, and discards all other syntactic information. This phase of partial parsing has a role to play in disambiguating lexical items. According to Reifler (1955), grammatical structure can help disambiguate lexical items. For example, the word *keep* can be disambiguated by determining whether its object is gerund, adjectival phrase or noun phrase, as in the following three sentences respectively.

1.7 He kept eating.

1.8 He kept calm.

1.9 He kept a record.

We focus on certain syntactic relations in our implementation of the dependency parsers. This will be illustrated when we discuss both parsers in chapter 5.

With regard to step (v), the bilingual proposer we have built relies on the statistics of co-occurrences of lexical items in the parallel corpus to extract the translational equivalents.

We apply the proposer on raw texts as well as linguistically annotated texts. The annotated texts are either POS-tagged or DR-labelled. Hence, the three different types of texts are classified as follows:

(i) Raw Texts.

(ii) POS-Tagged Texts.

(iii) Texts with DRs.

Being generally applied to two types of text, i.e. raw texts and annotated texts, the proposer's approach to lexical selection exploits, as indicated by Ide and Véronis (1998), two types of context.

- The **bag of words** approach: where context is considered as SL words in parallel with TL words on the same structural level, i.e. on the verse[3] level in our parallel corpus. This way of context is made use of in testing the proposer on raw texts in the parallel corpus.
- **Relational information**: here context is considered in terms of both POS tags and syntactic relations between SL words and corresponding TL words. This way of context is used when testing the proposer on POS-tagged as well as DR-labelled texts in the parallel corpus.

It is worth mentioning that Ide and Véronis (ibid.) have pointed out that these two types of context are exploited in terms of word sense disambiguation. So, both types of context are used with respect to the target word that needs to be disambiguated. The relational information may include other types, such as selectional preferences, phrasal collocation, semantic categories, etc. But we draw on the two types of context in our selection process and so we have adapted the way they are used to suit our purpose. Step (vi) above refers to the fact that we have written a stemmer for Arabic and English. This has been done to test the proposer on both stemmed and unstemmed texts and compare the results we obtain in these tests.

The seventh step above is concerned with the use of bootstrapping techniques to improve the proposer. Having labelled the parallel corpus with the DRs, we extracted

[3] Our parallel corpus is composed of verses. A Qur'anic verse is one of the numbered subdivisions of a chapter in the Qur'an. A verse may contain one sentence or more, but we will use the terms verse and sentence interchangeably.

a number of dependency pairs, i.e. equivalents consisting of 'head-dependent' pairs (the dependent in such a pair may be an argument or a modifier). Then we filter these pairs to obtain a number of one-word translation pairs which we call 'seeds'. We have used these trusted seeds to resegment the parallel corpus after removing them from the corpus. This, in turn, assists in realigning the verses after shortening them and filtering out some of the wrong translational candidates.

The final step refers to writing an algorithm for automatically detecting ambiguous words where each sense has the same POS tag. This contrasts with cases where the different senses have different tags, since these will be disambiguated by the POS tagger. The problem cases are words with the same POS category which have different interpretations. Those words, which are basically polysemes, homonyms and homographs, are translated differently according to the context in which they are used. We have tried to disambiguate these words automatically in the corpus. But due to time constraints, we managed only to detect them automatically and will pursue the way to handle them automatically in future work.

Generally speaking, our approach to lexical selection comprises two phases. The first phase deals with learning bilingual equivalents, and the second phase is concerned with applying the approach to actual text. The system's architecture for lexical selection in the two stages can be illustrated in figures (1.2) and (1.3) respectively.

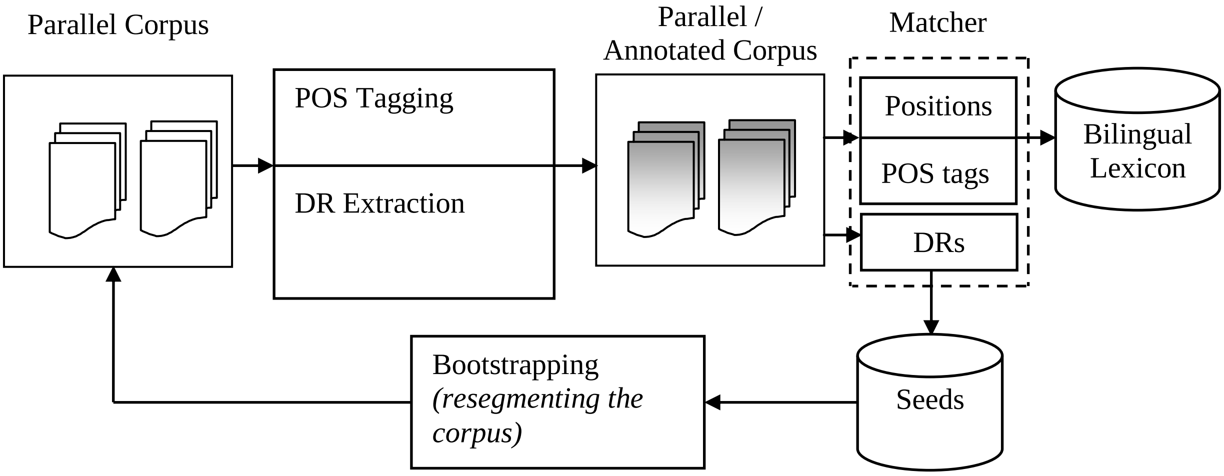


Figure 1.2: The system's architecture for learning bilingual equivalents

As for the second phase of application, it is shown in the following figure.

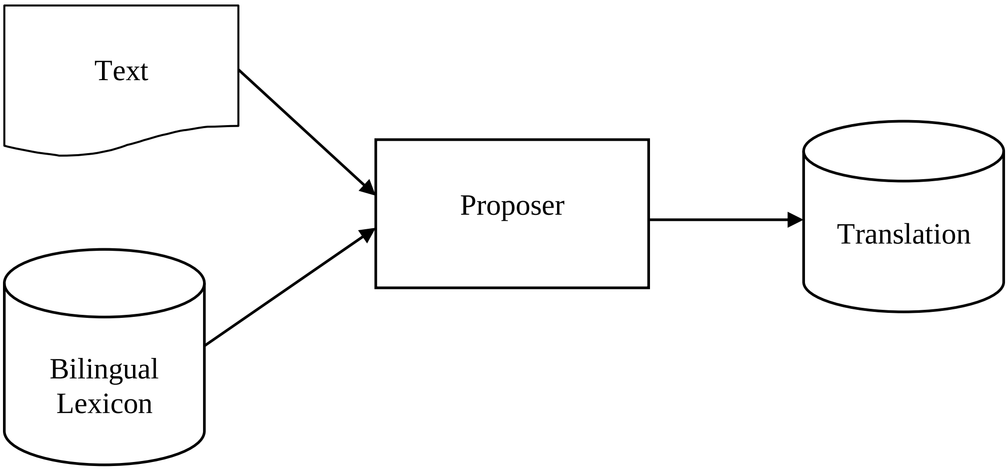


Figure 1.3: The system's architecture for the application phase

The Qur'anic corpus that we use has specific characteristics which make the current task of lexical selection more difficult. This, consequently, emphasizes the robustness of the adopted approach, since applying our approach to a challenging type of text means that it is expected to do better if applied to a less challenging text. The description of the corpus along with the Qur'anic linguistic features that make the project underway more challenging will be presented in the following chapter. We wrap up this introductory chapter by giving an outline for the structure of the thesis as a whole in the following section.

1.4 Thesis Structure

In this introduction we have presented the research problem and our approach towards achieving the goal of the current research. Our approach can be applied to any language pair and any direction, but we use the Arabic-English pair for our investigation. Accordingly, we have reviewed the different varieties of Arabic, shedding light on the inherent problem of ambiguity and how it is hugely pervasive in undiacritized Arabic. This, consequently, poses a challenge for the lexical selection task. We have explained that we adopt a lexicon-free approach, using statistical information that is automatically extracted from corpora. We have clarified the reasons for deliberately choosing not to construct a lexicon. We have also pointed out that we use very little, if any, manual intervention for training all our classifiers.

Thus, we provide very little hand-coded information for the Arabic POS tagger. For English we only use a similarly built POS tagger. The same approach is also applied to the Arabic and English dependency parsers as well as stemmers. Finally the proposer, which is the main tool for lexical selection, is also built on data-driven methods, without using any hand-coded information.

Using Arabic and English as a language pair for application, we discuss our parallel Arabic-English corpus in chapter 2. We illustrate the rationale behind choosing the Qur'anic source text and the English translation as our corpus of analysis. Then we outline some of the distinctive properties that characterize our corpus and how far this can point to the robustness of the adopted approach.

Chapter 3 gives an overview of MT, illustrating the different strategies that are used in the field of MT, i.e. direct, interlingua and transfer. In addition, the various approaches that are adopted towards solving the MT problem are discussed in detail. These approaches are generally classified as rule-based and corpus-based (or data-driven). We end the chapter by presenting the state of the art in lexical selection for MT.

In order to achieve the current goal of lexical selection for MT, we carry out some preprocessing steps before executing the selection process. These preprocessing steps consist in (i) POS tagging the Arabic and English texts, as explained in chapter 4. As pointed out earlier in this chapter, undiacritized Arabic is hugely ambiguous. So, POS tagging texts removes part of the inherent ambiguity in lexical items. This type of ambiguity, sometimes called categorical ambiguity, permeates lexical as well as structural levels. In (ii) we label both texts with DRs, as shown in chapter 5. In fact, labelling bi-texts with DRs between words is carried out to extract a number of 'head-dependent' translational pairs to be used as anchor points to bootstrap the selection process. Thirdly, in (iii) we stem the parallel corpus, aiming mainly for clustering semantically related words and assigning one stem for all of them, as illustrated in chapter 6.

Thus, in chapter 4 we discuss the problem of POS tagging for natural texts. We start with discussing Arabic morphology, throwing light on Arabic grammatical parts of speech. We also pinpoint Arabic word structure and the non-concatenative nature of Arabic morphology which is based on the root and pattern notion. Then we review the different approaches to POS tagging in general and Arabic POS tagging in particular. We also discuss the different challenges for Arabic POS tagging and review the state of the art as far as Arabic POS taggers are concerned. Then we describe the lexicon-free POS tagger that we have built for Arabic and evaluate its different stages. We use this tagger to tag the Arabic text in the parallel corpus. We conclude with presenting the English tagger that we use in our work and the tagset used to tag the English text of the parallel corpus.

Chapter 5 investigates the DRs in Arabic and English. Firstly, we give a descriptive analysis of the main sentence structure in Arabic and the related issues of agreement and word order. Then we explore the main approaches to syntactic analysis, i.e. phrase structure grammar (PSG) and dependency grammar (DG), so as to compare between them. We give a brief account of PSG and elaborate on the theoretical framework of DG, on which our framework is based. We also discuss the implementation of dependency parsing for Arabic as a free word order language. Then we describe a lexicon-free shallow dependency parser for Arabic and the DRs that we use to parse the Arabic corpus. We conclude with discussing a similarly lexicon-free English shallow parser and the DRs that are used.

In chapter 6 we discuss the main tool for selecting translation equivalents, namely the proposer. We start with discussing the way we normalize the parallel texts as well as data preparation. Then we describe our general proposed method for learning bilingual equivalents through lexicon building and then applying the approach to actual text to do lexical selection. In this chapter we also present the Arabic and English stemmers. In this section we show our approach to stemming, which focuses primarily on grouping semantically related words as a way to guide the proposer. We then use the stemmer to stem the parallel corpus. Thus, we have two versions of the corpus, i.e. stemmed and unstemmed versions. Afterwards, we start to apply the general proposer method on the parallel corpus in its raw nature, whether stemmed or unstemmed, and evaluate the results. We use the same method on tagged texts, also both stemmed and unstemmed, but with some modifications to suit the tagged corpus. We then move on to use the same method on the dependency-labelled version of the parallel corpus. This allows us to extract a number of seeds, i.e. Arabic-English pairs. We evaluate the accuracy of such seeds. We then use such seeds as a means towards bootstrapping techniques, where we resegment the parallel corpus after removing the seeds from it. This slightly improves the alignment of the bi-texts. We extract other trusted seeds from the corpus in this round after bootstrapping. We evaluate the extracted seeds and apply the proposer on tagged texts. We do one round of bootstrapping and then stop after observing that no further improvement can be obtained. We conclude with discussing those ambiguous words of the same POS tag in an extracted bilingual lexicon, focusing on the way to automatically detect them.

In Chapter 7 we finally conclude the thesis, discuss the main contributions and give some suggestions for further research.

Chapter 2

Description of the Corpus

We describe below the corpus that we use in our study. We begin with throwing light on the different types of corpora then discuss the rationale behind selecting the Qur'an as our corpus. Finally, we discuss the robustness of the proposed approach, shedding light on some features of the linguistic style of the Qur'an.

2.1 Types of Corpora

Depending on the number of languages involved, one can distinguish between monolingual and multilingual corpora (Aijmer, 2008). A monolingual corpus is composed of texts in one language. As regards multilingual corpora, a fundamental distinction is made between comparable corpora and translation corpora. According to Altenberg and Granger (2002), "comparable corpora consist of original texts in each language, matched as far as possible in terms of text type, subject matter and communication function". Corpora of this kind can either be restricted to a specific domain (e.g. genetic engineering, job interviews, religious texts) or be large balanced corpora representing a wide range of genres. Genres here refer to the text categories that can be easily distinguished such as novels, newspaper articles, public speeches, etc. This should be distinguished from text types which are distinguished on a linguistic basis. For instance, text types are normally given such labels as 'informational interaction', 'learned exposition' and 'involved persuasion' (Biber and Finegan, 1991). Translation corpora contain original texts in one language and their translations into one or several other languages. If the translations go in one direction only (from English to Arabic for example) they are unidirectional; if they go in both directions (from English to Arabic and from Arabic to English) they are bidirectional. The term 'parallel corpus' is sometimes used as an umbrella term for both comparable and translation corpora, but it seems more appropriate for translation corpora, where a unit (paragraph, sentence or phrase) in the original text is aligned with the corresponding unit in the translation (Altenberg and Granger, 2002). The classification of corpora can be illustrated in the following figure.

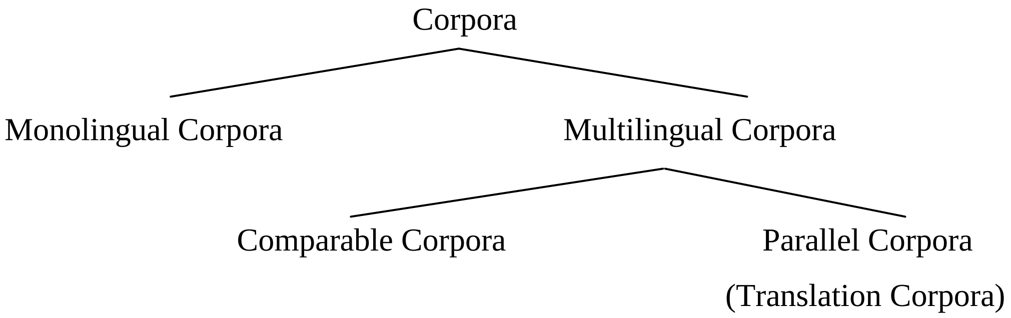


Figure 2.1: Types of corpora

It should be noted that when two languages are involved the corpus is referred to as bilingual.

Our corpus is classified as a parallel corpus. It is also known as bi-texts (Melamed, 2000). In other words, it is a translation corpus, where a verse in the Arabic original text is in parallel with the corresponding verse in the English translation. The texts in the corpus belong to a specific domain. They are religious texts because our corpus contains the texts of the Qur'an as has been mentioned above. We use different versions for our parallel corpus. The first version contains raw texts without any linguistic annotations. The second version of the corpus contains texts annotated with POS tags. As for the third and final version, it contains POS tags along with DRs for some basic constructions. This will be made clear when we discuss our dependency parser in chapter 5.

2.2 The Rationale behind our Selection

We have indicated above that we use the original Arabic text of the Qur'an and its translation into English as our parallel bilingual corpus for extracting translational equivalents. The Qur'anic text, as pointed out earlier, is basically diacritized. The Qur'anic corpus consists of around 78,000 tokens; around 19,000 vowelized word types and about 15,000 non-vowelized word types. This corpus is small in size by statistical analysis standards (Church and Mercer, 1993). However, this size is relatively enough for our work as far as POS tagging and DRs are concerned. As is well-known, one needs less data for learning about word classes than one does for learning about individual words. Hence, as regards our investigation of different techniques for lexical selection, i.e. proposing translational equivalents, the size of the corpus we have is not big enough, since this module deals with individual words. Nonetheless, the results we obtain are promising and can be a preliminary step for further research.

A number of English translations for the Qur'an have become available for non-Arabic-speaking people. Some translators of the Qur'an generally attempt to remain as close as possible to the original text in order to reflect some features of the Qur'anic style in their translations. The English translation that we use in our work is that rendered by Ghali (2005)[4]. In a review by Johnson-Davies in Al-Ahram Weekly (2002), it was mentioned that "the translation by Dr. Ghali shows clearly that its translator has gone to the trouble of consulting the well-known Arabic commentaries. The result is therefore a translation which has all the appearance of accuracy." We have chosen Ghali's translation (2005) from among a number of other translations that we have reviewed. The reason for this choice is that we have found that Ghali's translation is less interpretive or less explanatory than other translations. He sticks as much as possible to the SL wording, giving explanatory notes when necessary. This idea is expressed in his preface to the book as he says "one has to ..... emphasize the strict adherence to the Arabic text, and the obvious avoidance of irrelevant interpretations and explications" (Ghali, 2005). Furthermore, his explanatory notes are given between parenthetical brackets, which makes them easy to remove by using regular expressions (REs).

Two facts about the Qur'an have been referred to in the two previous paragraphs. The first fact is that it is a diacritized text and the second one is that it has been translated into English. These two facts are the motive behind choosing the Qur'anic text to be our corpus.

- Firstly, we need an available Arabic-English parallel corpus.
- Secondly, we need the Arabic text to be diacritized to get our lexicon-free POS tagger off the ground.

[4] Ghali's (2005) "Towards Understanding The Ever-Glorious Qur'an" is the 4th edition of the book, which started to appear in the 1990s. It is available online at: http://quran.com/

It is noteworthy that we use the diacritized text in the early training stages of the POS tagger but we then remove diacritics and end up with a POS tagger for undiacritized text. We then use the Arabic undiacritized text for all subsequent stages of processing. The reason for working on the undiacritized form of the Qur'an is that we believe that the results we obtain would also be obtained if our framework were applied to MSA, which is normally written without diacritics, if we had a diacritized parallel MSA corpus. As pointed out earlier, it is possible to automatically diacritize text with reasonable accuracy. We should make a word of caution here. We are not trying to translate the Qur'an by the machine, but we use the Qur'anic text and its English translation as a source of data for investigating our approach towards lexical selection for MT.

2.3 Robustness of the Approach

As pointed out above, we use the Qur'anic original text and an English translation of it as our parallel bilingual corpus to extract translational equivalents. We have noted that we start with the diacritized text of the Qur'an then remove diacritics from the text. The methods we apply to the Qur'anic corpus could be applied to MSA if we had a parallel corpus of initially diacritized Arabic and English translation. This is because our methods are not specific to the text of the Qur'an but can be workable for other types of Arabic texts. Moreover, using the Qur'anic corpus for implementing our methods emphasizes the robustness of our approach. This is because the Qur'anic text has some common rhetorical peculiarities or features that are uncommon in MSA texts. These peculiarities pose a challenge for our methods to achieve lexical selection for Arabic-English MT. There are many linguistic or rather rhetorical features of the Qur'an, which make its language unique. However, we will discuss only some of those features that are problematic for our techniques. Consequently, those problematic features indicate that the approach we use is robust and effective.

Before introducing the linguistic features that are peculiar to the Qur'an, there are some linguistic features that characterize the Arabic language in general and pose a challenge for any NLP task for Arabic. These features are discussed in detail in chapter 5. But we will refer here to only two characteristics that are more problematic for our current task and apply to both CA and MSA.

- Arabic is morphologically rich, and often a single word will consist of a stem with multiple fused affixes and clitics. Thus, one word may correspond to a number of English words, which poses a challenge for the selection process. The following example throws light on this point.

2.1 ﴿ فَأَسْقَيْنَاكُمُوهُ ﴾

faOasoqayonaAkumuwhu

fa	*Oasoqayo*	*naA*	*kumuw*	*hu*
then	gave to drink	we	you.pl	it

[then we gave it to you to drink] (Qur'an, 15:22) [5]

So, one Arabic word, which stands as a complete sentence, has eight corresponding words in English.

- Arabic is a relatively free word order language. Therefore, the subject may precede the verb or come after it. Also the object may precede the subject in certain contexts. Thus, the orders: SVO, VSO, VOS, OVS are all acceptable sentence structures in Arabic. This point will be made clearer in chapter 5.

2.3.1 Some Linguistic Features of the Qur'anic Corpus

Expressions in the Qur'an are worded in the shortest of forms without loss of clear meaning. Allah (God) challenged the Arabs to produce even a verse like the Qur'an but they could not.

Due to a high degree of lack of exact equivalence between the Qur'anic words and English, the translator of Arabic tends to a rendering which is more or less a paraphrase (Awad, 2005). Hence, as pointed out by Almisned (2001), the English target text (TT) of the Qur'an is wordier than the Arabic source text (ST). Besides the main features of Arabic in general that are mentioned above, there are some

[5] We cite the verse reference with the notation [x:y], where x indicates the chapter number and y indicates the verse number. All translations are taken from Ghali (2005), which we use as the English text of our parallel corpus.

linguistic, or rather rhetorical, features that are very common in the Qur'anic corpus. These features are discussed below.

2.3.1.1 Lack of Punctuation

The Qur'an is written without punctuation marks. Thus, it is difficult to know where a sentence ends and another one begins. There are only verse markers that denote the end of verses. It is usually the case that one verse may contain a number of sentences, separated by conjunctions rather than punctuation marks.

There are many long verses in the Qur'an. Such long unpunctuated verses pose a challenge for any alignment algorithm, which consequently makes the selection process a difficult task. It is well-known that punctuation marks are useful features for detecting sentence boundaries (Mubarak et al., 2009b), and for identifying some aspects of meaning, e.g. in case of question marks, quotation marks, or exclamation marks (Jurafsky and Martin, 2009). Since there are no punctuation marks in the Qur'an, we deal with entire verses, not sentences, in Arabic and English. It should be made clear that the English translation of the Arabic original is punctuated so as to convey the meaning to the foreign reader. However, we remove all punctuation marks from the English text to be similar to the Arabic text.

2.3.1.2 Foregrounding and Backgrounding

As pointed out above, the language of the Qur'an is known for its stylistic features. It uses many devices to achieve its stylistic characteristics. One of such devices is the foregrounding and backgrounding. It is also called 'hysteron proteron', which is a figure of speech in which the natural or rational order of its items is reversed. For example, *bred and born* is used instead of *born and bred* . The Qur'an contains many instances of extraposition, fronting and omission for rhetorical reasons. As noted above, Arabic is a relatively free word order language. Thus, we find that the word order is inverted in the Qur'an to achieve specific stylistic effects. This preposing and postposing of elements within a sentence is often referred to in Arabic as التقديم والتأخير *Altaqdiym wa AltaOxiyr* (lit. bringing forward and moving back) or foregrounding and backgrounding. It is a linguistic feature that is used to highlight or downplay certain elements in speech or writing (Elimam, 2009). The fact that Arabic is a morphologically rich language, where verbs are inflected for person, number and gender and words are marked for case by the use of vowel markers at the final letter of a word, allows for this flexibility of word order, since grammatical meaning does not depend completely on the position of words in the sentence, but rather on case marking.

According to Al-Baydawi (1912), there are functions for foregrounding in the Qur'an. These functions include 'specification', 'restriction', 'emphasis', 'glorification' and 'denial'. The following example sheds light on the first function. For more details about such functions the reader is referred to Al-Baydawi (ibid.).

2.2

﴿ يَا بَنِي إِسْرَائِيلَ اذْكُرُواْ نِعْمَتِيَ الَّتِي أَنْعَمْتُ عَلَيْكُمْ وَأَوْفُواْ بِعَهْدِي أُوفِ بِعَهْدِكُمْ وَإِيَّايَ فَارْهَبُونِ ﴾

*yaA baniy IisoraA}iyla A*okuruwAo niEomatiya Al~atiy OanoEamotu Ealayokumo waOawofuwAo biEahodiy Ouwfi biEahodikumo waIiy~aAya faArohabuwni*

[O Seeds (Or: sons) of Israel remember My favor wherewith I favored you, and fulfil My covenant (and) I will fulfil your covenant, and do have awe of Me (only).] (Qur'an, 2:40)

Al-Baydawi (1912) and Al-Alusi (n.d.) explain that the last clause of the previous verse features foregrounding of the object إِيَّايَ *Iiy~aAya* "Me" before فَارْهَبُون *faArohabuwni* "fear Me" or "be in awe of Me" for specification, i.e. indicating that a believer should fear, or have awe of, Allah specifically and no one else.

2.3

﴿ فَقَالَ رَبِّ إِنِّي لِمَا أَنزَلْتَ إِلَيَّ مِنْ خَيْرٍ فَقِيرٌ ﴾

faqaAla rab~i Iin~iy limaA Oanzalota Iilay~a mino xayorK faqiyrN

[Then he said, "Lord! Surely I have need (Literally: I am poor) of whatever charity You will have sent down to me."] (Qur'an, 28:24)

In this Qur'anic structure the word فَقِيرٌ *faqiyrN* "in need" has been backgrounded at the end of the structure in the SL but the English translation has a different word order.

The previous examples have shown the use of non-canonical word order. Since English, unlike Arabic, is a language with a relatively fixed word order, the translation of such foregrounded structures into English normally follows the English word order which is the reverse of the order shown in the previous examples. This, consequently, poses a challenge for the task of lexical selection. Needless to say that the free word order is characteristic of Arabic in general, which poses a challenge for any NLP application, particularly MT. But the Qur'anic text uses many structures that utilize this variation of word order, which makes the task under consideration more complicated.

2.3.1.3 Lexical Compression

According to Abdul-Raof (2001), the lexical items in the Qur'an are generally characterized by lexical compression, where lengthy details of semantic features are compressed and encapsulated in a single word. The following verse contains lexically compressed items.

2.4

﴿ حُرِّمَتْ عَلَيْكُمُ الْمَيْتَةُ وَالدَّمُ وَلَحْمُ الْخِنزِيرِ وَمَا أُهِلَّ لِغَيْرِ اللَّهِ بِهِ وَالْمُنْخَنِقَةُ وَالْمَوْقُوذَةُ وَالْمُتَرَدِّيَةُ وَالنَّطِيحَةُ
وَمَا أَكَلَ السَّبُعُ إِلاَّ مَا ذَكَّيْتُمْ ﴾

*Hur~imato Ealayokumu Alomayotapu waAlod~amu walaHomu Aloxinoziyri wamaA Ouhil~a ligayori All~hi bihi waAlomunoxaniqapu waAlomawoquw*apu waAlomutarad~iyapu waAln~aTiyHapu wamaA Oakala Als~abuEu IilA~a maA *ak~ayotumo*

[Prohibited to you are carrion, (i.e. dead meat) and blood, and the flesh of swine, and what has been acclaimed to other than Allah, and the strangled, and the beaten (to death), and the toppled (to death), and the gored (to death), and that eaten by wild beasts of prey-excepting what you have immolated] (Qur'an, 5:3)

The previous ayah has contained a number of lexically compressed items that are fraught with emotive meanings that are language and culture-specific. Thus, the words المَوْقُوذَة *Alomawoquw*apu* "the beaten to death", المُتَرَدِّيَةُ *Alomutarad~iyapu* “the toppled to death” and النَّطِيحَةُ *Aln~aTiyHapu* “the gored to death” are all pregnant with culture-bound meanings. Abdul-Raof (2001) defines the word المَوْقُوذَةُ *Alomawoquw*apu*, for instance, as “any animal that receives a violent blow, is left to die, and then eaten without being slaughtered.” Here the translator has attempted to render the Arabic words into single English words but he had to add other words between brackets to clarify the meaning. In order to translate the lexically compressed items into English, the translator often uses Multi-word expressions (MWEs) in the TL, which poses a challenge for the current task. Here is another obvious example to illustrate this point. The verb بشّر *ba$~ir* “give good tidings” and other forms of the same meaning are used frequently in the Qur’anic corpus. Such words have no direct equivalents in English, and thus are translated as MWEs.

2.3.1.4 Culture-Bound Items

There are a large number of cultural expressions in the Qur’an, which are also lexically compressed. This leads to wordier English translation so as to convey the intended meaning, which represents a challenge for the selection process. This is made clearer through the following example.

2.5

﴿ وَإِذَا الْمَوْؤُودَةُ سُئِلَتْ بِأَيِّ ذَنبٍ قُتِلَتْ ﴾

> *waIi*aA AlomawoWuwdapu su}ilato biOay~i *anbK qutilato*
>
> [And when the female infant buried alive will be asked. For whichever guilty deed she was killed,] (Qur’an, 81:8-9)

The previous ayahs (or verses) contain the culture-bound item المَوْؤُودَةُ *AlomawoWuwdapu*. This item refers to the pre-Islamic act of burying newborn girls alive. In order to transfer the meaning of this cultural word to English, the translator had to use a number of words. Thus, it is translated as “the female infant buried alive”.

2.3.1.5 Metaphorical Expressions

Many figures of speech are used in the Qur’an. Such colourful images include metaphor, simile, metonymy, hyperbole, synecdoche, irony, etc. All these figures of speech constitute pitfalls for both human translators and MT systems. The current study is by no means investigating these figures of speech in the Qur’an. However, we will highlight metaphorical expressions briefly through giving an example, and see their implication for MT lexical selection. It is not easy for a translator to convey directly the Qur’anic metaphor into English. Thus, he mostly has to use a number of lexical items so as to be able to render the metaphor in English. This, therefore, results in a wordier TT than ST, which consequently poses a challenge for the proposer. The following example illustrates this point.

2.6

﴿ قَالَ رَبِّ إِنِّي وَهَنَ الْعَظْمُ مِنِّي وَاشْتَعَلَ الرَّأْسُ شَيْبًا وَلَمْ أَكُن بِدُعَائِكَ رَبِّ شَقِيًّا ﴾

> *qaAla rab~i Iin~iy wahana AloEaZomu min~iy waA$otaEala Alr~aOosu $ayobFA walamo Oakun biduEaA}ika rab~i $aqiy~FA*
>
> [He said, "Lord! Surely the bone (s) within me have become feeble, and my head is turned white with hoary (hair) (Literally: is aflame with hoary "hair") and I have not been wretched in invoking you, Lord!] (Qur’an, 19:4)

In the previous verse the words وَاشْتَعَلَ الرَّأْسُ شَيْبًا *waA$otaEala Alr~aOosu $ayobFA* “and my head is turned white with hoary (hair)” are used as a metaphorical expression, where, according to Al-Baydawi (1912) and Al-Alusi (nd), grey hair is likened to flames of fire on the common ground of bright light. Then the likened element (flames of fire) is deleted whereas its likened-to element (grey hair) is mentioned. This metaphorical expression, which consists of three words, was translated by Ghali (2005) into nine English words. The translator here has not kept the metaphor in English. He rendered the meaning of it but gave a literal translation of the metaphor between brackets. The fact that three Arabic words have been translated into nine English words in our parallel corpus makes it hard for the proposer to choose the right translation for every SL word.

2.3.1.6 Verbal Idioms

Generally speaking, Qur'anic discourse is extensively rich with verbal idioms which constitute a significant component of Qur'anic vocabulary (Abdul-Raof, 2001). First and foremost, we will give a brief account of idioms, throwing light on their definition and main characteristics. Then, we will give some Qur'anic examples for verbal idioms, the way they are translated in the corpus we use, and their implication for our research objective.

Idioms have been defined by many within the framework of linguistic studies. According to Crystal (2008), an idiom refers to a sequence of words which are semantically and often syntactically restricted, so that they function as a single unit. From a semantic viewpoint, the meanings of the individual words cannot be summed to produce the meaning of the idiomatic expression as a whole. From a syntactic viewpoint, the words often do not permit the usual variability they display in other contexts, e.g. *it's raining cats and dogs*, which means "to rain very heavily", does not permit **it's raining a cat and a dog/dogs and cats*, etc.[6]

It follows from the definition of idioms above that they have generally two major linguistic features: semantic non-compositionality and syntactic inflexibility. However, it is broadly claimed that idioms are not completely non-compositional or inflexible, but show a certain degree of both features. Hence, some idioms are compositional while others are non-compositional. Semantic compositionality, according to Sag et al. (2002), is "a means of describing how the overall sense of a given idiom is related to its parts." So, the idiomatic expression *spill the beans* can be analyzed as being decomposable into *spill* in the sense of "reveal" and *the beans* in the sense of "secrets", which results in the overall compositional reading of "reveal a secret". The idiomatic *kick the bucket,* in contrast, is semantically non-compositional, since its overall meaning of "die" has no relation to any word in the idiomatic expression. As for flexibility, it refers to the syntactic behaviour of idioms. Broadly speaking, Baker (1992) points out that one cannot do the following with an idiom:

- Change the order of words in it;
- Delete a word from it;
- Add a word to it;
- Replace a word with another;

[6] The asterisk is used before a given structure to indicate that it is ungrammatical.

- Change its grammatical structure.

Schenk (1995) explains that some idiom parts are reluctant to undergo certain syntactic operations. Such operations include, for instance, passivization, relativization, clefting and modification. Thus, the idiomatic structure *kick the bucket* cannot undergo the above-mentioned syntactic operations without violating its idiomatic meaning.

2.7 John kicked the bucket

Passivization: * The bucket was kicked by John.
Relativization: * The bucket that John kicked.
Clefting: * It was the bucket that John kicked.
Modification: * John kicked the yellow bucket.

However, the feature of idiom syntactic flexibility is a matter of degree. Therefore, idioms can be classified into fixed, semi-fixed and syntactically flexible expressions (Sag et al., 2002). Fixed expressions are lexically, morphologically and syntactically immutable, such as *by and large*. Semi-fixed expressions are those expressions that undergo some degree of lexical and morphological variations (e.g. in the form of inflection), but the word order is still the same, such as *kicked the bucket.* As for syntactically flexible expressions, they exhibit syntactic variability, such as passivization. Thus, *the cat was let out of the bag* is also acceptable.

It is time now to give some examples of the Qur'anic verbal idioms and their translation in our corpus.

2.8

﴿ إِنَّ فِي ذَلِكَ لَذِكْرَى لِمَن كَانَ لَهُ قَلْبٌ أَوْ أَلْقَى السَّمْعَ وَهُوَ شَهِيدٌ ﴾

*Iin~a fiy *alika la*ikoraY liman kaAna lahu qalobN Oawo OaloqaY Als~amoEa wahuwa $ahiydN*

[Surely in that there is indeed a Reminding to him who has a heart, or is eager (Literally: cast "his" hearing) on hearing, and is a constantly present witness (to the Truth).] (Qur'an, 50:37)

The previous verse contains the verbal idiom أَلْقَى السَّمْعَ *OaloqaY Als~amoEa*. This idiom is composed of the words أَلْقَى *OaloqaY* which means “throw” or “cast”, and السَّمْعَ *Als~amoEa* which means “hearing”. The translator has referred to their literal meaning as “cast … hearing”. However, the two words mean idiomatically “listen attentively”. An idiomatic translation can be given in English as “to give an ear” or “to lend an ear”. The SL words have been rendered in our translation corpus as “is eager on hearing”. This conveys the meaning expressed by the SL words. But the TL words are not a word-to-word translation of the SL. This, consequently, poses a challenge for the proposer which basically relies on the statistical information about the frequency of words in the corpus. This is because the word *OaloqaY* in this example has the corresponding TL words “is eager” in the parallel corpus, but the most frequent translation for this word in the corpus is the TL word “cast”. Even worse, the TL words “is eager on hearing” have different POS categories from the SL words, since the SL words consist of Verb + Noun, while the TL words are composed of Aux + Adj + Prep + Noun.

2.9

﴿ فَرَجَعْنَاكَ إِلَى أُمِّكَ كَيْ تَقَرَّ عَيْنُهَا وَلَا تَحْزَنَ ﴾

> *farajaEonaAka IilaY Oum~ika kayo taqar~a EayonuhaA walaA taHozana*
>
> [So We returned you to your mother so that she might comfort her eye (Literally: that her eye might settle down) and might not grieve.] (Qur’an, 20:40)

The verbal idiom تَقَرَّ عَيْنُهَا *taqar~a EayonuhaA* in this verse means “someone’s eyes become cool, i.e. pleased.” (Abdul-Raof, 2001). It is translated in our English corpus as “she might comfort her eyes”. In addition, a literal translation of the verbal idiom is provided between brackets.

2.3.1.7 Grammatical Shift

Grammatical shift is the most common feature of Qur’anic discourse (Abdul-Raof, 2001). This linguistic device, which is called التفات *AlotifaAt* “change of addressee”, is described by Arabic rhetoricians as شجاعة العربية *$ajaAEap AlEarabiy~ap* “the daring nature of the Arabic language” (Abdel Haleem, 1992). Grammatical shift can be classified into a number of types. These include ‘person and number shift’, for which the Arabic word is basically used, ‘word order shift’, ‘verb tense shift’, and ‘voice shift’ (adapted from Abdul-Raof, 2001 and Abdel Haleem, 1992). We will give an example for the first type, which is the most common of all.

2.10

﴿ وَمَا لِي لاَ أَعْبُدُ الَّذِي فَطَرَنِي وَإِلَيْهِ تُرْجَعُونَ ﴾

*wamaA liy lAa OaEobudu Al~a*iy faTaraniy waIilayohi turojaEuwna*

[And for what should I not worship Him who originated me, and to Him you will be returned?] (Qur’an, 36:22)

In this example there is a shift from first person singular in فَطَرَنِي *faTaraniy* “originated/created me” to second person plural in تُرْجَعُونَ *turojaEuwna* “you will be returned”.

2.4 Summary

In this chapter we have shed light on the different types of corpora which are generally subdivided into monolingual and multilingual corpora. Multilingual corpora are then subdivided into comparable and parallel corpora. We have also clarified that the corpus used in this study is classified as a parallel corpus, where it consists of an Arabic original text and its English translation. The reasons for using the current corpus have also been discussed. These reasons are succinctly summarized in the two following points:

(i) The need for an available Arabic-English parallel corpus.

(ii) The need to start with a diacritized text in the early stage of the entire project.

The Qur’anic corpus meets the two above-mentioned requirements. The nature of the Qur’anic text is challenging owing to a number of features that characterize its linguistic style. This, consequently, means that using such a challenging corpus illustrates the robustness of the adopted approach, since using a less challenging parallel corpus is likely to result in improvement in accuracy scores. In this stream we have given a brief account of only those features which, we believe, make the current corpus very challenging for the task of lexical selection. Seven features have been discussed in this regard as follows:

1- Lack of Punctuation
2- Foregrounding and Backgrounding
3- Lexical Compression
4- Culture-Bound Items
5- Metaphorical Expressions
6- Verbal Idioms
7- Grammatical Shift

Besides these linguistic features, the Qur’anic discourse is, nonetheless, full of rhetorical and stylistic features that need many volumes to talk about.

It goes without saying that it is normally expected that the approach adopted in this study can work better for other types of text in which such linguistic features are absent or rare. However, MSA does share some of these characteristics, particularly the lack of punctuation and consequent long sentences. The Penn Arabic Treebank, for instance, contains numerous sentences with 100 words or more. According to Mubarak et al. (2009b), Arabic texts have inconsistent use of punctuation marks, since, as indicated by Attia (2008), Arabic writers shift between ideas using coordinating conjunctions and resumptive particles instead of punctuation marks.

Chapter 3

An Overview of Machine Translation (MT)

3.1 Introduction

In this chapter we present the state of the art in machine translation (MT), starting with defining MT. Then we discuss the basic strategies that are adopted in the field. In addition, we investigate the different approaches to MT, which are generally classified into rule-based and corpus-based approaches. Since our main task is lexical selection for MT, we will shed light on the related work in this area and the different approaches taken toward achieving the goal of lexical selection. Finally, a summary of the chapter is given, indicating where our work fits in as far as MT is concerned.

MT is defined as “the automatic translation of text or speech from one language to another” (Manning and Schütze, 1999). It is thus the use of computers to automate some or all of the process of translating from one language to another. This involves making the computer acquire and use the kind of knowledge that human translators need in order to embark on a translation task. However, this is not an easy task, since translators need to have four types of knowledge to successfully carry out such a task. These are outlined by Eynde (1993) as follows:

(1) Knowledge of the source language (SL) (lexicon, morphology, syntax, semantics, and pragmatics) in order to understand the meaning of the source text (ST).

(2) Knowledge of the target language (TL) (lexicon, morphology, syntax, semantics, and pragmatics) in order to produce a comprehensible and well-formed text. Both (1) and (2) are called ‘monolingual knowledge’.

(3) Knowledge of the relation between SL and TL in order to be able to transfer lexical items and syntactic structures of the SL to their nearest equivalents in the TL. This is called ‘bilingual knowledge’

(4) Knowledge of the subject matter. This enables the translator to understand the contextual usage of words and phrases. This is called ‘extra-linguistic knowledge’.
As Newmark (1988) puts it, translation is a craft based on the attempt to replace a written message in one language by the same message in another language.

The idea of MT was first brought to the attention of the general research community by the memorandum of Weaver (1949). In the beginning of MT application computer engineers and linguists faced many failures. But now they understand the complexity of the task. Thus, many MT researchers today are fully aware of the elusiveness of the colossal task (Attia, 2008). MT has become a “testing ground for many ideas in Computer Science, Artificial Intelligence and Linguistics and some of the most important developments in these fields have begun in MT” (Arnold et al., 1994).

Although the goal of fully automatic high quality translation (FAHQT) is still far away, many advances have been made in the MT research community. Also many translation applications have now hit the market. In fact, no MT system can produce a 100% accurate translation, and this is “an ideal for the distant future, if it is even achievable in principle” (Hutchins and Somers, 1992). This is because the translation process is so complicated for the machine to handle. Actually, the machine cannot deal with all types of texts. But when an MT system is designed for a small set of the whole language, a high accuracy translation might be achieved. This means that the design of MT systems for small domains is expected to have better results than the case when the domain is unrestricted. This is because the grammar and vocabulary used in a well-defined domain are smaller than what is required for the whole language. In this way lexical and structural ambiguities can be reduced. Some MT systems are specially designed to be applied to small domains, such as the successful Météo project, which translates weather forecasts (Somers, 2003a).

3.2 Basic MT Strategies

Different strategies have been adopted by different research groups since the birth of MT in the 1940s (Hutchins, 1986). The major strategies for MT have been traditionally classified into direct, transfer and interlingua. The differences between the three strategies can be captured in the Vauquois triangle (adapted from Trujillo, 1999) in figure (3.1):

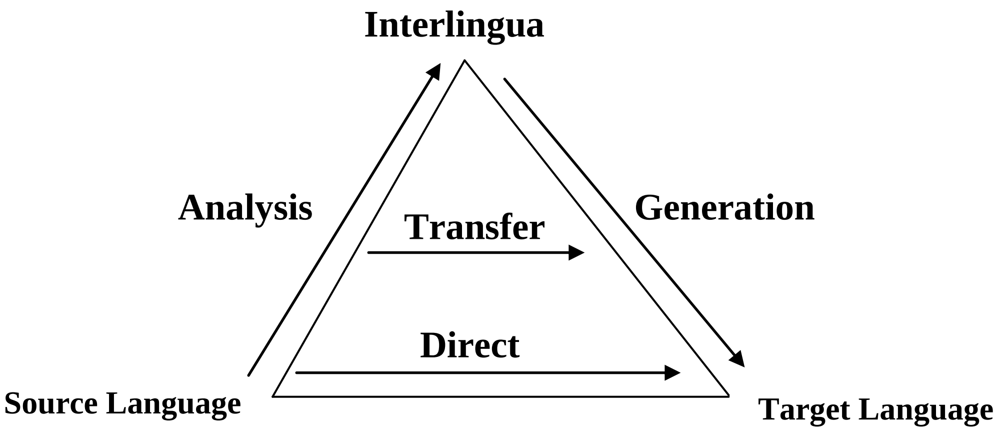


Figure 3.1: The Vauquois triangle

As can be seen in Figure (3.1), **direct** MT systems depend on finding direct correspondences between SL and TL lexical units. The direct method has no modules for SL analysis or TL generation but applies a set of rules for direct translation. Thus, in this type the most important resource is the translation lexicon. The translation is performed word by word with a few rules for local reordering. **Transfer** systems involve three phases: analysis, transfer and generation. The analysis is usually syntactic, since the input sentences in the SL are given a parse of some form according to the employed linguistic framework. These syntactic representations are then transferred to corresponding syntactic structures in the TL. Notably, their result allows substituting SL lexical items by TL lexical items in their context. This transfer is followed by the phase of generating the equivalent sentences in the TL. As for **interlingua** systems, the SL and the TL are never in direct contact. The processing in such systems normally involves two major stages: (i) representing the meaning of an SL sentence in an artificial formal language, i.e. the interlingua, and then (ii) expressing this meaning using the lexical items and syntactic structures of the TL. In other words, in this method the SL is fully analyzed into an abstract language-independent meaning representation from which the TL is generated (Jurafsky and Martin, 2009).

In both the interlingua and the transfer methods a sentence is converted to some representation of its structure or meaning. Both methods make use of abstract representations, but they place different demands on these representations (Bennett, 2003). The transfer strategy can be viewed as “a practical compromise between the efficient use of resources of interlingua systems, and the ease of implementation of direct systems” (Trujillo, 1999). It is noticeable that the transfer method is a middle course between the direct and interlingua approaches. Both interlingual and transfer approaches rely on linguistic knowledge. Several linguistic theories which were adapted to the wider application area of NLP have had an impact on the development of MT. Some of these theories are based on phrase structure grammar (PSG). Among those there are Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982), Generalized Phrase Structure Grammar (GPSG) (Gazdar et al., 1985) and its successor Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994). Some other theories are based on Dependency Grammar (DG). Among the well-known theories that are based on DG are Meaning-Text Theory (MTT) ((Mel’čuk, 1988), and Word Grammar (WG) (Hudson, 1984; 1990). A number of DG-based MT projects have been carried out in Europe, such as Distributed Language Translation (DLT) project (Schubert and Maxwell, 1989).

3.3 Paradigmatic Approaches to MT

The major strategies for MT can be carried out through using different approaches. These approaches can be broadly divided into rule-based MT (RBMT) and corpus-based MT (CBMT). This division is sometimes referred to as rationalist vs. empiricist methods in MT respectively (Somers, 1999).

3.3.1 Rule-Based Machine Translation (RBMT)

In RBMT, which is the original approach to MT, the MT system uses grammatical rules, generally hand-written by linguistic experts, to establish translational equivalence between SL and TL. RBMT systems are developed using one of the three strategies outlined above: direct, transfer or interlingua (Hutchins and Somers, 1992). In the direct method there is very little involved in the analysis stage. The translation draws largely upon a large lexicon to generate a target sentence, allowing for a few rules for some reorganization but with no inherent knowledge of the syntactic relation between the SL and TL strings. In a transfer-based system, translations are produced by analyzing the SL input using rules, translating this analysis into a corresponding TL analysis and then generating an output string. As for interlingua systems, the representation of SL sentences are language-neutral. Then this representation is used to generate the TL sentences.

In controlled environments, RBMT systems are capable of producing translations with reasonable quality due to the large-scale, fine-grained linguistic rules which they employ. Météo system, which was designed to translate short Canadian weather reports from English into French, is a case in point in this regard (Hutchins and Somers, ibid). However, the linguistic resources required for such an MT system can be expensive to build because of the degree of linguistic sophistication they require. Moreover, constructing RBMT systems is very time-consuming and labour-intensive because such linguistic resources need to be hand-crafted. This is usually referred to as the 'knowledge acquisition bottleneck'. In actual fact, RBMT components are often feasible only for the language pair, language direction and text type for which they were initially designed. Thus, switching to other languages and text types can often mean starting from scratch. In an RBMT system, coverage of data can be difficult to achieve, since it is often not possible to predict how newly-added rules will interact with those already in use (Hearne, 2005). Creating rules to deal with different linguistic phenomena can be complex and lead to lack of robustness (Gough, 2005). For instance, if the input is either ill-formed or not covered by the rules then the system will fail to generate a translation (Hearne, ibid). Here lies the advantage of CBMT over RBMT, since adding more examples to an Example-based MT or statistical MT database can improve the system (Gough, ibid).

3.3.2 Corpus-Based Machine Translation (CBMT)

In the early 1990s, research in MT was hit by an apparently new paradigm in which the reliance on linguistic rules was replaced with the use of a corpus of already-translated examples to serve as models to the MT system on which it could base its new translation (Somers and Diaz, 2004). This came to be known as CBMT (or data-driven MT). Generally speaking, this empirical approach to MT uses a corpus of source language sentences and a parallel corpus of target language translations. In point of fact, much recent research in MT tends to focus on the development of corpus-based systems which automatically acquire translation knowledge from aligned or unaligned bilingual corpora (Menezes, 2002). These systems are not generally associated with the manual development of rules and thus can overcome the problem of knowledge acquisition that RBMT systems are prone to (Gough, 2005). In addition, the increasing number of available bilingual corpora and the rapid expansion of the World Wide Web (WWW) have encouraged research towards CBMT. This paradigm shift coincided with a revival of statistical methods, with researchers borrowing ideas heavily from the quickly developing Speech Processing community (Brown et al., 1988). Two types in the domain of CBMT are normally distinguished. These are classified as Example-based (EBMT) and Statistical (SMT). We will throw more light on each type in the following lines.

3.3.2.1 Example-Based Machine Translation (EBMT)

The basic idea behind the EBMT "is to collect a bilingual corpus of translation pairs and then use a best match algorithm to find the closest example to the source phrase in question. This gives a translation template, which can then be filled in by word-for-word translation" (Arnold et al., 1994). Trujillo (1999) refers to the same basic idea in EBMT that in order to translate a sentence you can use previous translation examples of similar sentences. The assumption is that many translations are simple modifications of previous translations. This way of translating saves time and promotes consistency in terminology and style as well. In this regard, EBMT has a strong similarity to the use of translation memory (TM). In fact, both EBMT and TM involve matching the input string against a database of real examples, and identifying the closest matches. The difference between them is that in TM it is up to the translator to decide what to do with the proposed matches (i.e. any adaptation to the output must be done by a translator), whereas in EBMT the automatic process continues by identifying corresponding translation fragments, and then recombining these fragments to produce the target text (Somers, 2003b). The idea of EBMT can be traced to Nagao (1984). He was the first to outline the example-based approach to MT, or 'machine translation by example-guided inference'. Other alternative names are sometimes used by individual authors to refer to the same MT paradigm, such as 'analogy-based', 'memory-based', 'case-based' and 'experience-guided' (Somers, 2003c). The essence of EBMT is succinctly captured by Nagao's much quoted statement:

> "Man does not translate a simple sentence by doing deep linguistic analysis, rather, man does the translation, first, by properly decomposing an input sentence into certain fragmental phrases ..., then by translating these phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference." (Nagao 1984: 178f.)

It is obvious from Nagao's words that EBMT implements the idea of machine translation by the analogy principle. It is based on the intuition that humans translate a new unseen input by making use of previously seen translated examples, rather than performing 'deep linguistic analysis'. Nagao (1984) identifies the three main components of EBMT, which are pointed out by Somers (1999) as follows:

(i) Matching fragments against a database of real examples.

(ii) Identifying the corresponding translation fragments.

(iii) Recombining these fragments to give the target text.

The EBMT model shares similarities in structure with that of the transfer-based RBMT model. As pointed out above, the transfer-based model is composed of three stages: analysis, transfer and generation. In EBMT the search and matching process replaces the source text analysis stage in conventional MT. As for transfer, it is replaced by the extraction and retrieval of examples. This means that once the relevant examples have been selected, the corresponding fragments in the TT are also selected. According to Somers (1999), this is termed 'alignment' or 'adaptation'. Recombination takes the place of the generation stage. This is illustrated in figure (3.2) (taken from Somers, 1999).

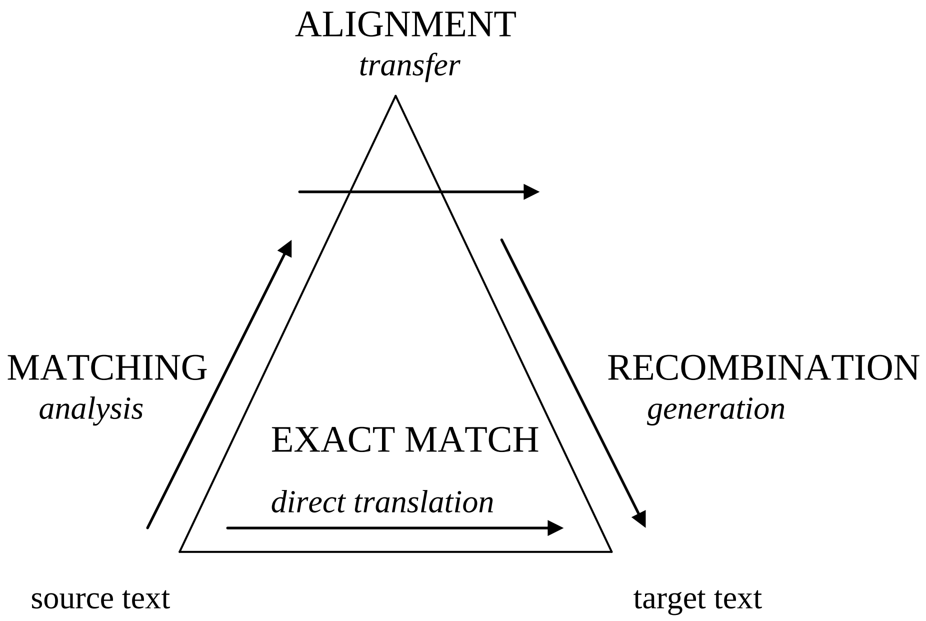


Figure 3.2: The 'Vauquois pyramid' adapted for EBMT (taken from Somers, 1999, Figure 1). The traditional labels are shown in italics, while the EBMT labels are in capitals.

To further illustrate the EBMT process, consider that we wish to translate the sentence in 3.1 (from Trujillo, 1999) into Spanish.

3.1 Julie bought a book on economics.

Let us suppose that we have the corpus in (1), consisting of just 2 simple sentences:

1- (a) Julie bought a notebook ⟺ Julie compró una libreta
(b) Ann read a book on economics ⟺ Anne leyó un libro de economía

Taking the sentences in (1) and applying a bilingual fragment extraction algorithm such as that of Nirenburg et al. (1993) or Somers et al. (2003c), we can then identify and extract the useful bilingual fragments given in (2),

2- (a) Julie bought ⟺ Julie compró
(b) a book on economics ⟺ un libro de economía

We can then combine the fragments in (2) to produce a translation for the new input sentence as shown in (3):

3- Julie bought a book on economics ⟺ Julie compró un libro de economía

We can notice that the sentence pair in (3) did not appear in the original corpus in (1). The sentence pair in (3) can now be added to the example base so that if this same source sentence is encountered later it can then be retrieved as a whole via exact sentence matching and the corresponding target language translation output, thus avoiding the recombination step.

3.3.2.2 Statistical Machine Translation (SMT)

The idea of SMT was in fact first proposed by Weaver (1949) who suggested that statistical methods and ideas from information theory could be applied to the task of automatically translating text from one language to another. SMT systems rely on statistical models of the translation process trained on large amounts of bilingual aligned corpora. Many such systems make use of little or no explicit linguistic information, relying instead on the distributional properties of words and phrases to extract their most likely translational equivalents (Trujillo, 1999). Brown et al. (1988) initiated the approach on which the earliest SMT systems were modelled. Their approach was based only on word-level correspondences. However, the situation has now changed slightly, as more recent research in SMT (Och et al., 1999; Yamada and Knight, 2001; Marcu and Wong, 2002; Charniak et al., 2003; Koehn et al., 2003) has focused on handling phrase-based correspondences. Furthermore, SMT researchers have started to use information about the syntactic structure of language (e.g. Yamada and Knight, 2001; Charniak et al., 2003; Melamed, 2004). The translation model of Yamada and Knight (2001), for instance, assumes bilingual aligned sentence pairs where each SL sentence has been syntactically parsed. The model transforms an SL parse tree into a TL string and the best translation is determined by the language model. According to Menezes (2002), these systems typically obtain a dependency/predicate argument structure for SL and TL sentences in a sentence-aligned bilingual corpus.

It goes without saying that there are usually many acceptable translations of a particular word or sentence, and the choice among them is largely a matter of taste. The initial model of SMT proposed by Brown et al. (1990) takes the view that every sentence in one language is a possible translation of any sentence in the other. The model is based on Bayes' theorem as the following equation shows.

$$P\ (S|T) = \frac{P\ (S)\ P\ (T \mid S)}{P\ (T)} \tag{3.1}$$

The previous equation can be read as follows: for every pair of source and target sentences (*S*, *T*) respectively, we assign a probability *P (S |T)* to be interpreted as the probability that a translator will produce *T* in the target language when presented with *S* in the source language. As Brown et al. (1990) point out, *P* (*T* | *S*) is expected to be very small for the French-English pair in 3.2 below.

3.2 Le matin je me brosse les dents | President Lincoln was a good lawyer.

and relatively large for pairs like 3.3 below

3.3 Le president Lincoln était un bon avocat | President Lincoln was a good lawyer.

Thus, according to this model, the problem of MT is viewed as follows. Given a sentence *T* in the target language, we search for the sentence *S* from which the translator produced *T*. The chance of error is to be minimized by choosing that sentence S so as to maximize *P* (*S* | *T*). The equation to choose the S that maximizes the product can be simplified to give us the equation in 3.2 below.

$$\hat{S} = \underset{S}{\text{argmax}}\ P\ (S)\ P\ (T \mid S) \tag{3.2}$$

In the previous equation this SMT system has two models. The first is the statistical language model that contains monolingual information and the second is a statistical translation model that contains bilingual information. Hence, the previous equation summarizes the three computational challenges that SMT faces. These challenges are summed up as follows:

(a) Estimating the language model probability, P (*S*).

(b) Estimating the translation model probability, P (*T*|*S*).

(c) A technique to search for the TL string which maximizes these probabilities.

To sum up, both EBMT and SMT are data-driven approaches, which require parallel aligned corpora. But the difference between them lies in the fact that EBMT is not statistical and can work on less data, while SMT employs large quantities of data.

3.3.3 Hybrid Approaches

Nowadays, the MT research community is increasingly employing hybrid approaches to MT. Such approaches integrate both rule-based and corpus-based techniques in the development of MT systems. For instance, rules can be learned automatically from corpora, whereas corpus-based approaches are increasingly incorporating linguistic information. In hybrid MT the best techniques are selected from various paradigms. The emergence of hybrid MT systems was due to the fact that neither the example-based nor the statistics-based approaches to MT have turned out to be obviously better than the rule-based approaches, though each of them has shown some promising results in certain cases. MT researchers have started to recognize that some specific problems were particularly suited to one or another of the different MT approaches. For instance, some hybrid systems combine rule-based analysis and generation with example-based transfer. Another combination seems particularly suited to the problem of spoken language translation, where the analysis part may rely more heavily on statistical analysis, while transfer and generation are more suited to a rule-based approach (Somers, 2003b).

3.4 State of the Art in Lexical Selection

Parallel texts (also known as bi-texts or bilingual corpora) have been recently used as useful resources for acquiring linguistic knowledge for a number of NLP applications, especially for MT (Dagan et al., 1991; Matsumoto et al., 1993). A parallel text is composed of a pair of texts in two languages, where one is a translation of the other (Melamed, 1997). These parallel texts, whether sentence-aligned or not, have been used for automatically extracting word correspondences between the two languages concerned. In this regard, different researchers have applied various techniques, using either purely statistical methods (Brown et al., 1990; Gale and Church, 1991) or a combination of both statistical and linguistic information (Dagan et al., 1991; Kumano and Hirakawa, 1994).

Broadly speaking, most approaches to target word selection focus on the word co-occurrence frequencies in the parallel corpus (Gale and Church, 1991, Kumano and Hirakawa, 1994; Melamed, 1995; Kaji and Aizono, 1996). Word co-occurrence can be defined in various ways. The most common way is to have an equal number of sentence-aligned segments in the bi-text so that each pair of SL and TL segments are translations of each other (Melamed, 1997). Then, researchers begin to count the number of times that word-types in one half of the bi-text co-occur with word-types in the other half (Melamed, 2000).

MT researchers have used various knowledge resources (or linguistic information) along with the statistical technique of co-occurrence for lexical selection. Dagan et al. (1991) use statistical data on lexical relations in a TL corpus for the purpose of target word selection in MT. They use the term *lexical relation* to denote the co-occurrence relation of two (or possibly more) specific words in a given sentence, which have a certain syntactic relationship, e.g. between verbs and their different arguments. Thus, they consider word combinations and count how often they appeared in the same syntactic relation. In this way, they resolve the lexical ambiguity in the SL corpus. Their model was evaluated on two sets of Hebrew and German examples.

Melamed (1995) shows how to induce a translation lexicon from a bilingual sentence-aligned corpus using both the statistical properties of the corpus and four external knowledge sources that are cast as filters, so that any subset of them can be cascaded in a uniform framework. These filters are

- POS information
- Machine-Readable Bilingual Dictionaries (MRBDs)
- Cognate heuristics
- Word alignment heuristics

Each of these filters can be placed into the cascade independently of the others. He conducted his experiments on the English-French language pair. He points out that most lexicon entries are improved by only one or two filters, after which more filtering does not result in any significant improvement. Later, Melamed (1997) presents a word-to-word model of translational equivalence, without using any kind of the above-mentioned linguistic knowledge. This model, which assumes that words are translated one-to-one, produces lexicon entries with 0.99 precision and 0.46 recall (i.e. an F-score of 0.628) when trained on 13 million words of the Hansard corpus. However, using the same model on less data, French-English software manuals of about 400,000, Resnik and Melamed (1997) reported 0.94 precision with 0.30 recall (i.e. 0.455 F-score).

Machine-Readable Dictionaries (MRDs) have also been used in the area of lexical selection. Kaji and Aizono (1996), for instance, utilize a method that associates a pair of words through their co-occurrence information with the assistance of a bilingual Japanese-English dictionary that contains 60,000 entry words to extract word correspondences from a Japanese-English non-aligned corpus containing about 1,304 sentences. The bi-text is firstly preprocessed by sentence segmentation and morphological analysis. They report a recall score of 0.28 and a precision score of 0.76. The F-score, thus, stands at 0.41. Lee et al. (1999; 2003) use a three-step method for lexical selection, which consists of sense disambiguation of SL words, sense-to-word mapping, and selection of the most appropriate TL lexical item. The knowledge for each step is extracted from an MRD that contains 43,000 entries and a TL monolingual corpus that comprises 600,000 words. They use examples in English-to-Korean translation. Lee et al. (2003) report an accuracy of 54.45 % for translation selection.

Using structured parallel texts, Tiedemann (1998) introduces three different methods for the extraction of translation equivalents between historically related languages. He (ibid) conducts his experiments on Swedish-English and Swedish-German parallel corpora. The three approaches assume sentence alignment, strict translations, and structural and orthographic similarities. A number of preprocessing steps, which include tokenization and compilation of collocations, are carried out before the extraction of equivalents. The three approaches can be illustrated as follows:

(1) Extraction by iterative size reduction.

This method takes advantage of highly structured and short aligned texts like technical documentation. In this approach a basic set of translation equivalents is first extracted. Then this basic dictionary is used to analyze the remaining alignments in an iterative process by removing known translations from the total set of corpus alignments. As a result, the size of the alignments in the corpus decreases and newly alignments are extracted and added to the set of known translations. This process is repeated until no new alignments appear.

(2) Considerations to string similarity.

This method is used to identify slightly modified translation pairs or cognates in bilingual texts in case there are similar character sets and historical relations between the languages under consideration. This method is based on string matching algorithms to compare word pairs. This is particularly profitable in case of technical texts because of similarities in the origin of technical terminology.

(3) Extraction based on statistical measures.

These measures are based on co-occurrence frequencies of single words or word groups in corresponding subparts of the bi-text. The major advantage of statistical measures is that they are language-independent. However, these measures are usually problematic for infrequent words.

Tiedemann (ibid.) reports that the three extraction methods result in high precision but very low recall. Thus, a number of filters have been used to remove those pairs that are most likely wrong. Such filters include length-based filter, similarity filter, frequency filter or the combination of all of these. The final extracted dictionary achieves a precision score of 0.965 and a recall score of 0.283 (i.e. an F-score of 0.437) for the Swedish-English pair and a precision score of 0.967 and a recall score of 0.494 (i.e. an F-score of 0.653) for the Swedish-German pair.

Tufiş and Barbu (2001a) present a statistical approach to automatic extraction of translation lexicons from parallel corpora, which does not need a pre-existing bilingual lexicon for the considered languages. Their approach requires sentence alignment, tokenization, POS tagging and lemmatization. They applied their approach on six pairs of languages, using a parallel corpus of Orwell's 1984 novel (Tufiş and Barbu, 2001b). The TL in these multilingual corpora is English, while the SL is one of the following languages: Bulgarian, Czech, Estonian, Hungarian, Romanian and Slovene. The best score is achieved on the Romanian-English pair, with a precision of 0.983 and a recall of 0.252 (i.e. an F-score of 0.40). This score is achieved on the extracted lexicons that contain adjectives, conjunctions, determiners, numerals, nouns, pronouns, adverbs, prepositions, and verbs. But in later work (Tufiş and Barbu, 2002) report a higher accuracy for Romanian-English lexicon of nouns only, with a precision of 0.782 and a recall of 0.726.

Machine learning techniques have also been used for translation selection. Sato and Saito (2002) have used Support Vector Machines on non-aligned parallel corpora to extract word sequence correspondences (or translation pairs). Their method used features for the translation model which consists of the following:

(i) An existing translation dictionary.

(ii) The number of words.

(iii) The part-of-speech.

(iv) Constituent words (i.e. content words).

(v) Neighbour words (i.e. previous and following words).

Their experiments were also carried out on a Japanese-English corpus, which achieved 0.811 precision and 0.69 recall (i.e. an F-score of 0.745). In the same way, Lee (2006) has proposed a machine learning-based translation selection method that combines variable features from multiple language resources. The utilized resources are: a mono-bilingual dictionary, WordNet, and a TL monolingual corpus. He applied his experiments on the English-Korean pair.

Other researchers have explored the relationship between word-senses and word-uses in a bilingual environment to carry out lexical selection. Piperidis et al. (2005) is a case in point. They used a context vector model for word translation prediction, making use of an English-Greek parallel corpus. The corpus comprises 100, 000 aligned sentences, containing about 830,000 tokens of the selected grammatical categories (nouns, verbs and adjectives). Their approach, which requires sentence alignment, tokenization, POS tagging and lemmatization, is composed of three main steps: bilingual lexicon extraction, context vector creation and lexical transfer selection. They report an overall precision of 0.85, while the maximum recall reaches 0.75 (i.e. an F-score of 0.8).

Syntactic contexts have been used by Gamallo (2005) to help with the extraction of translation equivalents, using an English-French parallel corpus that contains over 2 million token words. He focused on these contexts that he deemed sense-sensitive to link between them in both languages. Such contexts include, for instance, noun-noun, noun-preposition-noun, adjective-noun, and noun-adjective. His approach requires that the texts of both languages should be tokenized, lemmatized, POS tagged and superficially parsed by simple pattern matching to extract sense-sensitive contexts of words. His technique does not use sentence alignment, but aligns the SL and TL texts by detecting natural boundaries such as chapters, specific documents, articles, etc. His approach selected translation equivalents for nouns and adjectives with an average precision of 0.94 and recall of 0.74. This means that the F-score stands at 0.828.

Some researchers have exploited the use of comparable, non-parallel, texts to extract translation equivalents. Gamallo (2007) has used an unsupervised method to learn bilingual equivalents from comparable corpora without requiring external bilingual resources. But he uses some bilingual correspondence between lexico-syntactic templates previously extracted from small parallel texts to find meaningful bilingual anchors within the corpus. Gamallo’s approach is based on three steps: (1) text preprocessing (which includes POS tagging and binary dependencies) (2) extraction of bilingual lexico-syntactic templates from parallel corpora and (3) extraction of word translations from comparable texts using bilingual templates. The experiments were carried out on an English-Spanish comparable, non-parallel corpus selected from the European parliament proceedings parallel corpus. The English part consists of 14 million words, while the size of the Spanish part is nearly 17 million words. The reported accuracy score is 79% which is mostly a precision score. The same approach of using comparable corpora to extract bilingual equivalents has been exploited by Yu and Tsujii (2009). Their approach is based on the observation that a word and its translation share similar dependency relations. In other words, a word and its translation appear in similar lexical contexts or share similar modifiers and heads in comparable corpora. This is termed by Yu and Tsujii (ibid.) *dependency heterogeneity*. Thus, the modifiers and head of unrelated words are different even if they occur in similar context. They focus on extracting a Chinese-English bilingual dictionary for single nouns. To achieve this, they use a Chinese morphological analyzer and an English POS tagger to analyze the raw corpora. Then they use Malt-Parser (Nivre et al., 2007) to obtain dependency relations for both the Chinese corpus and the English corpus. In addition, they use a stemmer to stem the translation candidates in the English corpus, but keep the original form of their heads and modifiers to avoid excessive stemming. Next, they remove stop words from the corpus. Finally, they remove dependencies including punctuation and remove the sentences with more than 30 words from both the English corpus and Chinese corpus to reduce the effect of parsing error on dictionary extraction. They report an average accuracy of 57.58%.

Monolingual corpora have been also used for learning translation equivalents. For instance, Koehn and Knight (2002) present an approach for constructing a word-level translation lexicon from monolingual corpora, using various cues such as cognates, similar context, similar spelling and word frequency. They used their approach to construct a German-English noun lexicon which achieved 39% accuracy. More recently Haghighi et al. (2008) use also monolingual corpora to extract equivalents. In their approach word types in each language are characterized by purely monolingual features, such as context counts and orthographic substrings. They take as input two monolingual corpora and some seed translations. They report a precision of 0.89 and a recall of 0.33 on English-Spanish induction, with F-score standing at 0.48.

Another group of MT researchers has started to focus on the *global* not *local* associations of TL words or phrases with SL words or phrases in aligned parallel corpora. Thus, Bangalore et al. (2007) have presented a novel approach to lexical selection, where the TL words are associated with the entire SL sentence without the need to compute local associations. The result is a bag of words in the TL and the sentence has to be reconstructed (or permuted) using this bag of words. The words in the bag might be enhanced with rich syntactic information that could aid in reconstructing the TL sentence. Thus, they present an approach for both lexical selection and reconstruction of the selected words. The intuition, they argue, is that there may be lexico-syntactic features of the SL sentence that might trigger the presence of a target word in the TL sentence. In addition, they point out, it might be difficult to associate a TL word to an SL word in various situations: (i) when the translations are not exact but paraphrases. (ii) when the TL does not have one lexical item to express the same concept that is expressed by an SL word. They (ibid) maintain that this approach to lexical selection has the potential to avoid limitations of word-alignment based methods for translation between languages with different word order (e.g. English-Japanese). In order to test their approach they perform experiments on the United Nations Arabic-English corpus and the Hansard French-English corpus. They use 1, 000,000 training sentence pairs and tested on 994 test sentences for the UN corpus. As for the Hansard, they use 1.4 million training sentence pairs and 5432 test sentences. They report an F-score of 0.662 on open-class words and 0.726 on closed-class words in the UN Arabic-English corpus. The average score for all words is 0.695. The F-score for the Hansard corpus is lower, where it scored 0.565 for open-class words and 0.634 for closed-class words. The average score for all lexical items is thus 0.608. The same *global* lexical selection approach is further exploited by Venkatapathy and Bangalore (2009) in their model for translation from English to Hindi and vice versa. They indicate that this approach is more suitable for morphologically rich and relatively free-word order languages such as the Indian languages where the grammatical role of content words is largely determined by their case markers and not entirely by their positions in the sentence. The best F-score that Venkatapathy and Bangalore (ibid) report is 0.636 for English-Hindi dataset, and 0.68 for Hindi-English. Their experiments were carried out on a parallel corpus of 12300 sentences, containing 294,483 Hindi words and 278,126 English words.

With a similar focus on morphologically rich languages, Saleh and Habash (2009) present an approach for automatic extraction and filtering of a lemma-based Arabic-English dictionary from a sentence-aligned parallel corpus containing 4 million words. They use a morphological disambiguation system to determine the full POS tag, lemma and diacritization. This is done after the text is tokenized and word-aligned using GIZA++ (Och and Ney, 2003). Then translation extraction is done using Pharaoh system tool for phrase-table extraction (Koehn, 2004). In addition, a rule-based machine learning classifier, Ripper (Cohen, 1996), is used to learn noise-filtering rules. Saleh and Habash (ibid.) report a precision score of 0.88 and a recall of 0.59, which means an F-score of 0.706.

As far as our approach to lexical selection is concerned, it draws upon some of the techniques mentioned in the previous attempts. The main features of our approach can be summarized as follows:

- Using a small-sized, partially aligned parallel corpus for the Arabic-English language pair.[7]
- Exploiting word co-occurrence frequencies in the parallel corpus.
- Using POS information and co-occurrence syntax-based lexical relations of words in a given sentence, e.g. between verbs and their different arguments.

[7] The corpus is verse-aligned where each SL verse and its corresponding TL verse are on a separate line.

- Automatically extracting a lexicon without any manual intervention, which consequently constitutes the main step towards lexical selection.
- Picking up the word with the highest frequency to be the translation equivalent for a given SL word.
- Automatically detecting ambiguous words where each sense has the same POS tag with a view to handle them automatically in future work.

The final F-score we obtain is 0.701, with precision standing at 0.707 and recall at 0.695, which is comparable with state-of-the-art approaches for automatic lexical selection. Table (3.1) below compares between the above-mentioned models with respect to their reported accuracy and the linguistic resources they have used.

Model	**Languages**	**Corpus Type**	**Data Size**	**Linguistic Resources**						**F-Score**	**Accuracy**
				Lexicon	**Sentence Alignment**	**Clitic Segmentation**	**Lemmatization**	**POS Tagging**	**Dependency Parsing**		
Gamallo (2005) (Nouns & Adjectives)	English-French	Para.	2M words	×	×		√	√	√	0.828	
Piperidis et al. (2005) (open-class words)	English-Greek	Para.	830K words	×	√		√	√	×	0.80	
Tufiş and Barbu (2002) (nouns only)	Romanian-English	Para.	14K words	×	√		√	√	×	0.753	
Sato & Saito (2002)	Japanese-English	Para.	193K words	√	×		×	√	×	0.745	
Saleh & Habash (2009)	Arabic-English	Para.	4M words	×	√	√	√	√	×	0.706	
Our Approach (open-class words)	Arabic-English	Para.	78K words (Arabic) & 162K	×	×	×	×	√	√	0.701	

			(English)								
Bangalore et al. (2007) (All words)	Arabic-English	Para.	1M sentences	✗	√	√	✗	✗	✗	0.695	
Venkatapathy and Bangalore (2009)	Hindi-English	Para.	294K words (Hindi)	✗	√		✗	√	✗	0.68	
Bangalore et al. (2007) (open-class words)	Arabic-English	Para.	1M sentences	✗	√	√	✗	✗	✗	0.662	
Tiedemann (1998)	Swedish-German	Para.	36K short structures	✗	√		✗	✗	✗	0.653	
Venkatapathy and Bangalore (2009)	English-Hindi	Para.	278K words (English)	✗	√		✗	√	✗	0.636	
Melamed (1997)	French-English	Para.	13M words	✗	√		✗	✗	✗	0.628	
Bangalore et al. (2007)	French-English	Para.	1.4M sentences	✗	√		✗	✗	✗	0.608	
Haghighi et al. (2008)	English-Spanish	Mono.	100K sentences	✗	✗		✗	✗	✗	0.48	
Resnik & Melamed (1997)	French-English	Para.	400K words	✗	√		✗	✗	✗	0.455	
Tiedemann (1998)	Swedish-English	Para.	36K short structures	✗	√		✗	✗	✗	0.437	
Kaji & Aizono (1996)	Japanese-English	Para.	1304 sentences	√	✗		√	√	✗	0.41	
Tufiş and Barbu (2002)	Romanian-English	Para.	14K words	✗	√		√	√	✗	0.40	
Gamallo (2007)	English-Spanish	Comp.	14M words (English)&	✗	✗		✗	√	√		79%

			17M (Spanish)								
Yu and Tsujii (2009) (nouns)	Chinese-English	Comp.	1,132,492 sentences (English) & 665,789 (Chinese)	×	×		×	√	√		57.6%
Lee et al. (2003)	English-Korean	Mono.	600K words	√	×		√	√	√		54.5%
Koehn & Knight (2002) (nouns)	German-English	Mono.	N/A	×	×		×	×	×		39%

Table 3.1: Comparison between different approaches for lexical selection

This table shows different approaches for selection of lexical equivalents, using either parallel (Para.), comparable (Comp.) or monolingual (Mono.) corpora. Some of these attempts extract equivalents for all words; some others focus on open-class words, like our model, while a third group focuses on a particular grammatical category. For example, the model for Tufiş and Barbu (2002) is mentioned twice in the above table. They apply their model on all words and thus obtain an F-score of 0.40. Nonetheless, when they focus on nouns only, their model obtains a higher F-score of 0.753. Some of the models discussed above are evaluated by the standard F-measure, while some others are evaluated by an accuracy score which is probably just measuring the precision. That is why we have given two types of evaluation scores in the table. We firstly ranked the F-scores in descending order, and then ranked the accuracy scores in descending order as well. It should be noted that some factors have an effect on the scores obtained, such as the size of the data, the language pair and the number of linguistic resources that are used. Thus, training the different models on a larger data set results in higher scores. Also, a pair of languages may be unrelated, e.g. Arabic-English, and thus there exist linguistic differences between both languages that affect a given model's F-score. Thirdly, some approaches presume a number of linguistic knowledge resources to carry out the selection task. These resources are either preprocessing steps such as tokenization, POS tagging, or using a lexicon to guide the model in question. As far as our approach is concerned, we have not used a lexicon. All we exploited is a few linguistic resources that we have automatically developed with the least manual intervention, as will be shown in the coming chapters. We have mentioned 22 different models for translation selection and the F-score we obtained gave us a good rank in the first half of this table. If we just draw a comparison between our model and that of Saleh and Habash (2009), we will find that both models are very similar in F-score, with slight improvement for Saleh and Habash (ibid.). Nonetheless, our model uses fewer linguistic resources and is trained on a lesser data set (78K words versus 4M words).

3.5 Summary

This chapter has given an overview of the MT process. Three basic strategies are generally used in the field: direct, interlingua and transfer. Approaching the MT task can be done through a number of ways. These ways are either rule-based or data-driven. The three strategies mentioned above can be exploited in both the 'rule-based' and 'data-driven' approaches. But they are frequently used with the rule-based techniques, where grammatical rules generally written by linguists are used to establish translational equivalents between SL and TL. The data-driven or corpus-based approach uses a parallel corpus of SL sentences and their translation into the TL to serve as a model to the MT system on which it can base its new translation.

Each of the MT approaches has certain advantages that are lacking in other approaches. Thus, corpus-based approaches are robust in the sense that they most likely produce some translation, even if the input is ungrammatical or ill-formed. This characteristic can be lacking in a rule-based MT system, since if such a system cannot find a sequence of rules which can be applied successfully to the input then no translation will be produced. Another attractive characteristic of corpus-based approaches is ease of knowledge acquisition. Rule-based systems, on the other hand, are time-consuming, labour-intensive, expensive to build and difficult to maintain and update, while it is much easier to acquire new raw data. Nonetheless, corpus-based approaches (i.e. statistical and example-based) are not good at handling linguistic phenomena such as agreement. But rule-based systems can handle such linguistic phenomena. In addition, corpus-based systems have the potential to learn from new translations, while rule-based systems do not integrate this information. Consequently, hybrid systems attempt to combine the positive elements of both the rule-based and corpus-based approaches to MT. As a result, such a combined system has the potential to be highly accurate, robust, cost-effective to build and adaptable (Hearne, 2005).

As regards the lexical selection task, many approaches have been presented above. These approaches use either statistical techniques that rely on word co-occurrence frequencies in a bilingual corpus or couple these techniques with linguistic information (or knowledge resources). These resources include POS information, lexical relations and MRDs. Generally speaking, our approach to lexical selection falls broadly within the corpus-based paradigm, in which we use the least possible manual intervention**.** We should make it clear that the current study does not aim at building an MT system or evaluating a specific MT system, but rather deals with one of the main problems that face MT, i.e. target word selection, in a computational framework. We aim to automatically extract bilingual translational equivalents from the used parallel corpus without using a hand-coded lexicon. The results to be obtained from this study can then be incorporated into an MT system to tackle its lexical component and thus give better results.

Chapter 4

Part-of-Speech (POS) Tagging

4.1 Introduction

As pointed out in the introductory chapter, we will carry out a number of preprocessing steps to annotate the parallel corpus that we use for extracting lexical equivalents. This allows us to test our proposer on both raw texts and linguistically annotated texts. We annotate the bi-texts, i.e. Arabic and English texts, with both POS tags and dependency relations (DRs). This linguistic annotation removes part of the ambiguity inherent in lexical items and thus guides the proposer to select the right TL word for a given SL word. A number of examples in both Arabic and English have been given in chapter 1 to illustrate this point. In this chapter we discuss the first linguistic annotation we carry out, namely POS tagging for both Arabic and English. We begin with describing the morphological nature of Arabic words. Then we explore POS tagging and the different approaches in the field. In addition, we review the state of the art in Arabic POS tagging. We then move on to present the lexicon-free tagger that we have built for Arabic, using a combination of rule-based, transformation-based and probabilistic techniques. In conclusion, we throw light on the English POS tagger and the tagset used to tag the English translation.

4.2 Arabic Morphological Analysis

Broadly speaking, Arabic is a highly inflected language with a rich and complex morphological system, where words are explicitly marked for case, gender, number, definiteness, mood, person, voice, tense and other features (Maamouri et al., 2006; Diab, 2007). The Arabic morphological system is generally considered to be of the non-concatenative type where morphemes are not combined sequentially, but root letters are interdigitated with patterns to form stems (Soudi et al., 2001). Thus, the main characteristic feature of Arabic is that most words are built up from roots by following certain fixed patterns and adding infixes, prefixes and suffixes (Khoja, 2001a).

4.2.1 Arabic Grammatical Parts-of-Speech

It is expedient to start with explaining the grammatical categories that are basically used to classify Arabic words with regard to their parts of speech. Then we will shed light on the nature of Arabic morphology whether derivational or inflectional. Arabic grammarians traditionally analyze all Arabic words into three main grammatical categories. These categories could be classified into further sub-classes which collectively cover the whole of the Arabic language (Haywood and Nahmad, 2005). The subdivisions of the three main classes will be discussed when we talk about our POS tagger. The three main categories are described by Khoja (2001a; 2003) as follows:

1. **Noun**: A noun in Arabic is a name or a word that is used to describe a person, thing, or idea. The noun class in Arabic is traditionally subdivided into derivatives (i.e. nouns derived from verbs, nouns derived from other nouns, and nouns derived from particles) and primitives (i.e. nouns not derived from any other categories). These nouns could be further subcategorized by number, gender and case. In addition, this class includes what would be classified as participles, pronouns, relative pronouns, demonstratives, interrogatives and numbers.
2. **Verb**: The verb classification in Arabic is similar to that in English, though the tenses and aspects are different. Arabic verbs are deficient in tenses, and these tenses do not have precise time significances as in English. The verb category can be subdivided into perfect, imperfect, and imperative. Further subdivisions of the verb class are possible using number, person and gender.
3. **Particle**: The particle class includes: prepositions, adverbs, conjunctions, interrogative particles, exceptions, interjections, negations, and subordinations.

It is worth noting that the noun and verb categories are used to classify open-class words, while the particle category classifies the closed-class words. Arabic open-class words are generated out of a finite set of roots transformed into stems using one or more patterns. Thus, a single root can generate hundreds of words in the form of nouns or verbs (Ahmed, 2005). For example, the Arabic word مدرس *mdrs* “teacher” is built up from the root *drs* “study” by following the pattern مفعل **mfEl**[8], where the letter ف “**f**” is replaced by the first consonant in the root, letter ع “**E**” is replaced by the second consonant in the root, and letter ل “**l**” is replaced by the third consonant in the root (Khoja et al., 2001b).

4.2.2 Arabic Roots and Patterns

Arabic derivational morphology is based on the principle of **Root** and **Pattern**. A root is a sequence of mostly three or four consonants which are called radicals. The pattern, on the other hand, is represented by inserting a template of vowels in the slot within the root’s consonants (Beesley, 2001). Thus, as McCarthy (1981) points out, stems are formed by a derivational combination of a root morpheme and a vowel melody. The two are arranged according to canonical patterns. Roots are said to interdigitate with patterns to form stems. For example, the Arabic stem كَتَبَ *katab* “he wrote” is composed of the morpheme *ktb* “the notion of writing” and the vowel melody morpheme ’*a-a*’. The two are integrated according to the pattern CVCVC (C=consonant, V=vowel). This means that word structure in Arabic morphology is not built linearly as is the case in concatenative morphological systems such as English.

Arabic roots are subclassified into biliteral, triliteral, quadriliteral and quinquiliteral. For each of these types of roots Arabic has a set of patterns, which include the root consonants and slotted vowels between these consonants. Biliteral and quinquiliteral roots are rare, while triliteral and quadriliteral roots are the most common. Both triliteral (the most common of all) and quadriliteral roots have a number of derived forms. Such derived (or augmented) forms are expansions of the basic stem by various means, each of which implies (though not consistently) a specific semantic extension of the root meaning (Badawi et al., 2004). Both verbs and nouns are derived according to patterns. Here we will shed light on the verbal patterns in the following lines.

[8] The Arabic grammarians illustrate their measures with the use of the triliteral root فعل fEl.

Though Arabic is poor in verb tenses, it is rich in derived verb forms which extend or modify the meaning of the root form of the verb. We will shed light here on the most common derived forms of triliteral and quadriliteral verbs. The relations between the various forms are conventionally illustrated with respect to an abstract verb pattern, where the three consonants are written as **f*E*l** (Haywood and Nahmad, 2005; Badawi et al., 2004). The following tables follow this convention, with concrete examples of each case.

Derived Forms	**Examples**
Form I: **فعَلَ** ***faEala***	كَتَبَ *kataba* “to write”
Form II***:*** **فعَّلَ** ***faE~ala***	علَّمَ *Eal~ama* “to teach”
Form III: **فاعَلَ** ***faAEala***	كَاتَبَ *kaAtaba* “to correspond with”
Form IV: **أفْعَلَ** ***OafoEala***	أعلْمَ *OaEolama* “to inform”
Form V: **تفعَّلَ** ***tafaE~ala***	تَعلَّمَ *taEal~ama* “to learn”
Form VI: **تَفاعَلَ** ***tafaAEala***	تَقَاتَلَ *taqaAtala* “to fight each other”
Form VII: **انْفعَلَ** ***AnofaEala***	انْعقَدَ *AnoEaqada* “to be held”
Form VIII: **افْتَعَلَ** ***AfotaEala***	اجْتَمَعَ *AjotamaEa* “to assemble”
Form IX: **افْعَلَّ** ***AfoEal~a***	احْمَرَّ *AHomar~a* “to become red”
Form X: **اسْتَفْعَلَ** ***AsotafoEala***	اسْتَحْسَنَ *AsotaHosana* “to regard as good, admire”

Table 4.1: Derived forms for triliteral verbs

As for the derived forms of quadriliteral verbs, they are listed in the following table.

Derived Forms	**Examples**
Form I: **فعْلَلَ faEolala**	زَلْزَلَ *zalozala* “to shake”
Form II: **تفَعْلَلَ tafaEolala**	تَزَلْزَلَ *tazalozala* “to quake, or to be shaken”
Form III: **افْعَنْلَلَ AfoEanolala**	اخْرَنْطَمَ *AxoranoTama* “to raise the nose, be proud”
Form IV: **افْعَلَلَّ AfoEalal~a**	اطْمَأَنَّ *ATomaOan~a* “to be tranquil”

Table 4.2: Derived forms for quadriliteral verbs

It should be noticed that a root can be combined with a number of patterns to produce derived forms that may be grammatically different, but are related in their meanings. The following table shows different forms derived from the same root.

Root	**Derived Patterns**	**Derived Words**	**Grammatical Category**	**Gloss**
كتب *ktb* **فعل** ***fEl***	**فَعَلَ** ***faEala***	كَتَبَ *kataba*	verb	(he) wrote
	فَعَّلَ ***faE~ala***	كَتَّبَ *kat~aba*	verb	(he) caused (one) to write
	فَاعِل ***faAEil***	كَاتِبْ *kaAtib*	noun	writer
	مَفْعَل ***mafoEal***	مَكْتَب *makotab*	noun	desk/office
	مَفْعَلَة ***mafoEalap***	مَكْتَبَة *makotabap*	noun	library
	فُعُل ***fuEul***	كُتُب *kutub*	noun	books

Table 4.3: Example of derived forms for the root كتب *ktb*

We can notice that a number of verbal and nominal forms with related meanings have been derived from one root.

4.2.3 Linguistic Analysis of Non-concatenative Morphology

As shown by the examples above, Arabic morphology is of the non-concatenative (or non-linear) type, where morphemes are combined in more complex ways. Unlike English, for example, in which morphemes are combined linearly, Arabic words are formed in a non-linear way. This type of non-concatenative morphology is sometimes referred to as template morphology or root and pattern (Beesley, 1998a; 1998b).

The best known linguistic analysis of these examples is given by McCarthy (1981), and McCarthy and Prince (1990) who pointed out that the non-concatenative morphology of Arabic could be represented by separating the consonants and vowels of a word form onto three separate levels or tiers. Thus, the form كُتِب *kutib* "to be written" is represented as follows:

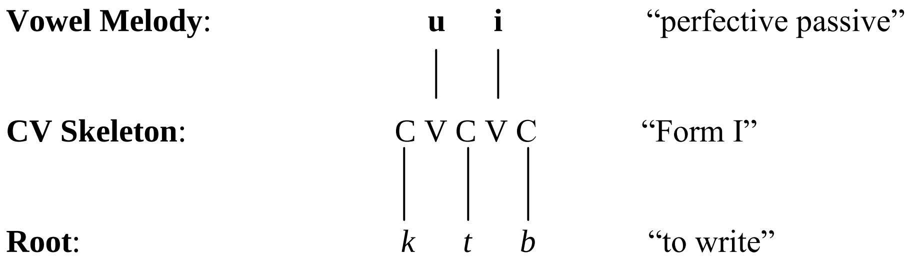


Figure 4.1: Representation of Arabic morphology on separate tiers

McCarthy's scheme illustrates that each of the three tiers contributes to the word form. These tiers are the consonantal root {ktb}, the vowel melody {u, i}, which indicates the passive in this case, and the skeletal pattern CVCVC that indicates how to combine the three parts. The non-linear combination of these three tiers makes the complete lexical form *kutib* "it was written". If one pattern is changed, the resulting word changes. Thus, if we change the vocalic tier to {a, a}, the resulting word will be كَتَبَ *katab* "to write". But if we change the CV skeleton to CVVCVC, it will result in كاتِبَ *kaAtab* "to correspond with".

The non-concatenative nature of Arabic morphology has been elaborated by Habash (2007). He distinguishes between two different aspects of morphemes, i.e. **type** versus **function**. Morpheme type refers to the different kinds of morphemes and their interactions with each other. Morpheme function, in contrast, refers to the distinction between derivational morphology and inflectional morphology. Morpheme type can be classified into three categories as illustrated in the following table.

Templatic Morphemes			**Affixational Morphemes**			**Non-Templatic Word Stems**
root كتب *ktb*	pattern 1V2V3	vocalism a-a	prefix يَ *ya*	suffix وُنَ *uwna*	example يَكْتُبُونَ *yakotubuwn* "they write"	These are word stems that are not constructed from a combination of root, pattern and

						vocalism. They tend to be foreign names such as وَاشِنطُن *waA$inTun* “Washington” or borrowed terms such as the word دِيمُوقرَاطِية *diymuwqraATiy~ap* “democracy”.

Table 4.4: Classification of morpheme type

As illustrated in the above table, a templatic word stem e.g. كتب *katab* “to write” is formed by a combination of a root, pattern and vocalism. Thus, an Arabic word is constructed by first creating a word stem from templatic morphemes or using a non-templatic word stem, to which affixational morphemes are then added. For example, the word وَسَيَكْتُبُونَهَا *wasayakotubuwnahaA* “and they will write it” has three prefixes, and two suffixes in addition to a root, a pattern and a vocalism.

As for morpheme function, a distinction is often made between derivational morphology and inflectional morphology. Derivational morphology is concerned with the formation of new words from other words where the core meaning is modified. In inflection morphology, on the other hand, the core meaning of the word remains intact and the extensions are always predictable. This type of morphology is concerned with inflectional categories that reflect grammatical processes such as pluralization of nouns. The difference between both types can be illustrated in the following table.

Derivational Morphology			**Inflectional Morphology**	
Root	**Pattern**	**Derived Forms**	**Singular Form**	**Plural Form**
درس *drs*	CVVCVC	دَارِس *daAris* “student”	دَارِس *daAris* “student”	دَارِسُون *daArisuwn* “students”
	muCVCCVC	مُدَرِّس *mudar~is*		

	maCVCVCVp	“teacher” مَدْرَسَة *madorasap* "school"		

Table 4.5: Classification of morpheme function

4.2.4 Arabic Word Structure

According to the above discussion, Arabic word forms are thus complex units which encompass the following:-

- **Proclitics**, which occur at the beginning of a word. These include mono-consonantal conjunctions (such as وَ *wa*-, “and”, لِ *li*-, “in order to”), prepositions (e.g. بِ *bi*-, “in”, “at” or “by”, لِ *li*-“for”),…etc.
- **Prefixes**. This category includes, for instance, the prefixes of the imperfective, e.g. يَ *ya*-, prefixed morpheme of the 3rd person. It also includes the definite article ال *Al* “the”.
- **A stem**, which can be represented in terms of a ROOT and a PATTERN. The root is an ordered triple or quadruple of consonants, as described above.
- **Suffixes**, such as verb endings, nominal cases, nominal feminine ending, plural markers …etc.
- **Enclitics**, which occur at the end of a word**.** In Arabic enclitics are complement pronouns. (Cavalli-Sforza et al., 2000; Dichy, 2001; Abbès et al., 2004; Smrž, 2007).

The following figure illustrates the Arabic word structure.

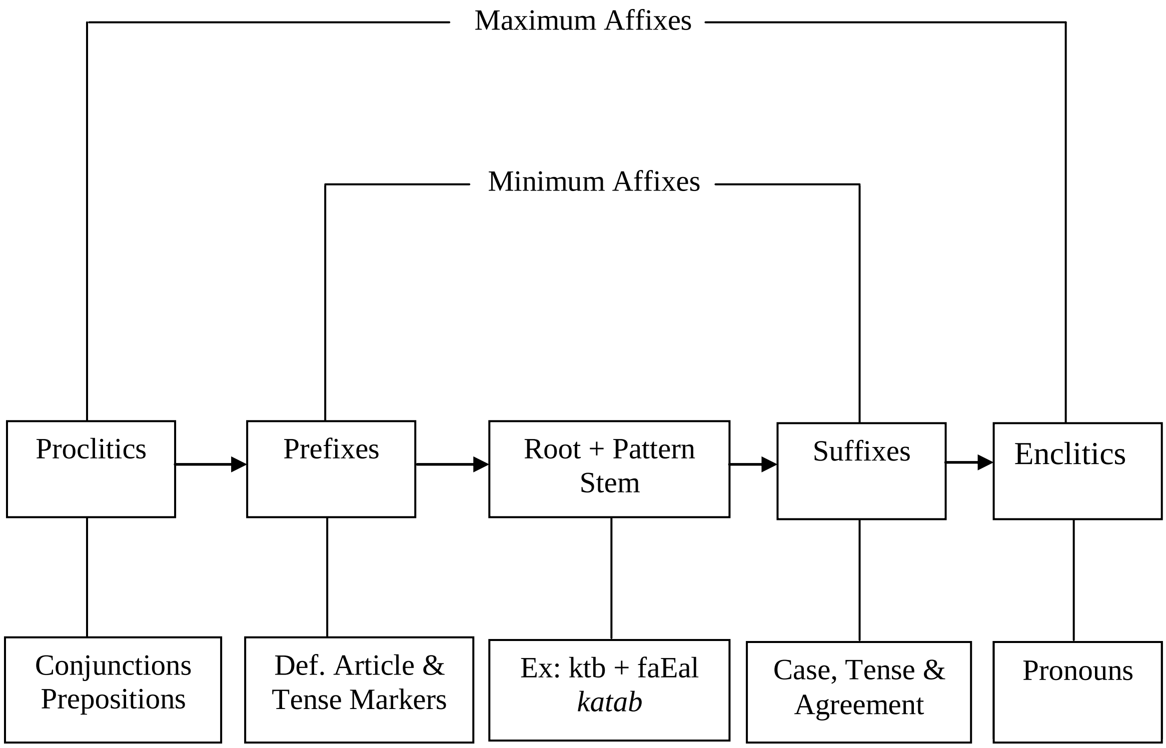


Figure 4.2: Arabic word formation

The details of all affixes will be discussed when we present our POS tagger later in this chapter.

4.3 POS Tagging

POS tagging, also called word-class tagging or grammatical tagging, is one of the basic and indispensable tasks in natural language processing. It is generally considered the commonest form of corpus annotation. It "is the process of assigning a part-of-speech or other syntactic class marker to each word in a corpus" (Jurafsky and Martin, 2009). This process usually forms a basis for more sophisticated annotation such as syntactic parsing and semantic disambiguation (Garside and Smith, 1997).

Many words are ambiguous as to which part of speech they belong to. For example, the word *book* in English has more than one possible part of speech tag. It can be a verb or a noun as in 4.1 and 4.2 respectively.

4.1 Book that flight.

4.2 Give me that book.

The task of POS tagging is to resolve these ambiguities, choosing the proper part of speech tag in the sentential context in which a word is used.

4.3.1 Tagsets

A tagset is simply a list of tags used for a given task of grammatical tagging. Tagsets vary according to the task they are designed for. Thus, it is relatively easy to increase or decrease the size of a tagset, according to the emphasis a particular project has (Leech, 1997). There are a small number of popular tagsets for English. These include the 87-tag tagset used for the Brown corpus (Francis and Kucera, 1979), the small 45-tag Penn Treebank tagset (Marcus et al., 1993), the medium-sized 61 tag C5 tagset used by the Lancaster UCREL project's CLAWS (the Constituent Likelihood Automatic Word-tagging System) tagger to tag the British National Corpus (BNC) (Burnard, 2007), and others. Descriptions of tagsets are extracted from Jurasfky and Martin (2009). Here is an example of a tagged sentence from the Penn Treebank version of the Brown corpus.[9]

4.3 *The*/DT *grand*/JJ *jury*/NN *commented*/VBD *on*/IN *a*/DT *number*/NN *of*/IN *other*/JJ *topics*/NNS.

The tags in the previous example can be illustrated, as pointed out by Jurasfky and Martin (ibid.), as follows:

DT → Determiner

JJ → Adjective

NN → Noun, singular or mass

NNS → Noun, plural

VBD → Verb, past tense

IN → Preposition.

[9] Tags are represented here after each word, following a slash, but they can also be represented in various other ways.

The above tags contain not only information about parts-of-speech, but also about inflectional properties. This led some scholars to consider ‘morphosyntactic tags’ as a more adequate name than ‘part-of-speech tags’ (Voutilainen, 1999a).

Generally, the choice of the appropriate tag for a word depends not only on the word itself; the context is also important. This means that a word by itself is often ambiguous: without a linguistic context, there is no way of knowing which of the alternative tags should be assigned. For instance, the English word *round* could be a preposition, adverb, noun, verb or adjective. It can only be unambiguously analyzed in a linguistic context.

4.4 It came round the corner.

The word *round* in sentence 4.4 is analyzed as a preposition. What makes designing accurate taggers a difficult task is mainly the question of how to model the linguistic context of homographs like *round* so fully and accurately that the contextually correct analysis can be predicted automatically (Voutilainen, 1999a).

4.3.2 POS Tagging Approaches

In the last decade, tagging has been one of the most interesting problems in NLP (Tlili-Guiassa, 2006). A number of techniques have been proposed for automatic tagging. These techniques can be classified into three main groups:

- Rule-based Tagging

Rule-based tagging was used by Greene and Rubin in 1970 to tag the Brown corpus (Greene and Rubin, 1971). This tagger (called TAGGIT) used a set of rules to select the appropriate tag for each word in a given text. It achieved an accuracy of 77%. More recently, interest in rule-based tagging has emerged again with Brill’s tagger that achieved an accuracy of 96% (Brill, 1992). Later on, the accuracy of this tagger was improved to 97.2% (Brill, 1994). Generally speaking, these rule-based systems used lexicons that gave all possible analyses to some of the input words. Heuristic rules, which rely on affix-like letter sequences at word-boundaries, capitalization and other graphemic cues about word category, were used to analyze those words that are not represented in the lexicons. Those words that are not analyzed by the pattern rules were given several open-class analyses as alternatives (noun, verb, adjective readings). Then linguistic rules were used to resolve ambiguity. Such linguistic rules eliminate alternative analyses on the basis of the local context (e.g. two words to the left and to the right of the ambiguous word). For example, a rule might discard a verb reading from an ambiguous word if the preceding word is an unambiguous article. The tags that remained intact after the application of the linguistic rules were the correct analyses of the input words (Voutilainen, 1999b). The first stage of our POS tagger for Arabic is based on a set of morphological rules for handling roots and affixes to assign tags for a diacritized Arabic text.

- Stochastic Tagging

In the 1980s, interest passed to probabilistic taggers. This type of tagging used Hidden Markov Models (HMM) to select the appropriate tag. Such taggers include CLAWS, which was developed at Lancaster University, and achieved an accuracy of 97% (Garside and Smith, 1997) and the Xerox tagger, which achieved an accuracy of 96% (Cutting et al., 1992). The intuition behind stochastic tagging is a simple generalization of the ‘pick the most likely tag for this word’ approach (Jurafsky and Martin, 2009). In our work to build the Arabic POS tagger we use probabilistic techniques within its general framework to assign tags for undiacritized Arabic. These techniques are based on Bayes theorem and maximum likelihood estimation. Remarkably, we tried using HMM, but it gave us lower accuracy.

- Transformation-based Tagging

As of 1990s a new approach called transformation-based tagging emerged. It is a combination of both rule-based and statistical techniques. Transformation-based tagging, sometimes called Brill tagging, is an instance of the Transformation-Based Learning (TBL) approach to machine learning. As a matter of fact, both Brill’s tagger and CLAWS are in essence a combination of both techniques (Jurafsky and Martin, 2009). Brill’s tagger achieved an accuracy of 97.2% (Brill, 1995). As far as our Arabic tagger is concerned, it uses a combination of rule-based, transformation-based and probabilistic techniques.

POS tagging for Arabic has been an active topic of research in recent years. However, the application of machine learning methods to Arabic POS tagging appears to be somewhat limited and recent (Marsi et al, 2005).

4.3.3 Challenges for Arabic POS Tagging

Tagging undiacritized Arabic text with parts-of-speech is not an easy task. This is because the absence of diacritics results in huge ambiguity as far as words are concerned. This lexical ambiguity might be due to a number of reasons. We will discuss two important reasons that represent a challenge for any Arabic POS tagger.

- Homographic ambiguity: homographic words have the same orthographic form but different pronunciations and meanings (Jackson, 1988). These homographs usually belong to more than one part of speech, which represents hurdles in the way of POS taggers. For example, the word *bow* could be either a verb meaning "to bend" or a noun meaning "the front section of a ship". Many words are homographic in unvowelized Arabic. The following table shows some homographs in Arabic with different POS categories, which consequently poses a challenge for the task underway.

Homograph	**Meanings**	**POS Category**	**Gloss**
قدم *qdm*	قَدِمَ *qadima*	verb	to arrive from
	قَدَّمَ *qad~ama*	verb	to introduce
	قَدَمْ *qadamo*	noun	foot
ذهب **hb*	ذَهَبَ **ahaba*	verb	to go
	ذَهَبٌ **ahabN*	noun	gold
أسد *Osd*	أَسُدُّ *Oasud~u*	verb	I block
	أَسَدْ *Oasado*	noun	lion

Table 4.6: Arabic homographs

The first two examples in the table are uninflected words, while the third homograph contains an inflected word in the imperfective tense as one of its meanings.

- Internal word structure ambiguity: a complex Arabic word could be segmented in different ways (Farghaly and Shaalan, 2009). Thus, a POS tagger has to determine the boundaries between segments or tokens to give each token its proper tag. This is best illustrated in the following table.

Complex Word	Possible Tokens			POS Category	Gloss
ولي *wly*	وَ *wa*	لِ *li*	ي *y*	conj. + prep. + pronoun	and for me
	وَلِي *waliy*			noun	a pious person favoured by God
بعقوبة *bEqwbp*	بِ *bi*	عُقُوبَة *Euquwbap*		prep. + noun	with the punishment of
	بَعْقُوبَة *baEoquwbap*			proper noun	a town in Iraq
كمال *kmAl*	كَ *ka*	مَال *maAl*		prep. + noun	as money
	كَمَال *kamaAl*			noun	perfection
	كَمَال *kamaAl*			proper noun	a person's name

Table 4.7: Arabic words with different segmentations

This word segmentation ambiguity is sometimes termed 'coincidental identity'. This occurs when clitics accidentally produce a word-form that is homographic with another full form word (Kamir et al., 2002; Attia, 2006). Tagging undiacritized Arabic is thus a much more difficult problem.

4.3.4 Review of Arabic POS Taggers

Various Arabic POS taggers have recently emerged. These taggers employ different techniques, which may be rule-based, statistical or hybrid. Khoja (2001a) has developed an Arabic tagger using a combination of statistical and rule-based techniques. She has compiled a tagset containing 131 tags, which is derived from traditional Arabic grammatical theory. She reported an overall accuracy of 86% (Khoja, 2003). Freeman's tagger (2001) is based on the Brill tagger and uses a machine learning technique. A tagset of 146 tags, based on that of Brown corpus for English, is used. Diab et al. (2004) use Support Vector Machine (SVM) method and the LDC's POS tagset, which consists of 24 tags. They report an accuracy of 95.5% for POS tagging. Other taggers have been developed using HMM, which takes into account the structure of the Arabic sentence. Al Shamsi and Guessoum (2006) have built one such tagger. This HMM-based POS tagger has achieved a state-of-the-art performance of 97% over 55 tags. Tlili-Guiassa (2006) has proposed a hybrid method of a rule-based and memory-based learning technique for tagging Arabic words. A tagset composed of that of Khoja tagger was used and a performance of 85% was reported. Similarly, Van den Bosch et al. (2007) have explored the application of memory-based learning to morphological analysis and POS tagging of Arabic. It should be noted that memory-based tagging is a machine learning technique that is based on the idea that words occurring in the same context will have the same POS tag. As far as POS tagging is concerned, they report an accuracy of 91%.

A new approach is explored by Alqrainy (2008) in his rule-based Arabic Morphosyntactic Tagger (AMT), where he uses pattern-based technique as well as lexical and contextual technique to POS tag a partially-vocalized Arabic corpus. The AMT system has achieved an average accuracy of 91 %. In addition, AlGahtani et al. (2009) applied the Transformation-Based Learning (TBL) to the task of tagging Arabic text. They used the reduced 24 tags of Arabic Penn Treebank, and reported an accuracy of 96.9%. The Arabic sentence structure is exploited by El Hadj et al. (2009) in their approach to POS tagging. Their tagger combines morphological analysis with HMM and relies on the Arabic sentence structure. They used a tagset of 13 tags and reported an accuracy of 96%. Most recently, Mohamed and Kübler (2010b) have presented two different methods for Arabic POS tagging using memory-based learning. The first method is concerned with assigning complete POS tags to whole words without segmentation, whereas the second one is a segmentation-based approach for which they have developed also a machine learning-based segmenter. They base their experiments on the Penn Arabic Treebank (ATB). The first whole word-based approach has surprisingly reached an accuracy of 94.74%, whereas the second segment-based approach has scored 93.47%. Notably, our approach to POS tagging utilizes the first whole word approach since we do not have a lexicon of open-class words. The overall accuracy of our POS tagger initially scored 95.8 % (Ramsay and Sabtan, 2009). However, when we extended the tagset the score decreased to 93.1% when we train and test on the same data set. But using 10-fold cross validation has decreased the score to 91.2%. The details of our POS tagger will be discussed in the following section. Broadly speaking, these taggers are not strictly comparable because the larger the tagset the lower the expected accuracy.

4.4 Arabic Lexicon-Free POS Tagger

It has been pointed out that tagging text with parts-of-speech turns out to be extremely useful for more complicated NLP tasks such as syntactic parsing and MT. Almost all MT approaches use POS tagging and parsing as preliminary steps. We present in this part our lexicon-free POS tagger for Arabic. The reasons for doing without a lexicon have been elaborated in the introductory chapter. We use a diacritized corpus in the first stage of the tagger, where we derive an initial rule-based tagged corpus that is used as a training corpus for a subsequent stage of transformation-based learning (TBL). Then we remove the diacritics from the corpus and use a combination of maximum likelihood estimates (MLE) using Bayes theorem and transition probabilities and TBL again to enhance the tagger. As noted above, the final tagger scores now 91.2 % after extending the tagset and doing 10-fold cross validation.

We present an approach to POS tagging for undiacritized Arabic text which avoids the need for a large training set of manually tagged material. We start with a diacritized corpus and a rule-based tagger (we refer to this initial tagger for diacritized text as ‘rule-based’ to indicate that it relies on hand-coded rules, unlike the final tagger which we use on the undiacritized corpus), and we use this to obtain an initial tagging. We then manually correct a portion of this training set, which is a much easier task than annotating it from scratch, and use a TBL tagger to improve the performance of the rule-based tagger, and we use this to generate a tagged undiacritized corpus (by tagging the diacritized one and then removing the diacritics), and we use this generated corpus as the training set for a combination of MLE and TBL tagging. The advantage of this approach is that it requires very little manual effort. The only manual intervention is in the correction of the original training set. This means that we can use it to obtain a tagger for a previously unseen type of text: all we need is a diacritized corpus from the genre, and we can produce a tagger with very little effort.

In the following sections we will discuss the tagset we have used in the tagger, our approach to Arabic tagging and finally the results we have obtained.

4.4.1 Arabic Tagset

It has been mentioned earlier that a tagset is simply a list of tags used for a given task of tagging. Tagsets can be large or small according to the task they are designed for. Sawalha and Atwell (2009; 2010) propose a fine-grained morphological features tagset, which can be used in developing morphological analyzers. We use a tagset that represents the main parts-of-speech without fine-grained morphological features. The tags used refer to the following parts of speech: *Noun, Verb, Auxiliary Verb, Exclamatory Verb, Particle, Pronoun, Relative Pronoun, Demonstrative, Preposition, Complementizer, Determiner, Conjunction, Number, Dhuw* and the *Question Particle*. We do not make further sub-classification for these classes. For example, we identify nouns without indicating whether they are in the singular or plural form. Similarly, we identify verbs without indicating whether they are in the perfect or imperfect tense.

The tagset that is described here follows the traditional Arabic grammar that has been used for centuries. As indicated earlier, Arabic grammarians traditionally analyze all Arabic words into three main parts-of-speech. These are noun, verb and particle. Then these parts-of-speech are further subdivided into more detailed parts-of-speech. Our tagset maintains the main parts-of-speech, i.e. noun, verb and particle, but with further subdivisions. In case of **nouns** we subdivide it into separate tags for *Noun, Pronoun, Relative Pronoun, Demonstrative, Determiner, Number* and the noun ذو **uw*. As for **verbs,** we distinguish between main verbs, auxiliary verbs and exclamatory verbs[10]. By **auxiliary** verbs we mean the verb كَانَ *kaAna* “was/were” (which is similar to auxiliaries in English) and its sisters. These verbs are connected with being or becoming and are called ناقصة *naAqiSap* “incomplete or defective”. They are so called because they are not syntactically complete without a following noun (semantically a subject) and another argument (semantically a predicate), which could be an NP, a VP or a PP. These verbs add tense or modality to sentences (Badawi et al., 2004). These auxiliary verbs can be used in two ways: (i) they can precede a main verb and in this case we tag them as **auxiliary**. (ii) they can be used as copulative verbs with a following nominal sentence (subject and predicate). In this case we tag them as **verb**. The auxiliary category includes also أفعال المقاربة *OafoEaAl AlmuqaArabap* “verbs of appropinquation or getting close”. These verbs are divided

[10] This name is used by Badawi et al. (2004).

into two types: (i) they may indicate simple proximity of the predicate such as كَادَ *kaAda* “almost”, أَوشَكَ *Oawo$aka* “about to” (ii) they may imply a hope of its occurrence such as عَسَى *EasaY* “may” (Wright, 1967). These verbs are either followed by independent verbs as with كَانَ *kaAna* and its sisters or subordinated أَنْ *Oan* clauses (Badawi et al., 2004; Hassan, 2007). As for the **exclamatory** verbs, they refer to the Arabic verbs نِعْمَ *niEoma* “how good/favorable” and بِئسَ *bi}osa* “how bad/miserable” (Badawi et al., 2004). In Arabic they are usually referred to as أفعال المدح والذم *OafoEaAlu AlmadHi wa Al*am~i* “verbs of praise and blame”. They are used as exclamations and are generally indeclinable, though the feminine forms نِعْمَتْ *niEomato* and بِئسَتْ *bi}osato* occur, especially in Classical Arabic (Wright, 1967). We have a separate tag for exclamatory verbs in our tagset. As for the part-of-speech *particle,* we subdivide it into more detailed parts-of-speech, adding other tags to the tag *particle*. These other tags are *Preposition*, *Complementizer* and *Conjunction* in addition to the *Question Particle* that is attached to closed-class or open class items. As for unknown words, we give it the tag *UN*. Thus, we have 16 different tags. Those tags cover both open-class and closed-class words. We will discuss both types in the following sections.

4.4.1.1 Open-Class Words

Open classes are those words that are continually coined or borrowed from other languages such as nouns and verbs (Jurafsky and Martin, 2009). In our tagger we do not use a lexicon for open class words. We use regular expressions to identify clitics and affixes in a given string to get its tag. We have divided open class words into two parts-of-speech. These are nouns and verbs.

1. **Nouns**: The noun has been described at the beginning of this chapter. It has been pointed out that the noun class in Arabic is subdivided into derivatives which are derived from verbs, other nouns, or particles and primitives which are not so derived. Derivatives include such classes as verbal nouns, e.g. قَتْل *qatol* “killing”, active and passive participles, e.g. قَاتِل *qaAtil* “killer” and مَقْتُول *maqotuwl* “killed” respectively, the elative, e.g. أفضل *OafoDal* “better”, diminutive, e.g. جبيل *jubayol* “hill”, relative noun, e.g. دمشقيٌّ *dima$oqiy~u* “born or living at Damascus”. These nouns could be further sub-classified by number, gender and case. Thus, in the tagger nouns include what, in

traditional European grammatical theory, would be classified as adjectives and participles. All the noun subclasses that are tagged as **noun** will be discussed below when we discuss our approach.

2. **Verbs**: The definition of verbs in Arabic has been also given earlier in this chapter. It has been indicated that verbs can be subdivided into perfect, imperfect, and imperative. The perfect indicates completed action or elapsed events, corresponding roughly to the English simple past and present or past perfect, e.g. كتبَ *kataba* “wrote”. As for the imperfect, it generally indicates an incomplete action, continuous or habitual, with the exact time reference depending on context, e.g. يكتبُ *yakotubu* “writes”. The letter سَ *sa* or word سَوْفَ *sawofa* may be used to indicate the future tense, such as سَيُسافِر *sayusaAfir* “he will travel” and سَوْفَ أزُورُكَ *sawofa Oazuwruk* “I will visit you” (Fischer, 2002; Badawi et al., 2004). Imperfect verbs may also be classified with regard to mood into declarative, subjunctive and jussive. Each mood has a different case marker. Thus, verbs in the declarative mood have the same case marker as nominative nouns, while those in the subjunctive mood have the same case marker as accusative nouns (Wright, 1967). With regard to jussive mood, it is denoted by the absence of any vowel or rather a zero vowel called ‘sukun’. Finally the imperative denotes direct commands or requests, e.g. اكْتُبْ *Akotub* “write”. The imperative indicates an action whose time is directly linked to the time of speaking (Hijazy and Yusof, 1999). Further subdivisions of the verb class are possible using number, person and gender. Transitivity is also another factor under which verbs can be sub-classified. For example, transitive verbs take objects, whereas intransitive verbs do not. However, in our tagger we tag all types of verbs as **verb** without any sub-classification. This coarse-grained tagging is suitable for our basic task of lexical selection, as will be shown in chapter 6.

4.4.1.2 Closed-Class Words

Closed classes are those words that have relatively fixed membership. For example, prepositions are a closed class because there is a fixed set of them in most languages. New prepositions are rarely coined (Jurafsky and Martin, 2009). We have classified the closed-class items into a number of categories. These are *pronouns*, (which may be free pronouns or clitic pronouns), *relative pronouns*, *demonstratives*, *prepositions*, *complementizers*, *particles*, *determiners*, *conjunctions*, *numbers*, different forms of the noun *dhuw* “possessing”, *auxiliary verbs*, *exclamatory verbs* and the *question particle*. Here is the classification of closed-class words with some examples in each category:

1- **Pronouns**, which may be free pronouns, e.g. أنا *OanaA* “I”, أنتَ *Oanta*, “you”, نحنُ *naHonu* “we”, هُوَ *huwa* “he”, هِيَ *hiya* “she” or clitic pronouns, e.g. كَ *ka* “you” (as an object pronoun) or “your” (as a possessive pronoun), and ـهُ *hu* “him” (as an object pronoun) or “his” (as a possessive pronoun).

2- **Relative Pronouns**, which introduce relative clauses. These special pronouns have the meaning of “who”, “whom” or “what” according to the context in which they are used, e.g. الَّذِي *Al~a*iy* (masc. sing.), الَّتِي *Al~atiy* (fem. sing.), and الَّذِينَ *Al~a*iyna* (masc. pl.), etc. Relative pronouns are linked to relative clauses by a pronoun which is called in Arabic العائد *AlEaA}id* “the linking pronoun” . This pronoun should be in full agreement with the relative pronoun. However, this linking pronoun can be omitted when it can be retrieved through the context (Omar et al., 1994).

3- **Demonstratives**: e.g. هَذَا *ha*aA* (masc. sing.) “this”, هَذِهِ *ha*ihi* (fem. sing.) “this”, ذَلِكَ **alika* “that”, and هَؤُلاءِ *haWulA'i* “those”.

4- **Prepositions**: e.g. بِ *bi* “with”, في *fiy* “in”, عَلَى *EalaY* “on”, and إِلَى *IilaY* “to”.

5- **Complementizers**: e.g. لِ *li* “in order to”, and لَ *la* “indeed/surely/verily”. Items of this kind are also often called subordinating conjunctions.

6- **Particles**: e.g. إِنَّ *Iin~a* “surely”, لقد *laqad* “indeed”, لا *lAa* “no/not”, and ما *maA* “no/not”.

7- **Conjunctions**: e.g. وَ *wa* “and”, فَ *fa* “then”, أو *Oawo* “or”, and ثُمَّ *vum~a* “then”.

8- **Determiners**, which are quantifiers, e.g. كلُّ *kul~u* “all”, بَعْضُ *baEoDu* “some” and أيُّ *Oay~u* “any”.

9- **Auxiliary Verbs**: كانَ *kaAna* “was/were”, أصْبَحَ *OaSobaHa* “became”, مازَالَ *maAzaAla* “still”, كادَ *kaAda* “almost”.

10- **Exclamatory Verbs**: نِعْمَ *niEoma* “how good/favorable”, بِئسَ *bi}osa* “how bad/miserable”.

11- **Numbers**, which include cardinal numbers, e.g. ثلاثة *valaAvapu* “three” and ordinal numbers, e.g. ثالثُ *vaAlivu* “third”.

12- The **noun** ذو **uw* “possessing” or “characterized by”. It is used in the definite form and has different forms according to gender and number differences. This noun is always used in a construct, which acts as a compound adjective (Mace, 1998). We give it the tag **DHUW**.

13- **The Question Particle** أ *Oa* “is it true that”.

It is worth noting that we give the tag **UN** for unknown words in the corpus. The following table shows our basic tagset.

POS Tag	**Description**	**Examples**
NN	This includes all types of nouns, with their various forms due to number, gender or person differences. It also includes proper nouns.	كتاب *kitaAb* “a book”, كاتب *kaAtib* “a writer”, عظيم *EaZiym* “great”, موسى *muwsaY* “Moses”.
VV	This tag applies to all verbs, regardless of tense, mood, number, gender or person.	يقول *yaquwl* “(he) says”, قالوا *qaAluwA* “(they) said”.
PRO	Pronouns may be free or clitic.	هُوَ *huwa* “he”, ـهُ *hu* “him/his”.
RELPRO	Relative pronouns, with all various forms due to gender and person differences.	الذي *Al~a*iy*, التي *Al~atiy*, الذين *Al~a*iyna*, من *man* “who”, ما *maA* “what”.
DEMO	All forms of demonstratives.	هَذا *ha*aA* “this”, ذَلِكَ **alika* “that”, هؤلاء *haWulA'i* “those”.
PREP	This applies to both cliticized prepositions that are attached to nouns and free prepositions.	بِ *bi* “with”, لِ *li* “to”, كَ *ka* “as”, في *fiy* “in”, إلى *IilaY* “to”.
COMP	This tag refers to complementizers, i.e. certain	لِ *li* “in order to” or لَ *la* which is used for emphasis

	prepositions that precede verbs.	“surely/verily”.
PART	All types of particles except the question particle, which has a separate tag.	إنَّ _Iin~a_ “surely”, لا _laA_ “no/not”.
CONJ	This applies to both cliticized conjunctions that are attached to words and free conjunctions.	وَ _wa_ “and”, فَ _fa_ “then”, ثُمَّ _vum~a_ “then”.
DET	Determiners. This tag is basically used for quantifiers.	كلُّ _kul~u_ “all”, بَعْضُ _baEoDu_ “some”
AUX	This refers to the verbs that are similar to auxiliaries and modals in English.	كانَ _kaAna_ “was”, أصْبَحَ _OaSobaHa_ “became”, عَسَى _EasaY_ “may”.
EXVV	This tag is used to refer to exclamatory verbs. In Arabic they are usually referred to as verbs of ‘praise and blame’.	نِعْمَ _niEoma_ “how good”, بِئسَ _bi}osa_ “how bad”.
NUM	This tag refers to numerals, which include both cardinal and ordinal numbers.	ثلاثةٌ _valaAvapu_ “three”, ثالثُ _vaAlivu_ “third”.
DHUW	This refers to all forms of the noun _*uw_.	ذُو _*uw_, ذَاتُ _*aAtu_, “possessing”.
QPART	This is the question particle that is attached at the beginning of closed class or open class items.	أ _Oa_ “is it true that”
UN	This tag is used to refer to “unknown words”.	Any word which the tagger cannot identify is tagged as _unknown_.

Table 4.8: Our basic Arabic tagset

Our tagset is not a fine-grained one. We label all types of nouns as **NN**, and all types of main verbs as **VV**. We do not make any distinction due to number, gender or person differences.

Generally speaking, there are two different methods for Arabic POS tagging (Mohamed and Kübler, 2010a; 2010b).

(i) Lexeme-based tagging, where POS tags are assigned to lexemes not whole words. Mohamed and Kübler (ibid.) and others who take this approach refer to the process of splitting tokens into lexemes as 'segmentation'. In discussion of their work we will follow their terminology. Thus, a word such as وبكتبهم *wbktbhm* "and with their books" is segmented into tokens with tags for each token as follows.

Segments	**POS Tags**
و *w*	CONJ
ب *b*	PREP
كتب *ktb*	NN
هم *hm*	PRO

Table 4.9: Segment-based POS tags

We can notice that this word form consists of a conjunction, a preposition, a noun and a possessive pronoun. Sometimes a distinction is made between segmentation and tokenization as far as Arabic is concerned. In the previous example there is no difference between both segmentation and tokenization. However, in a word such as وسيكتبونها *wsyktbwnhA* "and they will write it", both processes are treated differently. Thus, this word can be segmented as w+s+y+ktb+wn+hA but tokenized as w+syktbwn+hA. (Some authors also treat the future marker {s} as a separate lexeme (Diab, 2009)). Here the boundaries between segments or tokens are demarcated by the + signs. The word is thus segmented into 6 segments but tokenized into 3 tokens. Therefore, segmentation is a method to demarcate the boundaries between all the word parts, whereas tokenization delimits boundaries between syntactically independent units in a word.

(ii) Whole word tagging, where complete POS tags are assigned to whole words without word segmentation or tokenization.

In this method complex tags are given for words with clitics. Thus, the word وبكتبهم *wbktbhm* can be tagged as follows.

Whole Word	**POS Tags**
وبكتبهم *wbktbhm*	CONJ+PREP+NN+PRO

Table 4.10: Word-based POS tags

Here the + sign is used to mark the boundaries between tags. We follow this word-based approach to POS tagging, where words as a whole are tagged. This has been done because we do not use a lexicon of words that could have enabled us to do tokenization. In fact, doing segmentation or tokenization without using a lexicon is not feasible, as it will not be possible to know the boundaries between segments or tokens.

We have described our basic tagset which consists of 16 different tags. This basic tagset is used to describe words that have no clitics as well as words with clitics. The final tagset contains 96 distinct tags, ranging from simple tags like **NN** or **VV** to complex tags like **QPART+CONJ+VV+PRO** or **QPART+PREP+NN+PRO** for words with multiple clitics. This tagset, which is generated by the tagger, is given in Appendix B. In fact, this larger set of tags allows more scope for errors. However, most of these errors are about clitics, which do not matter for our overall goal.

4.4.2 Our Approach to Arabic POS Tagging

We propose an approach to tagging undiacritized Arabic text which exploits the fact that diacritized text is fairly easy to tag. This approach avoids the need for a large manually tagged corpus or for a lexicon of open-class words, though it does depend on the existence of a diacritized corpus. To do this, we use machine learning techniques. As with any machine learning task, there is a training phase during which we gather information from some dataset, and then this information is used to carry out the actual task. Figures (4.3 & 4.4) give a very general view of these two activities.

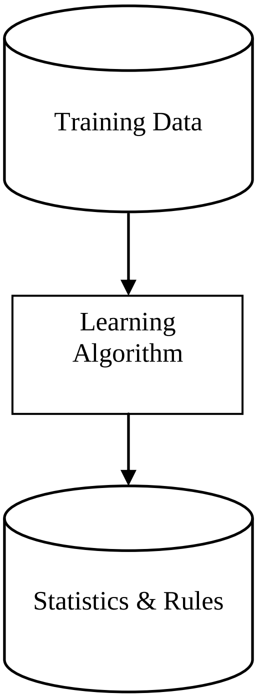


Figure 4.3: Training phase in machine learning

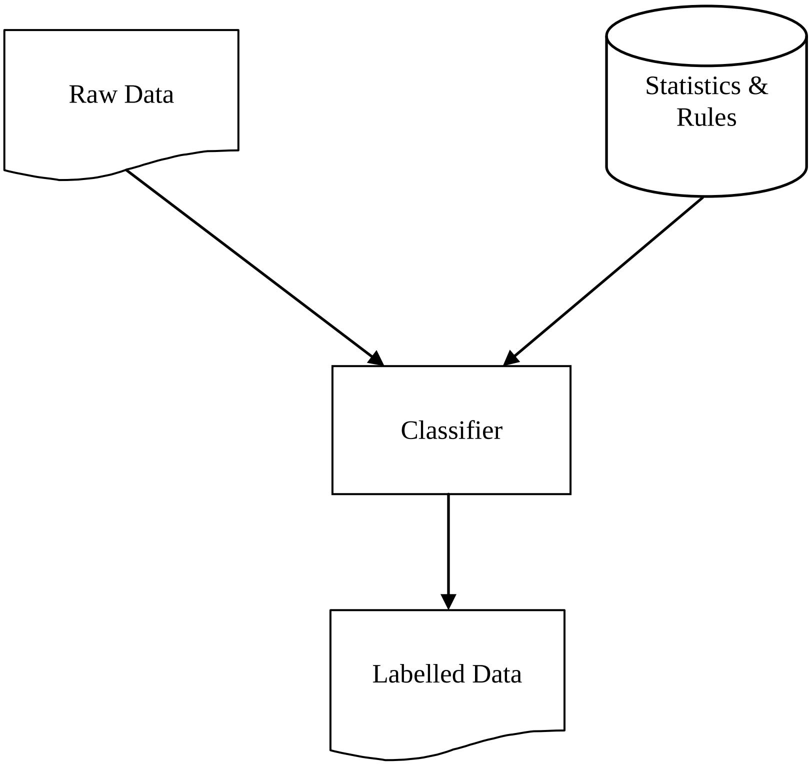


Figure 4.4: Application phase in machine learning

In any practical case how training is carried out and how the information is used for the actual task will vary. The steps that we go through in the training and application phases are described below. Firstly, we describe the steps we have taken to obtain the tagged data as shown in stage (1), which we use to train the final tagger, as shown in stage (2).

1. Obtaining a tagged undiacritized version of the corpus.
    - (1.1) Use a rule-based tagger to tag the original diacritized corpus. We will call the rule-based tagger T_{RB} and the original diacritized corpus C_D.
    - (1.2) Manually correct a portion of this: correcting a tagset by hand is easier and less time-consuming than manually creating one from scratch. We will call this corrected diacritized section of the original corpus GS_D (Diacritized Gold Standard).
    - (1.3) Apply a transformation-based tagger to the corrected tagset to learn rules that will compensate for errors in the tags assigned by the rule-based tagger. Most of these ‘errors’ will, in fact, be cases where the rule-based tagger has left the tag underspecified - there are comparatively few cases where the rule-based tagger assigns a single wrong tag, but it is quite often unable to choose between different cases, and hence assigns multiple tags. We will call the combination of the rule-based tagger and the corrective rules obtained by the transformation-based one $T_{RB+TBL.}$
    - (1.4) Use the rule-based tagger and the corrective rules obtained at step (1.3) to tag the entire diacritized corpus, *and then remove the diacritics*. This produces a tagged undiacritized corpus with very little manual effort. We will call this undiacritized corpus C_U, and we also obtain an undiacritized version GS_U of the Gold Standard simply by removing the diacritics.
2. Training the final tagger using the results of stage 1.
    - (2.1) Develop a Bayesian tagger T_B, based on the conditional frequencies of tags relative to the first and last three letters of the written form. This information provides a combination of the remaining inflectional material and the raw probabilities of particular words. T_B is then supplemented by considering the parts of speech assigned to the preceding and following words (maximum likelihood tagger, T_{ML}).
    - (2.2) We again use a transformation-based tagger to improve the situation, producing our final tagger T_{ML+TBL}. Note that the rules obtained at this

stage may not be the same as those obtained at (1.3). Transformation-based taggers derive rules which patch errors introduced by the previous stage, and there is no reason to suppose that the errors introduced at (2.1) are the same as those introduced at (1.1). They are, indeed, bound to be different: most of the errors introduced by the rule-based tagger involve ambiguity. The Bayesian tagger never produces ambiguous tags - it cannot- but it does make suggestions which are just wrong.

Figures (4.5 and 4.6) illustrate these two stages, which fill in the gaps in figures (4.3) above

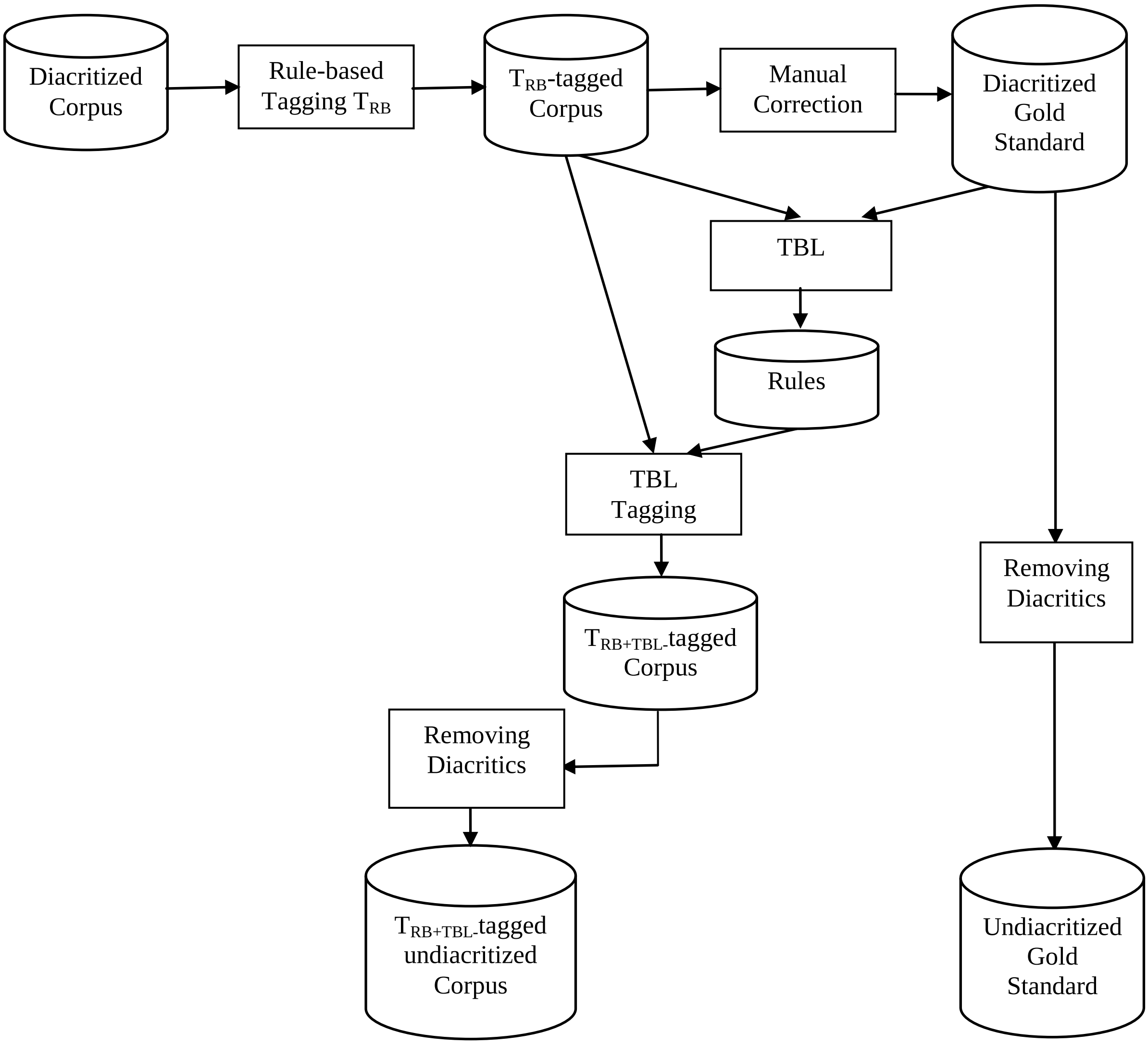

Diacritized Corpus
Rule-based Tagging T_{RB}
T_{RB}-tagged Corpus
Manual Correction
Diacritized Gold Standard
TBL
Rules
TBL Tagging
T_{RB+TBL}-tagged Corpus
Removing Diacritics
Removing Diacritics
T_{RB+TBL}-tagged undiacritized Corpus
Undiacritized Gold Standard

Figure 4.5: Stage 1: Obtaining undiacritized training data

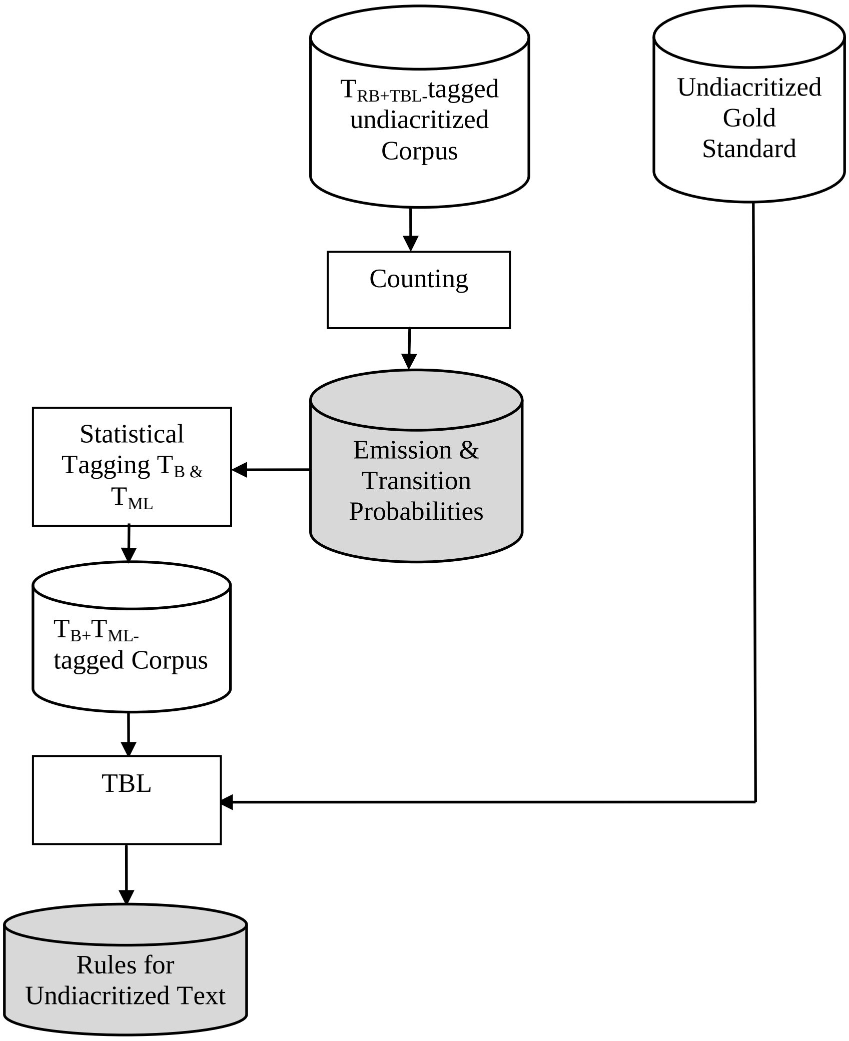

$T_{RB+TBL-}$tagged undiacritized Corpus
Undiacritized Gold Standard
Counting
Emission & Transition Probabilities
Statistical Tagging $T_{B\ \&}$ T_{ML}
$T_{B+}T_{ML-}$ tagged Corpus
TBL
Rules for Undiacritized Text

Figure 4.6: Stage 2: Training the final tagger

In figure (4.7) below we show the application phase of the tagger, which fills in the gaps in figure (4.4) above.

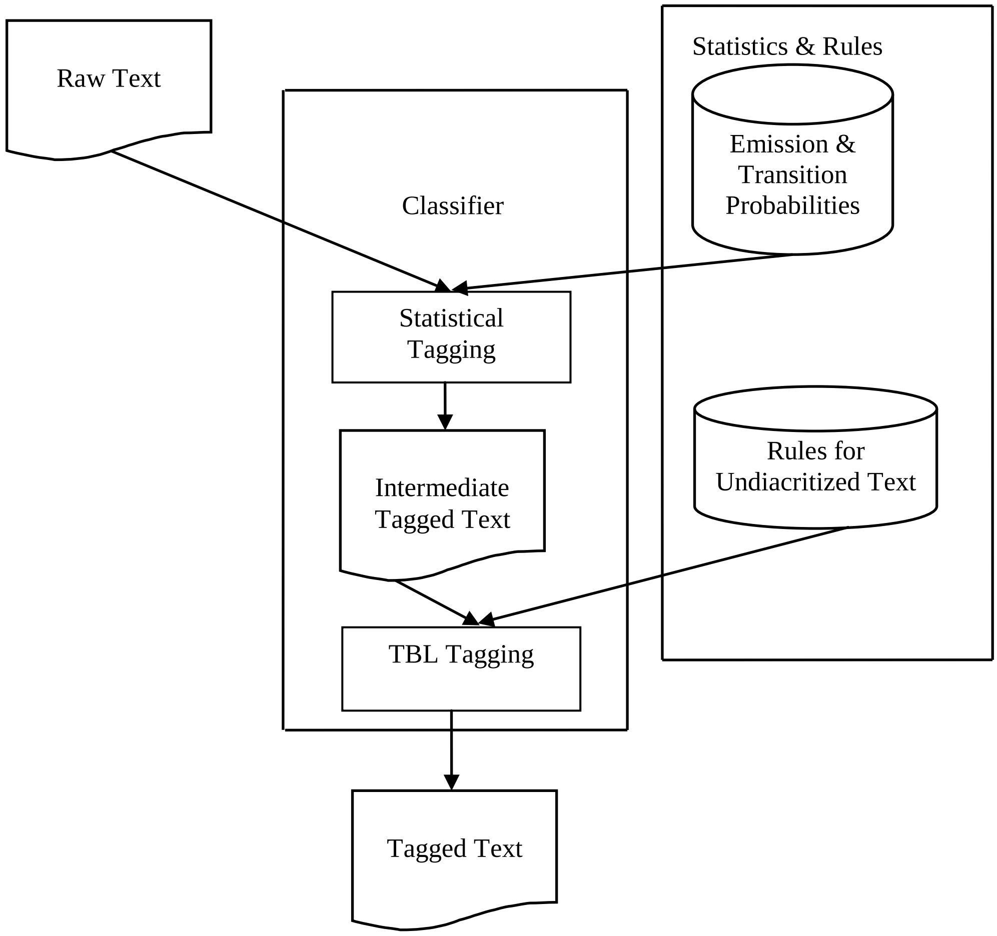


Figure 4.7: Application phase of the Arabic POS tagger

The great advantage of this approach is that it requires very little manual intervention, and that it also makes no use of a lexicon of open-class words. It thus involves very few resources. The only place where manual tagging has to take place is in the correction of the tags in the diacritized Gold Standard, and we do not have to construct a lexicon, which would also require considerable effort. The accuracy of the final tagger on undiacritized text is almost as good as that of the combination of the rule-based tagger and corrective rules on diacritized text, suggesting that the method is fairly robust, so that if the rule-based tagger (which is not perfect) were replaced by something which is nearly perfect then the final Bayesian+TBL tagger would also be near perfect. We will describe each stage of the tagger in detail below.

4.4.2.1 Rule-Based Tagging

This phase is applied to the diacritized text of the Qur'anic corpus. It has been illustrated that Arabic words are either open-class or closed-class. We do not use a lexicon of open-class words. As for closed-class words, we use a small lexicon for these items. The algorithm for rule-based tagging is described in the following section.

4.4.2.1.1 Tagging Algorithm

The rule-based tagger contains a set of patterns, which can be matched with the start or end of a word. This applies to both closed-class and open-class words. It is worth noting that there are some special cases of closed-class items that change their shape when attached to other closed-class items. For example, when the preposition مِن *min* "from" is attached to the relative pronoun مَا *maA* "what" it takes the form مِمَّا *mim~aA*. Since this form is not the concatenated form *minmaA* it is not easy for the tagger to identify it. Therefore, we have a number of patterns to match such special cases to be identified by the tagger. Here is an example of a special case, as written in the tagger.

```
SpecialCases = [("\\Amim~aA\\Z", ['PREP+RELPRO'])]
```

Accordingly, if a word is to be tagged, T_{RB} starts with matching such a word against special words patterns and tags it if it is matched. Otherwise, it moves to the closed-class words and tries to match the word in question against closed-class patterns. These patterns allow for complex items, which are split into their constituents when they are matched. So, the closed-class analyzer starts at the beginning of a word and sees if it matches one of the closed-class categories. If it matches only one category, it is given a single tag, e.g. فِي *fiy* "in" **PREP**. If it is, however, a complex word consisting of more than one closed-class category, the analyzer goes through all constituents of the word, seeing if the first part matches a closed-class category and tags it accordingly. It then moves on to match the remainder of the word, giving multiple tags to other parts. For instance, a word may consist of a preposition preceded by a conjunction and followed by a pronoun as one string such as وَفِيهِ *wafiyhi* "and in it". In this case, it is given the complex tag

CONJ+PREP+PRO. The following pattern for conjunctions is an example of the small closed-class dictionary.

```
Conjunctions = ["wa", "fa", "vum~a", "Oawo"]
```

If the word in question is not matched in the small closed-class dictionary, it is matched against a number of patterns for open-class words. We use regular expressions (REs) to identify roots, affixes and clitics in a given word to get its POS tag. Regular expressions can be easily efficiently matched with a string to determine whether it is potentially a verb, and if so to split it into the root, its inflectional affixes, and other cliticized items such as conjunctions and cliticized prepositions and pronouns. We will mention such patterns in passing to illustrate the tagging algorithm, but will discuss them in detail later on. These patterns are complex ones that comprise proclitics, prefixes, roots, suffixes and enclitics in order. The set of affixes that come before or after a root may be common to both nouns and verbs or may be peculiar to either class. We, thus, sub-classify these affixes according to whether they are attached to nouns, verbs or both. The following pattern, for instance, is used to indicate that a given verb may optionally contain a number of affixes and clitics besides the root.

```
Qmarker+conjs+Vproclitics+Vprefixes+vroot+Vsuffixes+Venclitics
```

As shown in this complex pattern, its components are written in order. In fact, each component is itself a pattern. So, we start with matching proclitics at the beginning of words. These proclitics are matched as they come in order in a word. Thus, the first proclitic to be matched is the question particle أ *Oa* “is it true that”, which may come before a verbal root. After the question particle is identified, it is given a separate tag **QPART** before handling the remainder of the word. The question particle pattern is then followed by patterns for cliticized conjunctions such as وَ *wa* “and” and فَ *fa* “then”. They are also given a separate tag **CONJ** before the remainder of the word is given its proper tag. Thirdly come patterns for proclitic complementizers such as لِ *li* “to” or emphatic لَ *la* “surely”. They are tagged as **COMP**. Having finished proclitics, we start to match prefixes. The following pattern describes a range of tense marking prefixes, and then marks this pattern as being optional.

```
Vprefixes = "(((O|y|t|n))(a|u)|sa|A))?"
```

This pattern will match a variety of verbal prefixes. There are items which are not verbs but whose initial letters will match this pattern: all we can hope for is that the pattern will make sensible suggestions. Some verbal tense markers may be used in the active or passive voice. Thus, the pattern includes ordered groups to indicate both the active and passive form of a tense marker. In the above-mentioned pattern the prefixes *Oa*, *ya*, *ta*, and *na* are tense markers attached to imperfective verbs in the active voice. However, when they are used in the passive voice for some verbs, they are normally written as *Ou*, *yu*, *tu*, *nu*, e.g. يُكْتَب *yukotabo* “is written”. In actual fact, some verbs in the active voice start with the same tense markers for the passive voice, as in يُناقِش *yunaAqi$* “(he) discusses”. The passive for such a verb is يُنَاقَش *yunaAqa$* “is discussed”. In any case, we do not have separate tags for active and passive verbs. We tag both types as **VV**. By and large, these tense markers distinguish verbs from other POS categories, which is all we need for our task. As for the prefix *sa*, it is a future tense marker, as in سَيَكْتُب *sayakotub* “(he) will write”. The final tense marker *A* is used with verbs in the imperative, e.g. اكْتُب *Akotub* “write”. The question mark at the end of the pattern signifies that these tense markers are optional and only one of them may occur. Notably, verbal or nominal prefixes are used as distinguishing markers to identify whether a given word is a verb or a noun, without having a separate tag like proclitics. We move on with the pattern in question to match a verbal root. We have patterns for tri-consonantal roots only in the rule-based tagger. The reason for focusing on triliteral roots will be discussed below. The root patterns cover both strong and weak roots as will be discussed later. The roots, as pointed out earlier, combine with a vowel melody to form stems. When a verbal stem is identified, it is tagged as **VV**. After matching roots, the tagger searches for verbal suffixes. Verbal suffixes include agreement and tense markers, as shown in the following pattern.

```
Vsuffixes = "(at|t(a|i|u)|iy|naA|Ani|uwA|uwna|na|tum|tun~a)?"
```

The word in question is matched against the above-mentioned pattern to see if it contains verbal suffixes, e.g. كَتَبْتُم *katabotum* “you (pl.) wrote”. Like prefixes, no separate tag is given for suffixes. We finally end the pattern matching process with identifying enclitic patterns. In case of verbs, enclitics are object pronouns that are attached to the end of verbs. The pattern for enclitics is shown below.

```
Venclitics = "(hu|haA|hum(aA)|hun~a|k(a|i)|kum(aA))|kun~a|naA|niy)*"
```

This enclitics pattern includes object pronouns that are distinguished according to person, number and gender. The pattern ends with the star * to indicate that zero or more of enclitics may occur with verbs. In actual fact, normally only one object pronoun is attached to verbs. However, there are some cases, which are more common in CA than MSA, where two object pronouns come following each other at the end of ditransitive verbs. The first one is the indirect object and the second one is the direct object, e.g. أعْطَيْتُمونيها *OaEoTayotumuwniyhA* "you gave it to me". That is why we used a star symbol instead of a query to capture such cases. The enclitic category is given a separate tag as **PRO**.

As a simple example for a complex open-class word, the word وَليَكْتُبَهُم *waliyakotubahum* "and to write them" is matched against the patterns we have discussed above and is thus tagged by the rule-based tagger as **CONJ+COMP+VV+PRO**. Finally, however, if the word in question is not matched against closed-class or open-class patterns, it is tagged as **UN**, i.e. 'unknown'.

Having described the main algorithm for Arabic rule-based tagging, we are going to describe in detail the way we have used regular expressions to compile patterns for both closed-class and open-class words.

4.4.2.1.2 Handling Closed-Class Words

The different closed-class categories have been discussed before under the section for Arabic tagset. Nonetheless, we will discuss in the following lines how we use REs to compile patterns to identify a given closed-class word and how it is decomposed into a number of components if it is a complex word. We will shed light on one of the various closed-class categories to grasp how these categories are identified by the tagger. The following pattern contains some of the category of prepositions that are matched by the closed-class analyzer.

```
PrepPatterns = ["min(o|a)","bi","fiy","EalaYa","IilaY", "Ean(o|i)",
"maEa", " la((?!mo|wo|Eal~a|n|m~aA|A|da|du|qado|kin))"]
```

As can be seen in the prepositions pattern, some of its strings contain the metacharacter ( ) to indicate that some words can have different diacritics at the end. Thus, the first string in the pattern means that the preposition *min* “from” can have different diacritic marks at the end. It can be written as *min, mino* or *mina*. Another point to be noticed is that the final string in the pattern which refers to the preposition *la* “surely” is followed by a negative lookahead assertion (?!...). This has been done to prevent other strings that start with the same two letters, e.g. *lamo* “not”, from being matched by the regular expression. This is because the string *lamo* is written in the particles category. So, the negative lookahead assertion means that if the contained regular expression does not match at the current position in the string, try the rest of the pattern. However, if it matches, the whole pattern will fail.

It is not always the case that Arabic closed-class items are used in isolation. In fact, one of these items can be attached to one or two others, resulting in a complex word. Thus, as noted above, prepositions can be preceded by conjunctions and followed by pronouns in one string. The tagger decomposes a given closed-class item to identify its different components. The following Python function illustrates part of this procedure.

```
def closedClassAnalyzer(string):
    string.strip ()
    m = PrepPatterns.match(string)
    if m:
        r = m.group('remainder')
        if r == "":
            return "PREP"
        if PronounPatterns.match(r):
            return "PREP+PRO"
```

As can be shown in the previous code, the closed-class analyzer goes through a given string and sees if it is found in one of the categories in the closed-class dictionary. We have referred here to handling prepositions. So, if the string in question is a single preposition, it is tagged as **PREP**. If, however, the string is a preposition attached to a pronoun, it is tagged as **PREP+PRO**. The same procedure is applied to other categories to see if a given preposition is attached to any other closed-class item. What is done with prepositions is also done for other closed-class categories in the closed-class analyzer.

4.4.2.1.3 Handling Open-Class Words

As noted above, concerning open-class words, i.e. nouns and verbs, we have a number of patterns for matching words. These patterns are for roots as well as affixes and clitics. We firstly set patterns for nominal and verbal roots. Then we set patterns for affixes and clitics. A root is an ordered triple or quadruple of consonants which are called radicals. In most cases roots are either triliteral such as كتب *ktb* “to write” or quadriliteral such as زلزل *zlzl* “to shake”. Roots, as pointed out earlier, are said to interdigitate with patterns to form stems. The pattern is a template of syllables, the consonants of which are that of the triliteral or quadriliteral root. Thus, stems are formed by a derivational combination of a root morpheme and a vowel melody; the two are arranged according to canonical patterns. Thus, the stem كَتَبَ *kataba* “he wrote” is composed of the morpheme *ktb* “notion of writing” and the vowel melody morpheme ‘a-a’. The two are coordinated according to the pattern CVCVC.

Broadly speaking, we focus on the triliteral roots as they are the most common in the Arabic language. As for non-triliteral roots, we leave them to be dealt with in the subsequent phase of TBL. The reason for leaving out quadriliteral roots is that, owing to the fact that we do not use a lexicon of words, setting patterns for such quadriliteral roots results in more ambiguous tags in this rule-based stage of the tagger. This is because it regards prefixes or suffixes attached to tri-consonantal roots as a main part of a quadri-consonantal root, and some words are thus given an ambiguous tag. Thus, a word such as الصَّلاة *AlS~alApa* “the prayer” is currently tagged correctly by the rule-based tagger as **NN**. But if we introduce a pattern for quadriliteral roots, it will be ambiguously tagged as **NN-VV**. So, in this case it identifies the consonant ل *l* in the definite article ال *Al* “the” as a main part of a quadri-consonantal root and not a prefix. In this way the nominal marker, the definite article, is excluded and thus the pattern can apply to both nouns and verbs. That is why we excluded quadriliteral roots from this rule-based stage, especially as they are also less common than triliteral roots.

Triliteral roots are combined with a vowel melody to form a stem with the pattern CVCVC. In the rule-based stage of the tagger we write C to mean a consonant, while V is used to refer either to a short or long vowel.

Stems are either nominal or verbal. **Nouns** include such patterns as CVCVVC, e.g. كِتَاب *kitaAb* “book”, CVVCVC, e.g. كَاتِب *kaAtib* “writer”, CVCVC, e.g. عنب

Einab “grapes”, CVCCV, e.g. رَبٍّ *rab~i* “lord”, etc. Some nominal patterns include two consonants as CVVC, e.g. نَار *naAr* “fire”, which is similar to the hollow verb, as shown in (2) of the classification of weak verbs below, and CVVCV قاضٍ *qaADK* “judge” which is a noun in the genitive case. This noun is called in Arabic الاسم المنقوص *AlAsom AlmanoquwS* “a type of noun that ends with the weak letter ي *y*” (Hijazy and Yusof, 1999). However, this final weak letter is deleted when the noun is in the genitive case as in the current example. As referred to above, we have no patterns to match quadriliteral roots. Nevertheless, there are some nominal quadriliteral roots that are matched in our rule-based tagger. These roots consist of patterns for triliteral roots in addition to either the letter ى *Y* or the two letters اء *A'* which are added at the end of words to denote that they are feminine nouns, e.g. شَكْوَى *$akowaY* “complaint” vs. صَحْرَاء *SaHoraA'* “desert”.

As for **Verbs**, we distinguish between strong and weak verbal roots. They are subdivided into the following:

Strong verbs: These are the verbs that contain no weak letters as one of their radicals. They are formed with the pattern CVCVC.

Three different types of verb are classified under this category.

(1) Regular verbs: These are the verbs whose radicals do not contain either a hamzated, doubled or weak letter, e.g. كَتَبَ *katab* “wrote”

(2) Hamzated verbs: These are the verbs that contain a hamza (glottal stop) among its radicals, e.g. أَكَلَ *Oakal* “ate”.

(3) Doubled verbs: These are the verbs that are composed of two letters and the second one is doubled, e.g. رَدَّ *rad~a* “replied”

Weak verbs: These are the verbs that contain a weak letter as one of their radicals. Weak letters are ألف *Oalif* for the long vowel ا *A* (which can be also represented by the letter ى *Y*), the letter واو *waAw* for the glide و *w* and the letter ياء *yaA'* for the glide ي *y*. Weak verbs are also classified into four categories:

(1) Assimilated or مثال *mivaAl*: These are the verbs with an initial weak radical, e.g. وَقَفَ *waqaf* “stopped”.

(2) Hollow or أجوف *Oajowaf*: These are the verbs with a middle weak radical, e.g. قَالَ *qaAl* “said”.

(3) Defective or ناقص *naAqiS*: These are the verbs with a final weak radical, e.g. سَقَى *saqaY* “irrigated”.

(4) Tangled or لفيف *lafiyf*: These are the verbs that have two weak letters among their radicals. When two weak radicals do not follow each other it is called لفيف مفروق *lafiyf maforuwq* such as وَقَى *waqaY* “guarded”. But when they come following each other it is called لفيف مقرون *lafiyf maqoruwn* such as رَوَى *rawaY* “recounted”.

This classification of triliteral verbal roots can be captured in the following figure.

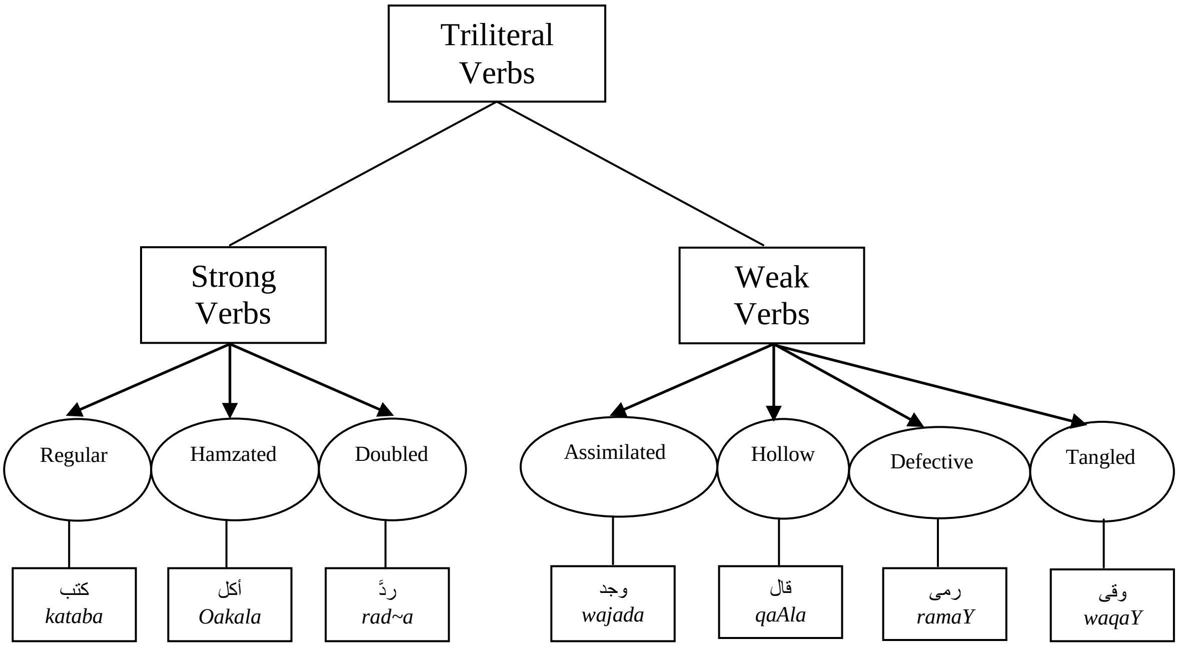


Figure 4.8: Classification of triliteral verbs

It should be recalled from section 4.2.4 that an Arabic word-form can be made up of several lexemes, with a base which may contain inflectional affixes, and possibly a number of cliticized items (conjunctions, prepositions, pronouns). All of these are useful when trying to identify POS tags. Accordingly, in the following lines we will discuss the possible concatenations that are attached to both nouns and verbs. In other words, we will describe the way we handle Arabic morphotactics, which is the way morphemes combine together to form words (Beesley, 1998c).

4.4.2.1.3.1 Nouns

As mentioned earlier, nouns are subdivided into primitive nouns such as شَمْس *$amos* “sun” and derived nouns. Derived nouns can be split into different categories. The main derivational subcategories can be illustrated in the following figure.

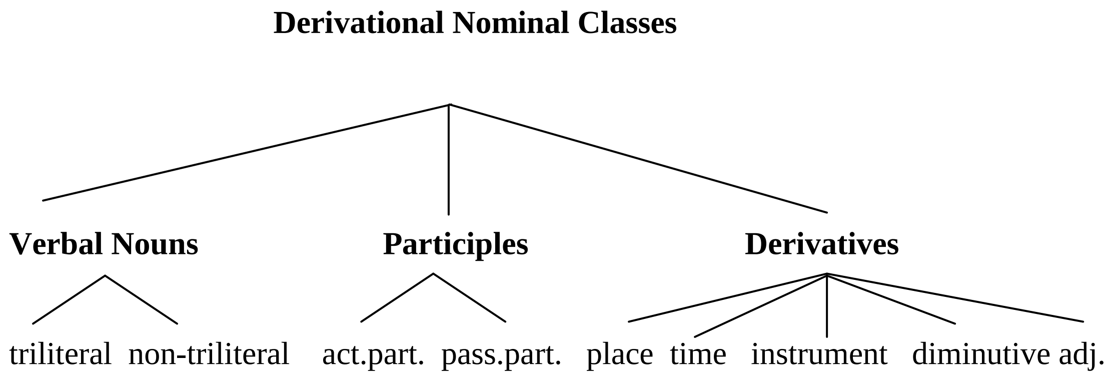


Figure 4.9: Derivational nominal classes

The Arabic POS tagger tags all categories of nouns as **NN**. Examples of nouns can be as follows:

Nominal Classes	**Examples**
Primitive	رَجُل *rajul* “a man”
Triliteral Verbal Noun	فَهْم *fahom* “comprehension”
Nontriliteral Verbal Noun	تَرْجَمَة *tarojamap* “translation”
Active Participle	كاتِب *kaAtib* “a writer”
Passive Participle	مَكْتُوب *makotuwb* “written”
Noun of Place	مَكْتَب *makotab* “an office”
Noun of Time	مَوْعِد *mawoEid* “an appointment”
Noun of Instrument	مِفْتاح *mifotaAH* “a key”
Noun of Instance	قَفْزَة *qafozap* “ a jump”
Noun of Manner	مِشْيَة *mi$oyap* “a gait”
Semi-participial Adjective	كَبِير *kabiyr* “big”
Comparative and Superlative Adjectives (The Elative or Noun of pre-eminence)	أكْبَر *Oakobar* “bigger” الأكْبَر *AlOakobar* “the biggest”
Relative Adjective or Noun	مِصْرِيٌّ *miSoriy~u* “Egyptian”
Diminutive	بُحَيْرَة *buHayorap* “a lake”

Table 4.11: Nominal classes with examples

Possible concatenations in Arabic nouns are shown in Table (4.12) below. These concatenations are (optional) bound morphemes representing affixes and clitics, which come in order before or after an (obligatory) stem as shown in the table.

Proclitics		**Prefixes**	**Stem**	**Suffixes**		**Enclitic**
Question Particle أ *Oa* "is it true that"	Prepositions بِ *bi* "with", لِ *li* "to", كَ *ka* "as" or لَ *la* "surely"	The Definite Article ال *Al* "the", Active or Passive Participle Prefixes مُ *mu* and مَ *ma* respectively	Noun Stem	Feminine Marker ـة *p*, Masculine Dual ان *Ani*, Feminine Dual تان *taAni*, Plural Masculine ونَ *uwna* and ينَ *iyna* or Plural Feminine ات *At*	Indefinite Case Markers (nunation) *N*, *F*, *K* or Definite Case Markers *u*, *a*, *i*	Genitive (or Possessive) Pronouns (Number/ Gender) First Person ي *y* "my", نا *naA* "our", Second Person كَ *ka*, كِ *ki*, كما *kumaA*, كم *kum*, كنَّ *kun~a* "your", Third Person ـهُ *hu* "his", هَا *haA* "her", هُمَا *humaA*, هُم *hum*, هُنَّ *hun~a* "their"
Conjunctions وَ *wa* "and" or فَ *fa* "then"						

Table 4.12: Possible affixes and clitics in Arabic nouns

The noun's affixes and clitics as shown in Table 4.12 are not all concatenated one after another. There are constraints on these concatenations. Some of these constraints can be summarized as follows:-

1- The definite article *Al* "the" cannot co-occur with an indefinite case marker for singular nouns, e.g. **AlkaAtibN*. The definite article and the indefinite case (nunation) markers are in complementary distribution (Hakim, 1995).

2- The definite article *Al* "the" cannot co-occur with a genitive pronoun, e.g. **AlkitaAbuka*.

3- The attached or cliticized genitive pronoun cannot co-occur with an indefinite case marker, e.g. **kitaAbNka*.

4- Prepositions cannot co-occur with nominative or accusative case markers. They occur only with a genitive case marker. Thus, *bilobayoti* “in the house” is correct, whereas **bilobayot(u|a)* is not.

It should be noted that in nouns a number of prefixes can come one after another. For example, the definite article can be followed by one of the derivative prefixes *mu* or *ma*, e.g. المُعَلّمُونَ *AlmuEal~imuwna* “the teachers”. Likewise, suffixes can be attached one after another. But the second suffix will be most likely a case marker. For example, طَالِبَاتٌ *TaAlibaAtN* “female students”.

As we mentioned earlier, we depend on affixes (prefixes and suffixes) to determine the tag of a given word. However, as far as nouns are concerned, it is not always the case that there is a prefix or suffix attached to a noun. This is particularly vivid in the case of the broken plural, which is the traditional grammarians’ term for describing the process of non-concatenative plural formation. Arab grammarians have traditionally distinguished between two types of plurals usually termed ‘sound’ (or regular) and ‘broken’. A sound plural is formed by adding the masculine plural suffix وُنَ *uwna* “nominative”, ينَ *iyna* “accusative and genitive” or feminine plural suffix ات *At* to singular nouns. A broken plural, on the other hand, is formed differently by a number of processes that involve prefixation and changing the diacritic patterns (Haywood and Nahmad, 2005). According to Ratcliffe (1990), the sound plurals are distinguished by characteristic external morphology, whose application to a nominal stem does not affect the internal form of the stem. The broken plurals, on the other hand, are formed in a variety of ways, all of which involve some sort of stem-internal change. Broken plurals are divided into جمع القلة *jamoEu Alqil~api* “plural of paucity”, denoting three to ten items, and جمع الكثرة *jamoEu Alkavorati* “plural of multiplicity”, denoting more than ten items (Ratcliffe, 1998).

There are broken plural cases that comprise four consonants and thus are not tagged correctly by the rule-based tagger. This is because, as mentioned earlier, the rule-based phase of the tagger handles only triliteral roots and ignores the quadriliteral roots at this point. Nonetheless, some other cases that contain three consonants are tagged correctly by the tagger. Table (4.13) illustrates the rule-based tagger’s output for some broken plural cases.

Singular Form	Broken Plural Pattern	Broken Plural Form	Tagger Result
قَلْب *qalob* “heart”	فُعُول **fuEuwl** (CVCVVC)	قُلُوب *quluwb* “hearts”	NN
رَجُل *rajul* “man”	فِعَال **fiEaAl** (CVCVVC)	رِجَال *rijaAl* “men”	NN
كِتَاب *kitaAb* “book”	فُعُل **fuEul** (CVCVC)	كُتُب *kutub* “books"	NN-VV
وَلَد *walad* “boy”	أَفْعَال **OafoEaAl** (CVCVCVVC)	أوْلاد *OawolaAd* “boys”	QPART+NN
شَاهِد *$aAhid* “witness”	فُعَلاء **fuEalaA’** (CVCVCVVC)	شُهَدَاء *$uhadaA'* “witnesses”	NN
شَرِيكَهم*$ariykahum* “their partner”	فُعَلاء **fuEalaA’** (CVCVCVVC)	شُرَكَائِهم *$urakaA}ihimo* “their partners”	UN

Table 4.13: Rule-based tagger’s output for some broken plural cases

In the previous table, the first two broken plural examples قُلُوب *quluwb* and رِجَال *rijaAl* are tagged correctly as **NN** by the rule-based tagger because their patterns contain three consonants only. The third example كُتُب *kutub* is given the ambiguous tag **NN-VV** because its pattern CVCVC matches both nouns and verbs (e.g. كُتُب *kutub* “books” and كَتَب *katab* “wrote”). However, when this broken plural word is attached to a distinctive nominal affix it is tagged as noun, e.g. كُتُبٍ *kutubK* “books (indef. gen.)”. The fourth case أوْلاد *OawolaAd* is tagged as **QPART+NN**. This is because the tagger identifies the first two symbols أ *Oa* as a question particle because they resemble the question particle *Oa* “is it true”. The fifth example شُهَدَاء *$uhadaA’* is tagged correctly as **NN** because though its pattern contains four consonants it ends with the two letters *A’* that are similar to the feminine marker ألف التأنيث الممدودة *Oalif AltaOniyv Almamduwdap* that we referred to earlier. It is thus identified by the tagger the same way as triliteral roots that end with this feminine marker are identified. As for the final example, it is tagged as **UN** for **‘**unknown’. The broken plural here has the same pattern as the previous one but with an enclitic pronoun. When a possessive pronoun is attached to the word شُرَكَاء *$urakaA’* “partners” the final letter ء *’*, which is a shape of the glottal stop همزة Hamza, is changed into the letter ئ *}*, which is another shape of the glottal stop, and thus the tagger could not identify it. This is called orthographic alternation. In fact, the rule-based tagger does not include alternation rules to handle such orthographic changes.

4.4.2.1.3.2 Verbs

Most verbs in Arabic are triliteral, i.e. they are based on roots of three consonants. For instance, the basic meaning of writing is given by the three consonants *k-t-b*. There are a comparatively small number of quadriliteral verbs, e.g. دَحْرَجَ *daHoraja* “to roll”. The simplest form of a verb is the third person masculine singular of the Perfect, e.g. كَتَبَ *kataba* “he wrote”. With regard to tenses, we have indicated earlier that Arabic verbs are subdivided into Perfective (Past), Imperfective (Present) and Imperative. Arabic verbs receive two types of markers: tense markers and agreement markers. Tense markers are represented in prefixes attaching before a verbal stem, whereas agreement markers are represented in suffixes attaching after a verbal stem.
Possible concatenations on Arabic verbs are shown in Table (4.14) below.

Proclitics		**Prefix**	**Stem**	**Suffix**	**Enclitics**
Question Particle أ *Oa* “is it true that”	Complementizers لِ *li* “to” or emphatic لَ *la* “surely/verily”	Tense Markers (Number/ Gender) Imperfective أ *Oa*, يَ *ya*, تَ *ta*, نَ *na* (Active Voice), أُ *Ou*, يُ *yu*, تُ *tu*, نُ *nu* (Active or Passive Voice), سَ *sa* “will” (future marker) , Imperative ا *A*	Verb Stem	Agreement Markers (Number/ Gender) Perfective , Imperfective, Imperative	Object Pronouns (Number/Gender) First Person نِي *niy* “me”, نَا *naA* “us”, Second Person كَ *ka*, كِ *ki*, كما *kumaA*, كم *kum*, كُنَّ *kun~a* “you”, Third Person ـهُ *hu* “him”, هَا *haA* “her”, هُمَا *humaA*, هُم *hum*, هُنَّ *hun~a* “them”
Conjunctions وَ *wa* “and” or فَ *fa* “then”					

Table 4.14: Possible affixes and clitics in Arabic verbs

Since there are various verbal suffixes, we will list them separately in Table (4.15) below.

Person	Number	Masculine	Feminine	Tense	Example
1st	Singular	تُ *tu*	تُ *tu*	Perfective	كتبتُ *katabtu*
1st	Dual	نا *naA*	نا *naA*	Perfective	كتبْنا *katabnaA*
1st	Plural	نا *naA*	نا *naA*	Perfective	كتبْنا *katabnaA*
2nd	Singular	تَ *ta*		Perfective	كتبْتَ *katabta*
2nd	Singular		تِ *ti*	Perfective	كتبْتِ *katabti*
2nd	Singular		ي *iy*	Imperfective Imperative	تَكْتبي *takotubiy* اكْتبي *Akotubiy*
2nd	Dual	تُما *tumaA*	تُما *tumaA*	Perfective	كتبْتُما *katabtumaA*
2nd	Dual	ان *Ani*	ان *Ani*	Imperfective	تَكْتبان *takotubaAni*
2nd	Dual	ا *aA*	ا *aA*	Imperfective Imperative	تكْتبا *takotubaA* (subjunctive & jussive mood) اكْتبا *AkotubaA*
2nd	Plural	تُم *tum*		Perfective	كتبْتُم *katabtum*
2nd	Plural	ونَ *uwna*		Imperfective	تكتبُونَ *takotubuwna*
2nd	Plural	وُا *uwA*		Imperfective Imperative	تكتبُوا *takotubuwA* (subjunctive & jussive mood) اكتبُوا *AkotubuwA*
2nd	Plural		تُنَّ *tun~a*	Perfective	كتبتُنَّ *katabtun~a*
2nd	Plural		نَ *na*	Imperfective Imperative	تكتبْنَ *takotubona* اكتبْنَ *Akotubona*
3rd	Singular		تْ *at*	Perfective	كتبَتْ *katabat*
3rd	Dual	ا *aA*		Perfective	كتبَا *katabaA*
3rd	Dual		تا *atA*	Perfective	كتبَتَا *katabatA*
3rd	Dual	ا *aA*	ا *aA*	Imperfective	يكتبَا *yakotubaA* تكْتبا *takotubaA* (subjunctive & jussive mood)
3rd	Dual	ان *Ani*	ان *Ani*	Imperfective	يكتبَان *yakotubaAni*

					تكتبَان *takotubaAni*
3rd	Plural	وُنَ *uwna*		Imperfective	يكتبُونَ *yakotubuwna*
3rd	Plural	وُا *uwA*		Perfective Imperfective	كتبُوا *katabuwA* يكتبُوا *yakotubuwA* (subjunctive & jussive mood)
3rd	Plural		نَ *na*	Perfective Imperfective	كتبْنَ *katabna* يكتبْنَ *yakotubona*

Table 4.15: Verbal suffixes

All the Arabic examples in Table (4.15) are various forms of the verb كتبَ *kataba* "to write". As can be noticed in the previous tables, the perfective, imperfective and imperative have each a range of prefixes or suffixes or both.

There are constraints on the concatenations and inflections in Arabic verbs. Some of these constraints can be summarized as follows:-

1- The Question Particle *Oa* "is it true that" cannot co-occur with imperative verbs. Thus, **OaAkotub* is not a correct form.

2- The complementizer *li* "to" does not co-occur with the nominative case. Thus, *litakotuba* (accusative) is correct, while **litakotubu* is not.

3- A first person object pronoun does not co-occur with a first person prefix. This applies also to other object pronouns. Thus, *naDoribuhum* "we hit them" is grammatically correct, whereas **naDoribunaA* is not. This rule, in fact, makes sure that the same person cannot act as subject and object at the same time (Attia, 2006; 2008).

4- Cliticized object pronouns do not occur with intransitive verbs or verbs in the passive voice. Thus, *katabotuhu* "I wrote it" is correct, while **nimotuhu* is not.

After discussing nominal and verbal patterns in the rule-based tagger, we now give a sample for the final output of the tagger. In the next section we will discuss some problems that face the rule-based tagger. Here is a sample of a tagged verse from the Qur'an:

4.5 ذَلِكَ الْكِتَابُ لاَ رَيْبَ فِيهِ هُدًى لِّلْمُتَّقِينَ

**alika AlokitaAbu lAa rayoba fiyhi hudFY l~ilomut~aqiyna*

"That is the Book, there is no suspicion about it, a guidance to the pious" (Qur'an, 2:2):-

Words	**POS Tags**
ذَلِكَ **alika*	DEMO
الْكِتَابُ *AlokitaAbu*	NN
لاَ *lAa*	PART
رَيْبَ *rayoba*	NN-VV
فِيهِ *fiyhi*	PREP+PRO
هُدًى *hudFY*	NN
لِّلْمُتَّقِينَ *l~ilomut~aqiyna*	PREP+NN

Table 4.16: A sample of the output of the rule-based tagger

We should make it clear that the plus (+) sign is used to refer to a complex tag, whereas the hyphen (-) is used to refer to an ambiguous tag. In the previous table the word *rayoba* "suspicion" is ambiguous as to whether it is a noun or verb. This is because the short vowel {a} can be attached to both nouns in the accusative case and verbs in the 3rd person singular perfective tense. This ambiguity is to be dealt with in the following stage of TBL. We assessed the accuracy of this tagger (T_{RB}) on a subset of 1100 words from the whole corpus, with the outcome that 75% of words in this 'Gold Standard' are unambiguously assigned the correct tag, and another 15% are assigned ambiguous tags which include the correct one. Of the remaining 10%, about a third are cases where the T_{RB} failed to assign a tag at all.

4.4.2.1.4 Problems

Generally speaking, the major problems with T_{RB}, then, concern either cases where there is insufficient evidence to distinguish between ambiguous tags or ones where the tagger simply fails to assign a tag at all. We will outline the reasons behind these two major problems, i.e. ambiguous and unknown tags.

First, we will discuss the reasons behind ambiguous tags then discuss the reasons that lead to unknown tags. Some words are ambiguous as to whether they are nouns or verbs. This ambiguity may be due to a number of reasons.

1- These words end with the marker {a} which can be attached to both nouns in the accusative case and verbs in the perfective tense and there is no prefix or suffix that can distinguish one of them from the other. This is clear in the previous example رَيْبَ *rayoba* "suspicion", which ends with the accusative case marker. This ambiguity is also clear in the word خَتَمَ *xatama* "sealed" which is a 3rd person singular perfective verb.

2- The **NN-VV** dichotomy has been also assigned to some hollow verbs that have no distinguishing prefix or suffix. The verb قال *qaAl* "said", for instance, is currently tagged as NN-VV. This is because the pattern for this verb, i.e. CVVC, applies also to nouns such as نَار *naAr* "fire".

3- Some broken plural cases that have no distinguishing affixes are tagged as **NN-VV**, e.g. كُتُب *kutub* "books".

As for unknown tags, this occurs due to a number of reasons, which can be discussed as follows.

1- We did not handle the quadriliteral roots in the rule-based tagger for the reason described earlier. So, verbs such as اطْمَأْنَنتم *ATomaOonantumo* "you feel composed", and nouns such as الهُدْهُدَ *Alhudohuda* "the hoopoe" are tagged as unknown in the current stage of the tagger. It is worth noting that these two words have been correctly tagged in the final stage of the tagger, i.e. after applying the probabilistic techniques.

2- Some broken plural forms that contain four consonants are tagged as unknown. This problem is due to the fact that the broken plural in Arabic is not formed by using suffixes like a regular plural, but is formed, as noted above, by a number of processes that involve prefixation and changing the diacritic patterns. This is not tackled in the rule-based tagger. Thus, شَعَائِرَ *$aEaA}ira* "waymarks/symbols/signs" is tagged as unknown.

3- The process of combining morphemes is not always a simple concatenation of morphemic components. Rather, it can involve a number of phonological, morphological and orthographic rules that modify the form of the created word (Habash, 2007). One example is the feminine morpheme ـة *p* called تاء مربوطة *taA'* *marbuwTap*, which is turned into *t* when followed by an enclitic pronoun. Thus, when the word شَهَادَةُ *$ahaAdapu* "testimony" is cliticized with the possessive pronoun *hum*, it is realized as شَهَادَتُهُم *$ahaAdatuhumo* "their testimony". Thus, the rule-based tagger cannot identify this form and so tags it as **UN**. In fact, this alternation creates a problem for POS tagging of undiacritized Arabic, since the pronoun could either be a possessive pronoun or an object pronoun where the pronouns look the same (Diab, 2009). As an example, the word حسنتهم *Hsnthm* could be either a noun + possessive pronoun with the underlying ـة *p* at the final position of the stem, originally حسنة *Hsnp* "good deed", or a verb + object pronoun, where the stem is حسنت *Hsnt*, and thus the whole word حسنتهم *Hsnthm* means "I beautified them".

Having tagged the diacritized corpus by T_{RB} we corrected a portion of it as a Gold Standard to derive some corrective rules from the training corpus by using transformation-based learning (TBL) in the next stage. This leads to the combined T_{RB+TBL} on diacritized text. The TBL technique will be discussed in the coming section.

4.4.2.2 Transformation-Based Learning (TBL)

In the absence of a lexicon, the best way of choosing between ambiguous tags is to look at the preceding and following words. Although Arabic word order is fairly free, some sequences are more common than others - the word immediately following a verb, for instance, is much more likely to be a noun than another verb (14% vs. 9%). In order to take account of this information we use Lager (1999)'s Prolog implementation of Brill (1995)'s 'transformation-based learning' (TBL) approach. TBL is used in this stage to correct the errors in the output of T_{RB}, leading to a combined tagger T_{RB+TBL}.

Nowadays manual encoding of linguistic information is being challenged by automated corpus-based learning as a method of providing an NLP system with linguistic knowledge (Brill, 1995). This has definitely the clear advantage of overcoming the linguistic knowledge acquisition bottleneck. TBL is a way of applying this approach to automated learning of linguistic knowledge. It draws inspiration from both rule-based and stochastic (or probabilistic) tagging. Like rule-based tagging, TBL is based on rules that specify when an ambiguous word should have a given tag. But like stochastic tagging, TBL is a machine learning technique, in which rules are automatically induced from a pre-tagged training corpus. TBL, like some HMM tagging approaches, is a supervised learning technique, since it assumes a previously tagged training corpus (Jurafsky and Martin, 2009).

In transformation-based tagging every word is first assigned an initial tagging. This can be done in a variety of ways. In the work described here we use either rule-based tagger (for diacritized text) or Bayes+MLE (for undiacritized text). Then a sequence of rules is applied that change the tags of words based upon the contexts in which they appear. These rules are applied deterministically, in the order they appear in the list. As a simple example, if ***race*** appears in the corpus most frequently as a noun, it will initially be mistagged as a noun in the sentence:

4.6 We can **race** all day long.

The rule *Change a tag from NOUN to VERB if the previous tag is a MODAL* would be applied to the sentence, resulting in the correct tagging. In fact, the environments that are used to change a tag are the words and tags within a window of three words (Brill and Wu, 1998). The following figure illustrates how TBL works.

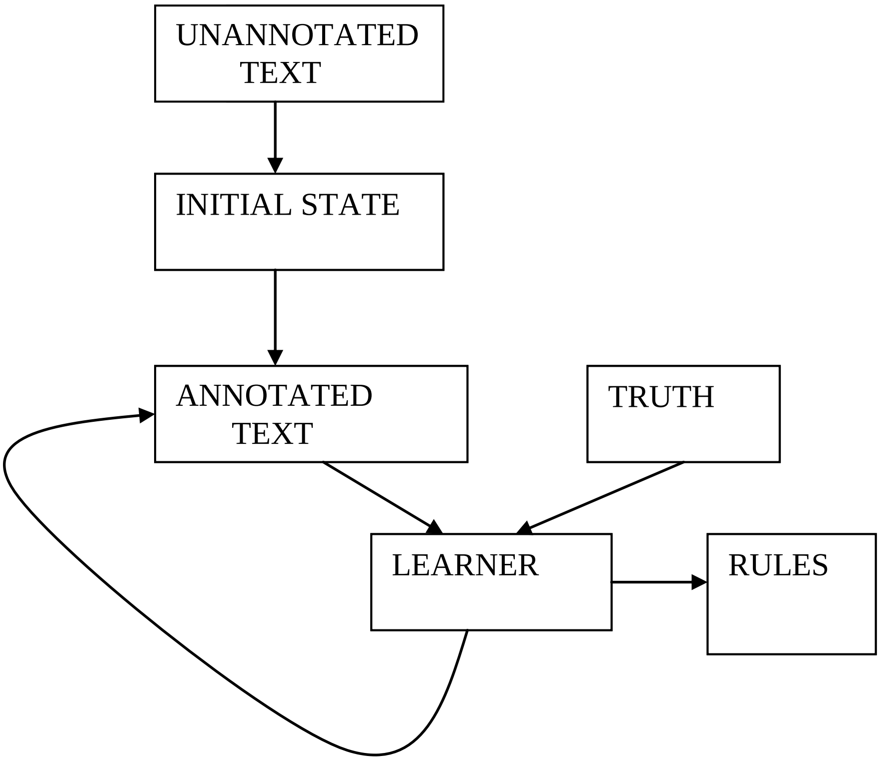


Figure 4.10: Transformation-Based Error-Driven Learning

First, as the figure shows, unannotated text is passed through an initial-state annotator. The initial-state annotator can range in complexity from just assigning random structures to assigning the output of a sophisticated manually created annotator. As far as POS tagging is concerned, various initial-state annotators can be used. These annotators may be, for instance, the output of a stochastic n-gram tagger; labelling all words with their most likely tags as indicated in the training corpus; and naively labelling all words as nouns. As noted above, we use either the rule-based tagger or the Bayesian tagger, depending on the text being analyzed.

Once the text has been passed through such an initial-state annotator, as shown in the previous figure, it is then compared to the *truth*. A manually annotated corpus is used as our reference for truth (i.e. a Gold Standard). In the work reported here we manually correct a small portion (1100 words) from the output of either T_{RB} for diacritized text or T_B+T_{ML} for undiacritized text. An ordered list of transformations is then learned and hence applied to the output of the initial-state annotator to make it better resemble the truth or the Gold Standard. Basically, there are two components to a transformation: a **rewrite rule** and a **triggering environment**. An example of a rewrite rule for POS tagging is:

Change the tag from modal to noun.

And an example of a triggering environment is:

The preceding word is a determiner.

Taken together, the transformation with this rewrite rule and triggering environment when applied to the word *can* would correctly change the mistagged:

4.7 *The/*determiner *can/***modal** *rusted/*verb.

To

4.8 *The/*determiner *can/***noun** *rusted/*verb.

In fact, TBL needs to consider every possible transformation, so as to pick the best one on each pass through the algorithm. Consequently, this algorithm needs a way to limit the set of transformations. This is done by designing a small set of **templates**, i.e. abstracted transformations. Every allowable transformation is definitely an instantiation of one of the templates. The following figure lists Brill’s set of templates, with some templates that we have added.

(A,B,C,D) # tag:A>B <- tag:C@[-1] & tag:D@[1].

(A,B,W) # tag:A>B <- wd:W@[-1,-2].

(A, B, W) # tag:A>B <- sf:W@[0].

(A, B, C, W) # tag:A>B <- sf:W@[0] & tag:C@[1].

(A, B, W) # tag:A>B <- pf:W@[0].

(A, B, C, W) # tag:A>B <- pf:W@[0] & tag:C@[1].

Brill's (1995) templates. Each begin with "*Change tag **A** to tag **B** when:…..*".

Figure 4.11: Examples of TBL templates

In the previous figure we have given some examples for Brill templates. The first two templates are of TBL original templates which comprise 26 templates. The first one deals with changing the tag of a word from A to B if the previous tag is C and the following tag is D. As for the second one, it is concerned with changing the tag of a word from A to B if W, i.e. a given word, is either the previous word or the one before that. Thus, the triggering environment of the first template deals with parts-of-speech categories, while that of the second one deals with particular words. The remaining four templates, however, are new additions that we annexed to the TBL templates. They use prefixes and suffixes as triggering environments to correct POS tags. Thus, the first one of our additions states that tag A should be changed to tag B if the current word ends with a specific suffix 'sf:W'. The second added template is similar to the previous one but takes into consideration the POS tag of the following word also. As for the third and fourth added templates, they are based on the same principle but with regard to prefixes this time: 'pf:W'.

The essence of TBL is to cycle through sets of potential corrective rules looking for the single rule that has the greatest net beneficial effect. You then apply this rule throughout the corpus and repeat the process, stopping when the best rule's effect is below some prespecified threshold (if the threshold is too low then the process tends to learn accidental patterns which do not generalize effectively beyond the corpus). Thus, TBL algorithm has three major stages. First, it labels every word in the corpus with its most-likely tag. Then, it examines every possible transformation, and selects the one that results in the most improved tagging. Finally, it re-tags the corpus according to this rule. The last two stages are repeated until some criterion is reached, such as insufficient improvement over the previous pass (Jurafsky and Martin, 2009).

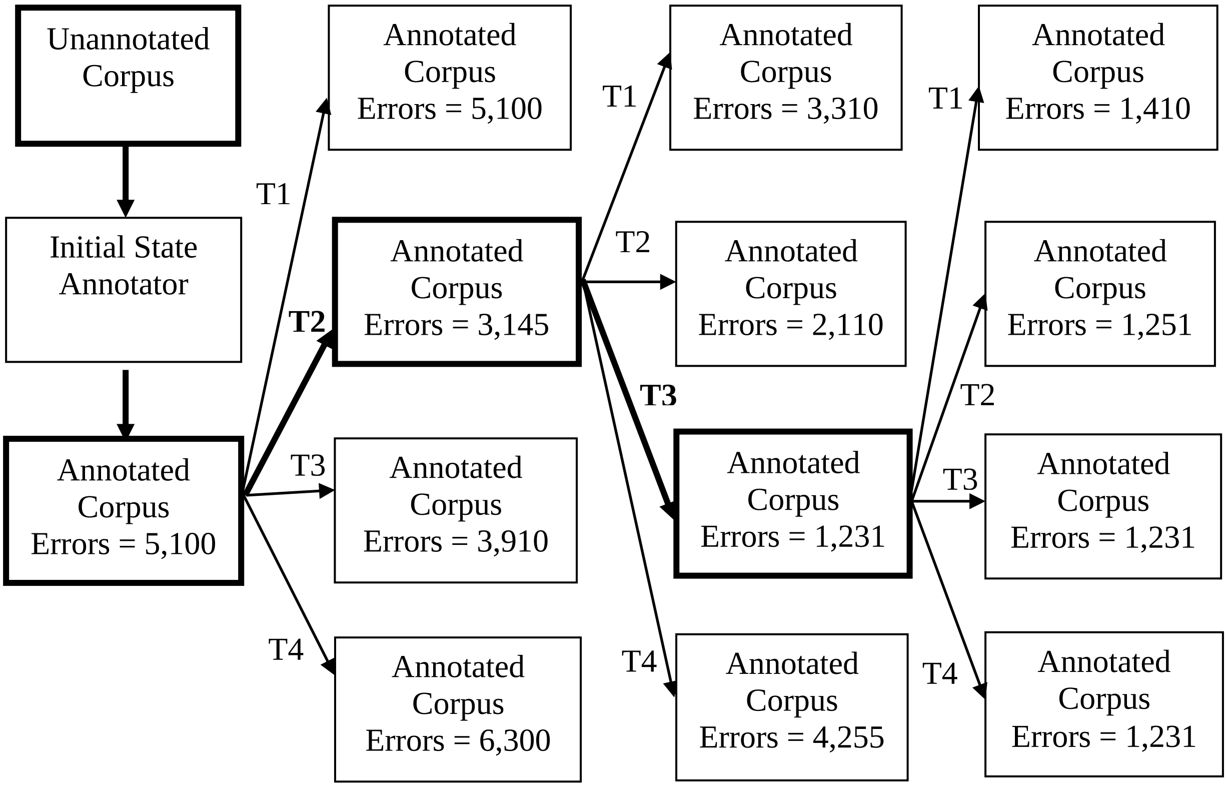

Unannotated Corpus
Initial State Annotator
Annotated Corpus Errors = 5,100
T1
T2
T3
T4
Annotated Corpus Errors = 5,100
Annotated Corpus Errors = 3,145
Annotated Corpus Errors = 3,910
Annotated Corpus Errors = 6,300
T1
T2
T3
T4
Annotated Corpus Errors = 3,310
Annotated Corpus Errors = 2,110
Annotated Corpus Errors = 1,231
Annotated Corpus Errors = 4,255
T1
T2
T3
T4
Annotated Corpus Errors = 1,410
Annotated Corpus Errors = 1,251
Annotated Corpus Errors = 1,231
Annotated Corpus Errors = 1,231

Figure 4.12: An example of Transformation-Based Error-Driven Learning (Brill, 1995).

The previous figure, (taken from Brill, 1995), shows an example of learning transformations. In this example, we presume that there are only four possible transformations; T1 through T4. First, the unannotated training corpus is processed by the initial state annotator, which results in an annotated corpus with 5,100 errors, determined by comparing the output of this initial state annotator with the Gold Standard. In the next step we apply each of the possible transformations in turn and score the accuracy of the resulting annotated corpus. In this example we see that applying transformation T2 results in the largest reduction of errors, and thus T2 is learned as the first transformation. T2 is then applied to the entire corpus, and learning continues. At this phase of learning, transformation T3 results in the largest reduction of errors, as can be see in the figure above, and so it is learned as the second transformation. After applying the initial state annotator, then T2 and then T3, we see that no further improvement (i.e. reduction in errors) can be obtained from applying any of the transformations, and so the learning process stops. To annotate new text, this text is first annotated by the initial state annotator, followed by the application of both transformation T2 and then T3 in succession.

4.4.2.2.1 Training and Test Sets

To use TBL to improve the performance of an existing tagger you need a corpus that has been assigned tags by the existing tagger and has then been manually corrected, so that TBL can see the kinds of errors that the initial tagger produces. We have used a fairly small subset of the full diacritized corpus (1100 words taken from the 77,800 in the Holy Qu’ran itself). These 1100 words are the Gold Standard which we use to be our training set for TBL.

Trying TBL with the Gold Standard (which is all we have correct tags for), and using tenfold cross-validation for testing, we obtain a set of 34 rules which lead to a score of 90.8% correct unambiguous tags. In other words, T_{RB+TBL} (rule-based tagger with TBL) disambiguates the choices left open by T_{RB} very effectively, but does very little to override other errors (90.8% correct unambiguous tags after TBL compared to 90% of tags which include the correct tag as an option after the rule-based tagger).

The top rule generated in this process simply says that the tag **NN-VV** (i.e. the tag for something which could be either a noun or a verb) should be changed to **VV** if any of the following three words are nouns. 26 of the 34 rules are similarly concerned with choosing between ambiguous readings, and another 3 assign tags in cases where T_{RB} failed to specify a tag at all. The remaining 5 correct mistakes made by T_{RB}, e.g. one that retags an auxiliary as a simple verb if it is followed by a noun. Some of the generated rules are listed in the following table along with corresponding templates.

Generated Rules	**Corresponding Templates**
Rule ('NN-VV','VV','NN').	(A,B,C) # tag:A>B <- tag:C@[1,2,3].
Rule ('UN','NN','Al').	(A, B, W) # tag:A>B <- pf:W@[0].
Rule ('VV','NN','VV','NN').	(A,B,C,D) # tag:A>B <- tag:C@[-1] & tag:D@[1].

Table 4.17: Examples of TBL-generated rules for diacritized text

The previous table shows some of the corrective rules generated by TBL along with their corresponding templates. The three rules refer to three different cases; with the first one dealing with ambiguous tags, the second with unknown tags, and the third with wrong tags. Consequently, the first one refers to the top rule mentioned above, i.e. an ambiguous **NN-VV** tag should be changed to **VV** if any of the following three words are **NN**. The second rule deals with changing the tag **UN**, i.e. ‘unknown’, to **NN** if this current word starts with the definite article ال *Al* “the”, which is a prefix attached to words. As for the third rule, it is concerned with correcting a mistake made by the rule-based tagger. This rule says that **VV** should be changed to **NN** if the preceding tag is **VV** and the following tag is **NN**. In some cases, however, TBL changes an ambiguous tag to a wrong tag, as shown below.

The following table shows some examples from the diacritized corpus with their initial tagging given by T_{RB} and the new tag given by TBL.

Word	**Initial State Tagging (T_{RB})**	**Gold Standard**	**TBL Tagging**
الْكِتَابُ *AlokitaAbu*	NN	NN	NN
رَبِّهم *r~ab~ihimo*	NN+PRO-VV+PRO	NN+PRO	NN+PRO
كَفَرُوا *kafaruwAo*	PREP+NN-NN-VV	VV	VV
رَيْبَ *rayoba*	NN-VV	NN	VV

Table 4.18: A sample of diacritized text tagged by TBL

In the previous table, the first word الْكِتَابُ *AlokitaAbu* “the book” is correctly tagged by T_{RB} and is kept as it is in the TBL phase. As for the words رَبِّهم *r~ab~ihimo* “their lord” and كَفَرُوا *kafaruwAo* “disbelieved”, they are initially tagged wrongly by T_{RB} then corrected by the TBL tagger in accordance with the Gold Standard. However, the TBL tagger has taken the wrong decision when it tagged the word رَيْبَ *rayoba* “suspicion” as **VV** instead of **NN**.

There does not seem to be much more that can be done without using a lexicon. In particular, T_{RB} has difficulty with cases where there are complex sets of clitics. There are quite a few such cases overall, but each one occurs only a few times in the Gold Standard, so that the transformation-based learner has very little evidence to work with. T_{RB+TBL} tagger has scored an estimated accuracy of 90.8%.

4.4.2.3 Bayesian Model

Having tagged the entire diacritized corpus with T_{RB+TBL}, we removed the diacritics from words and kept the tags. We removed the diacritic marks that are represented in the transliterated symbols (*u, a, i, o, N, F, K, ~*). The first four symbols stand for the three short vowels and the sukun "lack of a vowel" respectively, the second three symbols for tanween "nunation", and the final symbol for shadda "consonant gemination". Then we apply two subsequent stages of tagging on this undiacritized corpus, namely Bayes (T_B) and Maximum Likelihood (T_{ML}).

For tagging undiacritized text we used a very simple set of clues. We simply assumed that the first two or three and last two or three letters of a word would provide evidence about its part of speech (Bayesian tagger, T_B), and that this could be supplemented by considering the parts of speech assigned to the preceding and following words (maximum likelihood tagger, T_{ML}).

The information that we used in the rule-based tagger is largely unavailable in the undiacritized version of the text. Some inflectional affixes are still visible, but many of them are deleted when we remove the diacritics. In order to get a rough approximation to the set of affixes in the diacritized text, we simply collected the conditional probabilities linking the first and last two or three letters in a word with its tag. We were not expecting this to be particularly reliable, but given that we have no lexicon for open-class words there was not very much that we could use in order to get an initial assignment of tags for the undiacritized text. We therefore just used Bayes' theorem to compute the possibility that a given word that begins with two or three given letters and ends with two or three given letters is tagged as such and such.

Notably, the aim is to develop a tagger for use with undiacritized text. Thus the first and last three written letters will be a mixture of affixes of various kinds, together with some elements of the underlying words. This is a much messier, and much less theoretically motivated, set of patterns than the affix sets used in the rule-based tagger. Given that the information we are using seems likely to be unreliable, the results obtained are very gratifying. We will start with discussing the principles underlying Bayes model and the tagger we obtain by using this model. Then, we will discuss maximum likelihood estimation (MLE) and the results we obtain after applying it to the tagger.

Bayes' theorem (often called Bayes' law after Thomas Bayes) is a law from probability theory. It relates the prior (or marginal) and conditional probabilities of two random events. It is often used to compute posterior probabilities given observations. For example, a patient may be observed to have certain symptoms. Bayes' theorem can be used to compute the probability that a proposed diagnosis is correct, given that observation. Bayes' theorem can be expressed formally as follows:

$$P\,(x|y) = \frac{P\,(y|x) * P\,(x)}{P\,(y)} \tag{4.1}$$

Each term in Bayes' theorem has a conventional name:

- P (*x*) is the prior (or marginal) probability of *x*. It is "prior" in the sense that it does not take into account any information about *y*.
- P *(x|y)* is the conditional probability of *x*, given *y*. It is also called the posterior probability because it is derived from or depends upon the specified value of *y*.
- P *(y|x)* is the conditional probability of *y* given *x*.
- P (*y*) is the prior or marginal probability of *y*.

Here is an example applied to POS tagging in English. Suppose that we want to know whether a given word that ends with the suffix (*ing*) should be tagged as **noun**. According to Bayes' theorem, this can be computed as follows:

$$P\,(noun|ing) = \frac{P\,(ing|noun) * P\,(noun)}{P\,(ing)} \tag{4.2}$$

We are computing the hypothesis in case of evidence. Thus, we first need to know the following:

- The conditional probability *(y|x)*, i.e. the probability that a word ends with the suffix (*ing*) in case it is tagged as noun.
- The prior probability of *x*, i.e. the probability that a word is tagged as noun, regardless of any other information.

- The prior probability of *y,* i.e. the probability of all words ending with (*ing*) regardless of any other information.

In our POS tagger we just collected statistics about the 2- and 3-letter prefixes and suffixes, using Bayes' theorem to compute the probability that a given word that begins with two or three given letters and ends with two or three given letters is tagged as such and such. Here is the equation:

$$P\,(tag|prefix...suffix) = \frac{P\,(prefix|tag) * P\,(suffix|tag) * P\,(tag)}{P\,(prefix) * P\,(suffix)} \tag{4.3}$$

The previous equation can be illustrated as follows.

- Given the first and last two or three characters in the word, look up the conditional probability of each tag given those characters and multiply them together.

This means that if, for instance, a word begins with the definite article ال *Al* and ends with the masculine plural suffix *wn* such as المسلمون *Almslmwn* "the Muslims" and we want to know whether it is noun, verb, preposition ….etc. we can compute this through multiplying the number of words that have been tagged in our corpus as **noun** by the number of words that have been tagged as **noun** providing that they begin with the definite article ال *Al* and by the number of words that have been tagged as **noun** providing that they end with the masculine plural suffix ون *wn*, divided by the total number of words in our corpus that begin with the definite article *Al* and end with the masculine plural suffix *wn*. Note that we do not encode any facts about particular pairs of letters, e.g. that ال *Al* is often the definite article and ون *wn* is often the nominal masculine plural ending. We just collect the relevant statistics for every pair of letters. In this way we derive a POS tagger based on Bayes' theorem.

In order to train this tagger, we need a lot of tagged data. Part of the point of the current exercise is to see how far we can get without manually tagging a large amount of text. We therefore used the rule-based tagger, together with the corrective rules obtained by the TBL phase, to derive the training data. We therefore tagged the corpus using T_{RB+TBL}, and then undiacritized it. This produced a reasonable sized corpus (around 78,000 words): this is not large when considered as a resource for examining properties of individual words, but you need less data for learning about word classes than you do for learning about individual words.

It is noteworthy that we have tried different combinations of initial and final letters in a word. We have experimented with collecting statistics about the first two and last two letters and compared this with statistics about the first three and last three letters. It turned out that the latter combination gave a better result. Collecting statistics about the initial and final letters of words indicates how likely a word that starts or ends with particular letters is associated with some POS tag. We will give one example for experimenting with the first and last pair of letters, along with their associated tags and percentage of occurrence. First, the statistics for an initial pair of letters are given followed by those statistics for a final pair of letters.

Initial Pair of Letters	**POS Tags**	**Percentages of Occurrence**
ال *Al*	NN	0.85960
	RELPRO	0.1124
	VV	0.02259
	NN-VV+PRO	0.00370
	PART	0.00089
	NN+PRO	0.00080

Table 4.19: Statistics for an initial pair of letters

In the previous table, the first tag, namely **NN**, is the most common of all in the corpus for the distribution of words beginning with ال *Al*. This is due to the fact that these two letters are indeed a standard prefix, i.e. the definite article. Where some pair of letters is indeed a standard prefix, the statistics reflect this, but even obvious prefixes like the definite article will turn up in unexpected cases. An example of a final pair of letters is shown in the following table.

Final Pair of Letters	**Tags**	**Percentages of Occurrence**
ات *At*	NN	0.78265
	CONJ+NN	0.09121
	PREP+NN	0.04279

	VV	0.041666
	DHUW	0.02027
	PREP+DHUW	0.01351
	PART	0.00225
	CONJ+DHUW	0.00225
	NN-CONJ+VV	0.00112
	CONJ+PART	0.00112
	CONJ+PREP+NN	0.00112

Table 4.20: Statistics for a final pair of letters

It is also noticeable that the first tag **NN** is the most common of all in the corpus for the distribution of words ending with ات *At*.

Using the statistics obtained above, we derive a POS tagger based on the Bayesian model T_B. The following table shows a sample of the output of this tagger.

Words	**POS Tags**
الكتاب *AlktAb*	NN
للمتقين *llmtqyn*	PREP+NN
الذين *Al*yn*	NN
يؤمنون *yWmnwn*	VV
ويقيمون *Wyqymwn*	CONJ+VV
رزقناهم *rzqnAhm*	VV+PRO

Table 4.21: A sample of the output of the Bayesian tagger

We can notice in table (4.21) that all the examples have been tagged correctly, except one example, namely the masculine plural relative pronoun الذين *Al*yn* “who”. It is tagged wrongly as **NN**. This wrong tag is due to the fact that most of the words in our corpus that begin with the definite article ال *Al* are tagged as noun. In fact, there are words beginning with *Al* that are tagged as relative pronoun in our corpus, but their number is very few comparing to the big number of nouns. This is exactly the kind of error that you would expect from a Bayesian tagger - rare words that share some of the properties of common ones will be swamped.

4.4.2.4 Maximum Likelihood Estimation (MLE)

It has been pointed out above that T_B is supplemented by considering the parts of speech assigned to the preceding and following words (maximum likelihood tagger, T_{ML}). The general principle underling MLE can be outlined as follows.

- For each tag ‘x’ assigned a non-zero value by T_B, look at the tags assigned to the previous word and add the probability associated with each such tag ‘y’ multiplied by the probability that ‘y’ would be followed by ‘x’.

The above MLE principle can be formally expressed by the following equation.

$$P\ (t_n|w_n,\ w_{n-1}) = P_E\ (t_n|w_n) * \sum_{t_i \varepsilon T_{n-1}} P_T\ (t_n|t_i) * P\ (t_i|w_{n-1},\ w_{n-2}) \tag{4.4}$$

This equation can be illustrated as follows:

- $P\ (t_n|w_n,\ w_{n-1})$ is an estimate of the probability that the tag is t_n given that this word is w_n and the previous one is w_{n-1}. This is what we want to calculate.
- P_E is the emission probabilities (for which we use our Bayesian calculation on prefixes and suffixes).
- P_T is the transition probabilities (obtained by using T_{RB+TBL} on diacritized corpus).
- T_{n-1} is all the possible tags, t_i, for w_{n-1}.

This equation would give the best possible estimate of $P\ (t_n|w_n,\ w_{n-1})$ if P_E and P_T were equally accurate estimates of the emission and transition probabilities. We do not know which one is in fact more reliable. We, therefore, use a weighted version of the basic equation, including a parameter *a* which assigns more or less weight to the emission probability. This weighted version of the basic equation is illustrated in (4.5) below.

$$P\ (t_n|w_n,\ w_{n-1}) = a * P_E\ (t_n|w_n) + \sum_{t_i \varepsilon T_{n-1}} P_T\ (t_n|t_i) * P\ (t_i|w_{n-1},\ w_{n-2}) \tag{4.5}$$

We find the value of the parameter *a* by running it with lots of choices to see which does best. We carried out a number of experiments to determine the optimum weighting factor. Thus, we tried *a* = [0.1, 0.2, 0.3….., 2.0] and the optimal value we got for *a* is 1.4. The following illustrative example shows how we calculate both the emission probabilities and transition probabilities to derive the T_{ML}. Suppose we have a bigram, i.e. two words, which we will call w_1, w_2. These two words have different emission probabilities for a given number of tags. We can calculate the MLE as in the following table.

	w_1	**w_2**
Emission Probabilities	**VV**:0.7 **NN**:0.3	**VV**:0.2 **NN**:0.8
Transition Probabilities	w_1 VV → w_2 VV = 0.1 w_1 VV → w_2 NN = 0.9 w_1 NN → w_2 VV = 0.6 w_1 NN → w_2 NN = 0.4	

Table 4.22: An illustrative example for emission and transition probabilities

The previous figures can be summed to give the probability of the tag for w_2, adding the weighted factor we mentioned above as follows.

$$P\ (w_2\ VV) = weight * 0.2 + (0.7 * 0.1 + 0.3 * 0.6) \tag{4.6}$$

$$P\ (w_2\ NN) = weight * 0.8 + (0.7 * 0.9 + 0.3 * 0.4) \tag{4.7}$$

In this way we try to find the sequence of tags that maximizes the previous value, i.e. computing the probability that the second word (w_2) is a verb or noun relying on the lexical (emission) probability and transition (contextual) probability between tags. All these probabilities are calculated from our training corpus.

The outcome for both T_B and T_{ML} is rather surprising. Using T_B, i.e. just looking at the prefixes and suffixes, scores 91.1 %. Supplementing this with information about transition probabilities, i.e. using T_{ML}, increases this to 91.5%. In other words, the very simple technique of combining probabilities based on three letter prefixes and suffixes on undiacritized text outperforms the combination of rule-based and transformation-based tagging on diacritized text, with transition probabilities providing a small extra improvement. This is despite the fact that the statistics used for training T_B and T_{ML} were obtained by using T_{RB+TBL}: the tagger for undiacritized text actually corrects mistakes in the training data. It is worth noting that we carried out a back-off technique in our experiments. It sometimes happens that T_B and T_{ML} will fail to assign a tag at all if the first and last three letters have not been seen together in the corpus. In that case we back off to the first and last two, or even one, letters. The result, however, was very slight improvement. So, T_B and T_{ML} slightly increased to 91.3% and 91.6% respectively. It is interesting to note that using a hidden Markov model for exploiting transition probabilities turned out to perform substantially worse than the maximum likelihood model, though the reasons for this are unclear.

4.4.2.5 TBL Revisited

We used transformation-based learning to improve the performance of the rule-based tagger. The rules that are used in the rule-based tagger are generally correct, in that they very seldom suggest incorrect tags. The problem with the rule-based tagger is that there are numerous cases where more than one set of patterns applies, so that a large number of words are given ambiguous tags. We used the transformation-based tagger to learn how to disambiguate these cases.

The maximum likelihood tagger does not produce ambiguous tags, but it does make mistakes, so we again use transformation-based tagging to improve the situation, producing our final tagger T_{ML+TBL}. The outcome this time is that the 91.6% obtained by the maximum likelihood tagger goes up to 92.8%. Notably, we have applied the same back-off technique to T_{ML+TBL} but the score did not improve at all.

Some of the derived rules after applying TBL to undiacritized text can be shown in the following table.

Generated Rules	**Corresponding Templates**
Rule ('VV','NN+PRO','PREP')	(A,B,C) # tag:A>B <- tag:C@[-1].
Rule ('UN','VV','NN').	(A,B,C) # tag:A>B <- tag:C@[-1,-2,-3].

Rule ('VV','NN+PRO',bk).	(A, B, W) # tag:A>B <- sf:W@[0].

Table 4.23: Examples of TBL-generated rules for undiacritized text

In the previous table some of the corrective rules derived by TBL for undiacritized text are shown with their corresponding templates. The first rule says that a **VV** tag should be changed to **NN+PRO** if the previous tag is **PREP**. As for the second rule, it states that an unknown tag, i.e. **UN**, should be changed to **VV** if one of the three previous tags is **NN**. Finally, the third rule says that a **VV** tag should be changed to **NN+PRO** if the current word ends with the suffix (*...bk*).

There is a risk that T_{ML+TBL} is learning rules that are very specific to the Gold Standard which is only 1100 words. Inspection of the rules that are inferred suggests that this is not so. In particular, the only rules that refer to specific lexical items are ones that deal with closed class words, e.g. the fact that when من *mn* occurs before a verb then it must be the relative pronoun whose full form is *man* “who” rather than the preposition *min* “from”. Rules dealing with closed class items are likely to be generally applicable, so it seems likely that these rules will be reasonably accurate on the wider corpus.

The following table shows some examples from the undiacritized corpus with their initial tagging given by T_{ML} and the new tag given by TBL.

Word	**Initial Tagging (T_{ML})**	**Gold Standard**	**TBL Tagging**
ريب *ryb*	NN	NN	NN
الذين *Al*yn*	NN	RELPRO	RELPRO
شياطينهم *$yATynhm*	NN	NN+PRO	NN+PRO

Table 4.24: A sample of undiacritized text tagged by TBL

In the previous table, the first word ريب *ryb* “suspicion” is correctly tagged by T_{ML} and so it did not need any intervention by the TBL tagging. As for the word الذين *Al*yn* “who”, it is corrected by the TBL tagger in accordance with the Gold Standard. As for the word شياطينهم *$yATynhm* “their devils”, the TBL tagger has also corrected the tag of the word.

It is worth wondering about the effects of varying the length of prefixes and suffixes used by the Bayesian tagger. Using longer affixes produces considerable improvements in the overall performance of the tagger. So, collecting statistics about the first and last three letters produces better results than using the first and last two letters. It has been noted before that the initial score we got for the tagger was 95.8% (Ramsay and Sabtan, 2009). But the score decreased when we extended the tagset. The table below summarizes the final results for combinations of various techniques on the extended tagset.

Techniques	**Scores**
Just Bayes	0.911
Bayes+Backoff	0.913
Bayes+TBL	0.929
Bayes+TBL+Backoff	0.931
Just ML	0.915
ML+Backoff	0.916
ML+TBL	0.928
ML+TBL+Backoff	0.928

Table 4.25: Scores for the techniques used to POS tag undiacritized Arabic

The previous scores are obtained when we train and test on the same dataset. In the above table, using Bayes with ambiguities backed-off plus TBL scores better than ML+TBL+Backoff, i.e. the score is 0.3% higher. However, using 10-fold cross validation has decreased the above scores, where the final ML+TBL+Backoff scores 0.912, but Bayes obtains a lower score, i.e. 0.906. The reason for the lower scores after doing cross-validation may be attributed to the distribution of words in the Gold Standard, where we learn rules from words that occur more frequently in one portion of the Gold Standard, but, still, occur less commonly in other parts of the Gold Standard.

4.5 English Lexicon-Free POS Tagger

In order to POS tag the English text in the parallel corpus we used an English POS tagger that has been developed, adopting the same lexicon-free approach as for the Arabic POS tagger. This tagger has been developed by Prof. Allan Ramsay at the School of Computer Science at the University of Manchester. The English tagger has been developed using a combination of rule-based, TBL and stochastic techniques. The English tagger is based on the BNC basic (C5) tagset, but with some modifications. The modified tagset uses the BNC general tags and ignores the fine-grained details. Thus, in the BNC tagset the tags **AJ0**, **AJC** and **AJS** are used to mean general or positive adjectives, comparative adjectives and superlative adjectives respectively. But the more general tag **AJ** is used in the developed English tagset to cover all types of adjectives. This coarse-grained tagset is similar to the used Arabic tagset in which language-specific details are ignored. In fact, using these coarse-grained tagsets in Arabic and English is more feasible for our basic task of lexical selection. This is because the English morphological features are not identical to the Arabic ones. Emphasizing this notion, Melamed (1995) points out:

> "Tag sets must be remapped to a more general common tag set, which ignores many of the language-specific details. Otherwise, correct translation pairs would be filtered out because of superficial differences."

Therefore, we used more general tagsets in both Arabic and English.

The developed English tagset that is used to POS tag the English text in the parallel corpus is described in the following table.

POS Tag	**Description**	**Examples**
AJ	All types of adjective: positive, comparative or superlative.	*old, older, oldest*
AT	Article.	*a, an, the*
AV	All types of adverb: general adverb, adverb particle, or *wh*-adverb.	*often, up, where*
CJ	All conjunctions: coordinating and subordinating conjunctions.	*and, that, when*
CR	Cardinal number.	*One, 3, seventy-five, 3505*
DP	Possessive determiner.	*his, her, your, their, our*
DT	All types of determiner: general or *wh*-determiner.	*this, all, which, what*
EX	Existential *there*, i.e. there occurring in the *there is …* or *there are …* construction.	*there*

IT	Interjection or other isolate.	*oh, yes, wow*
NN	All types of common noun: neutral, singular or plural.	*aircraft, pencil, pencils*
NP	Proper noun.	*London, Michael, IBM*
OR	Ordinal numeral.	*first, fifth, last*
PN	All types of pronoun: indefinite pronoun, personal pronoun, Wh-pronoun, or reflexive pronoun.	*everything, you, who, yourself*
PO	The possessive or genitive marker *‘s*	*Peter’s*
PR	All types of preposition.	*in, at, of, with*
PU	Punctuation marks. (N.B. This tag is used only to mark the beginning of verses in the corpus for which we used a double colon (::), since we removed the punctuation marks in the English text to match the already unpunctuated Arabic text.)	*!, :, ., ?*
TO	Infinitive marker *to*	*to*
UN	Unknown items, i.e. the items that the tagger could not classify. These include non-English words that are kept in the translation with their Arabic pronunciation; they are mostly Arabic proper nouns.	*Iblîs* “Arabic name for the devil”, *shayatîn* “devils”, *Mûsa* “Moses”
VB	All forms of the verb BE: infinitive, present, past, progressive or past participle.	*be, is, was, being, been*
VD	All forms of the verb DO: infinitive, finite base, past, progressive or past participle.	*do, does, did, doing, done*
VH	All forms of the verb HAVE: infinitive, finite base, past, progressive or past participle.	*Have, ‘ve, had, ‘d, having, had*

VM	Modal verbs	*will, would, can, could*
VV	Main verbs in any tense: finite base, infinitive, -*s* form, past, progressive, or past participle.	*forget, forgets, forgot, forgetting, forgotten*
XX	The negative particle *not* or *n't*.	*not, n't*
ZZ	Alphabetical symbols.	*A, a, B, b,*

Table 4.26: The used English tagset

The total number of grammatical tags in the BNC basic tagset is 61, but the reduced tagset which was used to POS tag the English corpus contains 25 tags, as described in the above table. We are using the English tagger as a black box. It is not one of the contributions of this thesis and so we are not going to evaluate it as far as accuracy is concerned. In actual fact, it has been found out that using the developed English tagger was a useful tool in the current project, in spite of the wrong tags it produces. Nonetheless, if the English text had had a smaller number of wrong tags than those made by the current tagger, it would have resulted in a better accuracy score for the basic task in this study, namely lexical selection of open-class translational equivalents.

The following table shows examples from the POS-tagged English corpus (translation of Qur'an, 2:2).

Words	**POS Tags**
that	CJ
is	VB
the	AT
Book	NN
there	EX
is	VB
no	AT
suspicion	NN
about	PR
it	PN

a	AT
guidance	NN
to	PR
the	AT
pious	NN

Table 4.27: A sample of the POS-tagged English text

We can observe in this table that the English tagger has correctly tagged all words with the exception of the first word that should have been tagged as DT instead of CJ. Nonetheless, we will see in section 5.5 that we are not always so lucky.

4.6 Summary

In this chapter we have discussed the morphological nature of Arabic, paying attention to the complex structure of Arabic words. We have also touched upon the NLP task of POS tagging and the different approaches that are adopted in the field. In addition, we pinpointed the challenges facing Arabic POS tagging and reviewed some of the developed Arabic POS taggers. We have concluded with presenting our lexicon-free tagger for undiacritized Arabic, throwing light on the tagset we used to tag our corpus as well as the different techniques which we adopted in our approach for tagging Arabic.

As far as the Arabic POS tagger is concerned, it achieves 93.1% over a set of 97 tags. This tagger requires minimal manual intervention: we used a general purpose rule-based tagger T_{RB} which will work on any diacritized corpus, and we then manually corrected the output of T_{RB} on a set of 1100 words. Using a combination of the uncorrected output of T_{RB} on the 78,000 words in the Qur'an and the corrected tags on the Gold Standard, we were able to obtain a collection of conditional probabilities and a set of corrective rules which achieved a very respectable degree of accuracy. This is interesting in itself, since it shows that it is possible to tag even undiacritized Arabic reasonably accurately even without a large manually tagged corpus for training. The general approach also has applications in situations where you have a tagger which was trained on texts from one genre, but you want to adapt it for use in a new one. The distribution of words in one corpus may well be different from the distribution in another, so the existing tagger may not work well in the new domain. The steps taken for extracting T_{ML+TBL} from the output of T_{RB} are immediately adaptable to extracting a tagger from a corpus that has been already tagged. Recall that the original output of T_{RB} was just 75%, and that manually correcting a set of 1100 words from this corpus allowed us to achieve a final accuracy of 93.1%. If the initial tagger is more accurate than T_{RB}, as will often be the case, then the procedure outlined here should make it easy to adapt it so that it behaves well when used in a new setting.

As regards tagging the English text in the parallel corpus, we have used an English tagger that uses a tagset derived from the BNC basic tagset. The used tagset has been described above. Since the English tagger is not part of the contributions in this study, we do not attempt any evaluations of it. Nonetheless, looking at the tags in the corpus, it turned out that the tagger's accuracy is reasonable and thus sufficient for the basic task of finding translational equivalents in the corpus.

Having tagged the undiacritized corpus with the final tagger, we set out to write our shallow dependency parser for Arabic to be applied to the tagged corpus in order to extract the DRs between lexical items in the corpus. Likewise, we use the lexicon-free POS tagger for English described above to POS tag the translation of our parallel corpus. We also use a shallow dependency parser for English to get the DRs in the English translation. The description of the Arabic and English parsers will be presented in the following chapter.

Chapter 5

Dependency Relations

5.1 Introduction

It has been pointed out at the beginning of the thesis that in order to extract translational equivalents from the parallel corpus we carry out two types of annotation, namely POS tags and dependency relations (DRs). Both types of annotation are applied to both Arabic and English bitexts. Thus, we have built an Arabic POS tagger to tag the Arabic text and used an English tagger to tag the English translation in the corpus. Both taggers have been discussed in the previous chapter. In order to get the DRs between lexical items in the parallel corpus we had to write dependency parsers for Arabic and English. The Arabic dependency parser is not full or deep, but rather partial and shallow in two aspects.

(i) Parsers can be generally described as shallow or deep depending on how detailed their syntactic annotation is. The Arabic parser extracts the DRs from the tagged corpus without labelling them with grammatical functions. In other words, the parser gets the dependency attachment between predicates and their arguments without specifying the function in question, i.e. subject, object, modifier.etc. Thus, it is shallow in this sense.

(ii) Parsers can also be described as full or partial according to whether they produce partial parses or full parses, i.e. whether they generate hierarchical syntactic structure or not (Uí Dhonnchadha, 2008). We have focused on the main constructions in Arabic, leaving out other fine-grained constructions. In other words, we focus on those basic constructions that include a verb and following nouns as well as prepositional phrases. In a way we focus on phrase-like units that can be described as ‘chunks’. These phrase-like units may constitute a complete sentence or part of a sentence. Moreover, the parser does not cover co-ordination, long-distance dependencies and prepositional and clausal attachments. In this sense it is a partial parser.

There are two reasons for having a partial and shallow dependency parser for Arabic, which can be summed up as follows.

(1) As noted at the beginning of the current study, we do not use a lexicon of words. This, consequently, makes it extremely hard to do deep parsing, since such an attempt at deep parsing should have available some information about subcategorization frames or transitivity information on verbs, which are unavailable to us. This lack of cues precludes us from carrying out a deep dependency analysis.

(2) The special type of text we are experimenting with, as described in chapter 2, is another reason for having a partial parser. It has been made clear that this type of text has no punctuation marks which demarcate sentence boundaries. This makes it just nearly impossible to provide complete spanning parses.

As for the English parser, it is similarly a partial one, since we have removed punctuation marks from the English text to be similar to the Arabic text. We thus deal with phrase-like units not complete sentences. But when it comes to the notion of whether the English parser is a deep or shallow one like the Arabic parser, it can be described as a slightly deeper parser than the Arabic one, since we label the noun that precedes the verb in English as the ‘subject’ and the noun that follows it as the ‘object’. In other words, we use the syntactic cue of word order, which is relatively fixed in English, to label these two grammatical functions. But we do not distinguish between different types of subjects, such as subjects of simple declarative sentences, subjects of relative clauses or subjects of an infinitive. We also label the noun following a preposition as ‘object of preposition’. Apart from these labels, no other deep analysis for English is attempted. In actual fact, a further deep dependency analysis could be executed using the thematic relations, i.e. labelling the verb’s arguments with their thematic roles such as agent, patient, etc. (Fillmore, 1968).

The labelling of subjects and objects is extremely hard in the case of Arabic, since Arabic word order is relatively free, as will be discussed later. Also, Arabic is morphologically complex where clitics are attached to verbs. These clitics are syntactic units that could serve syntactic functions, as is the case with cliticized pronouns. These pronouns may be functioning as object pronouns. In addition, the rich agreement morphology of Arabic verbs allows for subjects to be dropped and could be recovered by such agreement features. This case of pro-drop subject will be illustrated when we talk about the Arabic syntax below. We do not include elliptical items such as the pro-drop subject in our attempt to extract DRs from the corpus. Only lexical items that are present in the surface structure are given a dependency analysis. It is worth noting that a similar approach of partial and shallow parsing has been carried out for other languages such as Irish (Uí Dhonnchadha, 2008).

Although both the Arabic and English parsers are not full ones, they are suitable for our current purpose of extracting a number of dependency pairs from the parallel corpus. In other words, we use the shallow dependency parsers to extract a number of 'head-dependent' translational pairs. Then we filter these pairs to obtain a number of one-word translation seeds so as to use them in our bootstrapping technique to resegment the parallel corpus to guide the proposer in a better way.

In this chapter we will discuss our syntactic framework which is based on dependency grammar. According to Ryding (2005), "Arabic can be seen as a language that has a network of dependency relations in every phrase or clause. These relations are key components of the grammatical structure of the language." As a matter of fact, there are two main approaches to syntactic analysis of a natural language. These two approaches are generally described as constituency-based and dependency-based. Our framework follows the second approach. The two approaches are basically different but there seem, however, to be common features between them. In order to grasp one or another of the two approaches, a contrast is sometimes made between them. Accordingly, we explore these two main approaches to syntactic analysis, focusing on the second approach, namely the dependency-based one, on which our framework is based. Before starting this discussion we will give a descriptive account of Arabic syntax, shedding light on the main sentence structure, construct phrases, and the phenomena of word order and agreement. We also discuss sources of syntactic ambiguity in Arabic. We conclude the chapter by describing both Arabic and English dependency parsers.

5.2 Arabic Syntax: A Descriptive Analysis

The way words are arranged together to form sentences is the concern of the linguistic discipline of syntax. Syntax, then, is the study of formal relationships between words (Jurafsky and Martin, 2009). In the following sections we will give a brief descriptive analysis of the syntactic phenomena in Arabic. We start with throwing light on some of the main characteristics of the Arabic language, and then proceed to describe the major Arabic syntactic structures.

As we have seen already, Arabic exhibits many complexities (Daimi, 2001; Chalabi, 2000), which makes Arabic language processing particularly difficult. Here is a summary of some of the major characteristics of Arabic that cause problems for language processing:

(i) The lack of diacritics and the complex morphological structure that we have seen so far lead to a vast degree of lexical ambiguity, which in turn makes syntactic analysis difficult.

(ii) Arabic is distinguished by its syntactical flexibility. It has a relatively free word order. Thus, the orders: SVO, VSO, VOS, OVS are all acceptable sentence structures in Arabic. The final word order OVS is used in Classical Arabic, but is uncommon in Modern Standard Arabic. Daimi (2001) emphasized that Arabic allows a great deal of freedom in the ordering of words in a sentence. Thus, the syntax of the sentence can vary according to transformational processes such as extraposition, fronting and omission.

(iii)In addition to the regular sentence structure VSO, Arabic has an equational sentence structure of a subject phrase and a predicate phrase, which contains no verb or copula (Attia, 2008).

(iv)Arabic is a clitic language. Clitics are morphemes that have the syntactic characteristics of a word but are morphologically bound to other words (Crystal, 2008). This phenomenon is very common in Arabic. These clitics include a number of conjunctions, the definite article, a number of prepositions and particles, as well as some pronouns. These clitics attach themselves either to the start or end of words. Thus, a whole sentence can be composed of what seems to be a single word. We have given an example from the Qur'an in chapter 1 to illustrate this point. Here is another similar example for a one word sentence that contains a complete syntactic structure.

5.1 أَنُلْزِمُكُمُوهَا

OanulozimukumuwhaA

Oa nu lozimu kumuw haA

should we impose you.pl it

"should we impose it on you" (Qur'an, 11:28)

These clitics are not common in English. There are some forms that are similar to clitics. In example (5.2) the possessive marker ('s) is considered a clitic.

5.2 *John's book*

(v) Arabic is a pro-drop language. This means that the subject in the sentence can be omitted. This situation causes ambiguity for any syntactic parser which has to decide whether the subject is present in the sentence or not.

(vi)According to Daimi (2001), there is no agreed upon and complete formal description of Arabic. It has been observed that there is no agreement among researchers on the classification of basic sentence structure in Arabic.

We start our descriptive analysis of Arabic syntax by giving an overview of the basic sentence structure in Arabic. This is followed by summarizing the basic DRs in a simple Arabic sentence, throwing light on two issues that add to the complexity of the basic structure of syntactic relations, i.e. verb-subject agreement and word order variation. Finally, we conclude with mentioning some sources of syntactic ambiguity in Arabic, which pose a challenge for any Arabic parser.

5.2.1 Basic Arabic Sentence Structure

Arabic grammatical tradition distinguishes between two types of sentence: (a) **verbal sentence** and (2) **nominal sentence**. Wright (1967) points out that a nominal sentence is one which begins with the subject, whether the predicate is another noun, a prepositional phrase or a verbal predicate. A verbal sentence, on the other hand, is one which starts with a verb (or one in which the verb precedes the subject). This classification follows traditional Arabic grammatical theory, where the division of sentences into these two categories depends on the nature of the first word in the sentence. Thus, if the first word is a noun, the sentence is nominal, and if it is a verb, the sentence is verbal (Ryding, 2005; Majdi, 1990).

Western researchers, however, have another classification of Arabic sentence structure. Ryding (2005), for instance, classified Arabic sentences into equational sentences and verbal sentences. The former type does not include a verb among its constituents, whereas the latter contains a verb. Thus, the criteria of the classification are different in Western academia from those applied among traditional Arabic grammarians. In the West, as Ryding (ibid) notes, researchers adopted a different criterion: the "distinction is based on whether or not the sentence contains a verb." If the sentence contains a verb, it is verbal, and if it does not contain a verb, it is equational.

Badawi et al. (2004) classified the Arabic basic sentences into three main types. The first type is equational sentences, which consist of subject + predicate only, and contain no verbal copula or any other verbal elements. An example of this type is the sentence الطريق طويل *AlTariyqu TawiylN* "the road (is) long". The second type is the topic + comment structure. This type also contains no verbal copula. In this type of sentence the topic is a noun phrase (NP) in the initial position and the comment is an entire clause (either an equational or verbal sentence, or another topic-comment sentence) anaphorically linked to the topic. Both the first and second types are traditionally labelled جملة اسمية *jumlap Asomiy~ap* "nominal sentence", because they begin with nouns (either as subject or topic). The third type is verbal sentences, which consist of a verb, always in the first position accompanied by the agent usually in the second position and the other complements usually in the third position.

5.2.1.1 Nominal Sentences

The first type of sentence is the nominal sentence. We will follow the traditional classification of Arabic sentence structure into nominal and verbal. But we follow Badawi et al. (2004) classification of nominal sentences into equational and topic-comment.

(i) Equational sentences: Arabic allows for sentences that have an NP as head and predication. In other words, an equational sentence consists of a subject and predicate, which are both in the nominative case. This type of sentence typically begins with an NP or pronoun. The predicate, on the other hand, can be an adjectival phrase (ADJP), an NP, adverbial phrase (ADVP) or a prepositional phrase (PP). The different types of subject and predicate are illustrated in the following examples.

5.3 الطالب ماهر

NP	ADJP
AlTaAlibu	*maAhirN*
the-student.sing.masc.nom	clever.sing.masc.nom

“The student is clever”

5.4 هو ذكي

PRO	ADJP
huwa	**akiy~N*
he	intelligent

“He is intelligent”

5.5 أبي طبيب

NP	NP
Oabiy	*TabiybN*
father.sing.masc-my	doctor.sing.masc

“My father is a doctor”

5.6 القلم هنا

NP	ADVP
Alqalamu	*hunaA*
the-pen	here

“The pen is here”

5.7 المدرس في المكتب

NP	PP	
Almudar~isu	*fiy*	*Almakotabi*
the teacher.sing.masc.nom	in-prep	the-office.gen

“The teacher is in the office”

As a matter of fact, the predicate does not always have to follow the subject. There are many constrained instances where the predicate can be fronted as in the following example.

5.8 في الدار رجل

PP NP

fiy *AldaAri* *rajulN*

in-prep the-house.gen man.indef.nom

“In the house there is a man”

These equational sentences are verbless because the Arabic verb كان *kaAna* “to be” is not normally used in the present tense indicative; it is simply understood (Ryding, 2005). According to Eid (1991), “Arabic, like many other languages, (e.g. Russian), does not have a present tense copula morpheme”. So, this type of sentence is often referred to as a zero copula. However, the verb كان *kaAna* “was/were” and its future form يكون *yakuwnu* “will be” are explicitly used to refer to the past and future actions (Fischer, 2002; Eid, 1991). In addition, when the sentence is negated in the present a copula verb must be explicitly expressed, as shown in examples (5.9-5.11).

5.9 كان الملك مريضا

VCOP.past NP ADJP

kaAna *Almaliku* *mariyDAF*

was the-king.sing.masc ill.sing.masc

“The king was ill”

5.10 سيكون الطعام جاهزا

VCOP.fut NP ADJP

sayakuwnu *AlTaEaAmu* *gaAhizAF*

will-be the-food.sing.masc ready.sing.masc

“The food will be ready”

5.11 ليس الملك مريضا

VCOP.neg NP ADJP

layosa *Almaliku* *mariyDAF*

is-not the-king.sing.masc ill.sing.masc

“The king is not ill”

It has been pointed out that both subject and predicate are in the nominative case. However, it is noticeable in examples (5.9-5.11) that when a copulative verb, i.e. كان *kaAna* "to be" and any of its sisters, comes at the beginning of a nominal sentence it changes the predicate to the accusative case and the subject remains in the nominative case. On the contrary, there are some particles which can precede the subject and change its case to the accusative while the predicate remains in the nominative case. These so-called external governors are the seven particles إن *Iin~a* "surely", أن *Oan~a* "that", لكن *lakin~a* "but", كأن *kaOan~a* "as if", ليت *layota* "if only", لعل *laEal~a* "perhaps" and لا *laA* "no" (Hassan, 2007). Example 5.12 sheds light on one of these particles.

5.12 إن الملك مريض

PART NP ADJP
Iin~a *Almalika* *mariyDN*
surely the-king.sing.masc.acc ill.sing.masc.nom
"Surely the king is ill"

(ii) Topic-Comment sentences: The topic is a noun phrase in the initial position and the comment is a clause which is always linked anaphorically to the topic by a pronoun, called الرابط *AlraAbiT* "lit. the (binding) element" in Arabic grammar (Badawi et al., 2004). This type of sentence has a strong resemblance to Western topicalization, since in both cases the grammatical and logical subjects may be different. However, the topic-comment structure in Arabic is a basic structure and not the result of any movement, fronting or extraction as is the case with the following English example (ibid.) *that film I have seen before*. According to Badawi et al. (2004), there are almost no restrictions on what may appear in topic position. The comment, on the other hand, may be either an equational or verbal sentence as shown in the following examples.

5.13 الحجرة التي أعمل فيها جوها خانق

NP1 NP2 ADJP
AlHujorapu *Al~ati OaEomalu* *fiyha* *jaw~uhaA* *xaAniqN*
the-room.sing.fem which work-I in it air-its suffocating
"The air of the room in which I work is suffocating".

5.14 الطالب يقرأ الكتاب

NP V NP

AlTaAlibu *yaqoraOu* *AlkitaAba*

the-student.sing.masc.nom read.pres.sing.masc.3 the-book.sing.masc.acc

“The student reads the book”

5.15 الدرس يكتبه الطالب

NP V+PRO NP

Aldarosu *yakotubuhu* *AlTaAlibu*

the-lesson.sing.masc write.pres-it.sing.masc.3 the-student.nom.sing.masc

“The lesson, the student writes it”

5.2.1.2 Verbal Sentences

As discussed above, verbal sentences are traditionally those sentences that start with a verb. The following structures start with a verbal constituent and are thus classified as verbal sentences:

5.16 كتب الكاتب

V NP

kataba *AlkaAtibu*

write.past.sing.masc.3 the-writer.sing.masc.nom

“The writer wrote”

5.17 كتبت الطالبة الدرس

V NP1 NP2

katabat *AlTaAlibapu* *Aldarsa*

write.past.sing.fem.3 the-student.sing.fem.nom the-lesson.acc

“The student wrote the lesson”

5.18 كتّب المدرس الطالب الدرس

V NP1 NP2 NP3

kat~aba *Almudar~isu* *AlTaAliba* *Aldarsa*

make to write.past.3 the-teacher.nom the-student.acc the-lesson.acc
"The teacher made the student write the lesson"

5.19 علّم الطالب على الكتاب

V NP PP
Eal~ama AlTaAlibu EalaY AlkitaAbi
mark.past.masc.3 the-student.masc.nom on-prep the-book.gen
"The student marked on the book"

5.20 أعطى المدرس الكتاب للطالب

V NP1 NP 2 PP
OaEoTaY Almudar~isu AlkitaAba lilTaAlibi
give.past.masc.3 the-teacher.nom the-book.acc to-the-student.gen
"The teacher gave the book to the student"

5.21 اعتقد المدرس أن الطالب يكتب الدرس

V NP COMPS
AEotaqada Al-mudar~isu Oan~a Al-TaAliba yakotubu Aldarsa
think.past.masc.3 the-teacher.nom that the-student.acc write.pres the-lesson.acc
"The teacher thought that the student was writing the lesson"

5.22 أخذ الطالب يكتب الدرس

V NP1 S
Oaxaza AlTaAlibu yakotubu Aldarsa
start.past.masc.3 the-student.nom write.pres.masc.3 the-lesson.acc
"The student started to write the lesson"

The above-mentioned examples have shown different verbal constructions, which differ according to the subcategorization frame of a given verb. Thus, some verbs are intransitive that require only a subject. Some others are transitive requiring one object or ditransitive requiring two objects. A third type of verbs may subcategorize for a whole sentence.

5.2.2 Construct Phrases

Arabic has a specific type of construction in which two nouns are linked together in a relationship where the second noun determines the first by identifying it, and thus the two nouns function as one phrase or syntactic unit. This construction is referred to in Arabic as إضافة *IDaAfap* "annexation", which is usually described in English as 'construct phrase', 'genitive construct' or 'annexation structure' (Ryding, 2005). In fact, English exhibits similar constructions, where two nouns occur together with one noun defining the other, as in *the Queen of Britain* and *Cairo's cafes*.

The first noun in an Arabic construct phrase, which is called مضاف *muDaAf* "the added", has neither the definite article nor nunation because it is in an 'annexed' state, determined by the second noun. However, the first noun, being the head noun of the phrase, can be in any case: nominative, accusative or genitive depending on the function of the *IDaAfap* unit in a sentence structure. The second or annexing noun, called مضاف إليه *muDaAf Ilayohi* "the added to", is marked either for definiteness or indefiniteness, and is always in the genitive case.

The two nouns in an Arabic construct phrase could have various semantic relationships. The following table lists some of these relationships (ibid).

Construct Phrases	**Gloss**	**Semantic Relationship**
مدينةُ القدسِ *madiynapu Alqudosi*	the city of Jerusalem	**Identity**: the second noun identifies the particular identity of the first.
زعماءُ القبائلِ *zuEamaA'u AlqabaA}ili*	the leaders of the tribes	**Possessive**: the first term can be interpreted as belonging to the second term.
كلَّ يَومٍ *kul~a yawomK*	every day	**Partitive**: the annexed term (first term) serves as a determiner to describe a part or quantity of the annexing term.
وصولُ الملكةِ *wuSuwlu Almalikapi*	the arrival of the queen	**Agent**: the second term is the agent or doer of the action.
رفعُ العلمِ *rafEu AlEalami*	The raising of the flag	**Object:** the second term is the object of an action.

صناديقُ الذهبِ *SanaAdiyqu Al*ahabi*	boxes of gold	**Content:** the first term denotes a container and the second term the contents of the container.
طائرةُ إنقاذٍ *TaA}irapu InoqaA*K*	a rescue plane	**Purpose:** the second term defines the particular purpose or use of the first term.

Table 5.1: Semantic relationships between nouns in Arabic construct phrases

In the previous table, the second noun in a construct phrase can be definite or indefinite.

5.2.3 Agreement & Word Order

Having described the basic sentence structure in Arabic, we set out to discuss a few issues that add to the complexity of the basic structure of syntactic relations. These have to do with verb-subject agreement and word order. Agreement or concord is defined by Ryding (2005) as the feature compatibility between words in a phrase or clause. This means that they match or conform to each other, one reflecting the other's features. Agreement is formally defined by Corbett (2001) as "systematic covariance between a semantic or formal property of one element and a formal property of another." He (ibid) uses a number of terms to distinguish between the elements involved. Thus, he uses the term 'controller' to refer to the element which determines the agreement, 'target' to refer to the element whose form is determined by agreement, and 'domain' to refer to the syntactic environment in which agreement occurs. In addition, when we indicate in what respect there is agreement, we are referring to agreement 'features'. For instance, number is an agreement feature that has the values: singular, dual, plural. This agreement environment can be diagrammed in the following figure (adapted from Corbett, ibid).

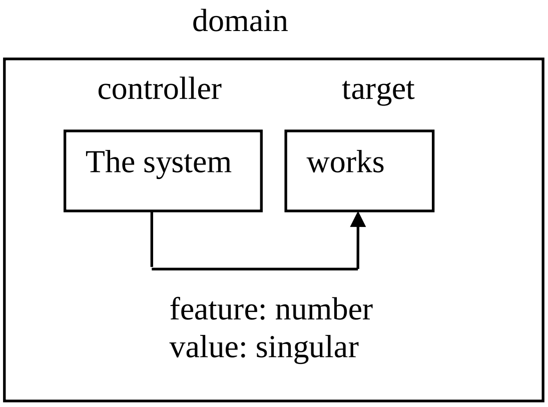


Figure 5.1: Description of agreement environment

According to Corbett (ibid), the relationship in agreement is generally asymmetrical because the target need not match all the features of the controller. A formal definition of the principle of asymmetric agreement is provided by Androutsopoulou (2001) as:

> "In an agreement relation between two elements α and β, where α is the head and β is the specifier, the set of agreeing features of β must be a subset of the set of agreeing features of α."

Platzack (2003) classified languages into 'uniform agreement' languages, where we find the same agreement independently of the position of the subject and 'alternate agreement' languages, where the finite verb only agrees in person, not in number, with the post-verbal subject. He (ibid) stated that Standard Arabic is a language with alternate agreement, where the verb shows full agreement in person, gender and number when the subject is in front of it, but partial agreement (only person and gender) when the subject follows the verb.

Accordingly, agreement in Arabic lies in its apparent dependence on the surface order of the subject and the verb (Mohammad, 1990). Thus, if subjects are in the pre-verbal position, verbs show full (rich) agreement with the subjects in the features of person, number and gender. If, on the other hand, subjects are in the post-verbal position, verbs show partial (weak or poor) agreement, as verbs agree with their subjects in gender and person only. In other words, they take the default singular form whether subjects are singular, dual or plural.

The rich agreement morphology that Arabic has allows it to show agreement relations between various elements in the sentence (Attia, 2008). The morphosyntactic features involved in agreement in Arabic are described in table (5.2) below.

Morphosyntactic Features	**Values**
Number	singular, dual and plural
Person	1st person, 2nd person and 3rd person
Gender	masculine and feminine
Case	nominative, accusative and genitive
Definiteness	definite and indefinite

Table 5.2: Morphosyntactic features involved in agreement in Arabic

An agreement relation can have one or more of the above-mentioned five morphosyntactic features. The strongest relation is that between a noun and a qualifying adjective, where four of the five agreement features are involved: number, gender, case and definiteness. This is shown in example (5.23).

5.23 جاء الطالبان الماهران (noun-adjective: number, gender, case, definiteness)

jaA'	*AlTaAlibaAni*	*AlmaAhiraAni*
come.past	the-student.dual.masc.nom	the-clever.dual.masc.nom

"The two clever students came"

As pointed out above, agreement between a verb and its subject differs according to their order in a sentence. Examples (5.24-5.25) show different word orders with different agreement features.

5.24 كتبت الطالبات الدرس (VSO)

katabat	*AlTaAlibaAtu*	*Aldarsa*
write.past.sing.fem.3	the-student.pl.fem.3.nom	the-lesson.acc

"The students wrote the lesson"

5.25 الطالبات كتبن الدرس (SVO)

AlTaAlibaAtu	*katabna*	*Aldarsa*
the-student.pl.fem.3.nom	write.past.pl.fem.3	the-lesson.acc

“The students wrote the lesson”

It is noticeable above that when the subject follows the verb, there is partial agreement between it and the verb, where the verb inflects only for person and gender but not number. Nevertheless, when the subject is in a pre-verbal position, it has full agreement with the verb with regard to the features of person, number and gender.

As far as word order is concerned, Arabic is characterized by its free word order. This is the case in both CA and MSA. Majdi (1990) gives the following examples to show word order variation in CA. The first three word orders are similarly common in MSA, but the final one, we believe, is uncommon.

5.26 اشترى سالمٌ كتابًا (VSO: Verb-Subject-Object)
A$otaraY *saAlimN* *kitaAbAF*
buy.past.sing.masc.3 Salim.nom book.acc
“Salim bought a book”

5.27 سالمٌ اشترى كتابًا (SVO: Subject-Verb-Object)
saAlimN *A$otaraY* *kitaAbAF*
Salim.nom buy.past.sing.masc.3 book.acc
“Salim bought a book”

5.28 اشترى كتابًا سالمٌ (VOS: Verb-Object-Subject)
A$otaraY *kitaAbAF* *saAlimN*
buy.past.sing.masc.3 book.acc Salim.nom
“Salim bought a book”

5.29 كتابًا اشترى سالمٌ (OVS: Object-Verb-Subject)
kitaAbAF *A$otaraY* *saAlimN*
book.acc buy.past.sing.masc.3 Salim.nom
“Salim bought a book”

It is worth noting that the feature of humanness plays an important role in agreement between targets and controllers in many varieties of Arabic. According to Belnap and Shabaneh (1992), with non-human plural controllers, targets are invariably in the singular and feminine form. The targets may be either verbs or qualifying adjectives as shown in examples (5.30) and (5.31) respectively.

5.30 الأرانب تأكل الجزر

AlOaraAnibu	*taOkulu*	*Aljazara*
the-rabbit.pl.masc.nom.3	eat.pres.sing.fem.3	the-carrot.acc

“The rabbits eat carrot”

5.31 السنوات الجديدة

AlsanawaAtu	*Aljadiydapu*
The-year.pl.fem.nom	the-new.sing.fem.nom

“The new years”

This phenomenon is referred to as ‘deflected’ as opposed to ‘strict’ agreement.

Having discussed Arabic sentence structure and the two related issues of agreement and word order, we can now, following Ryding (2005), summarize the basic **dependency relations** in a simple Arabic sentence with a verbal constituent as follows:

(i) The subject may be incorporated in the verb as part of its inflection.

(ii) The subject may also be mentioned explicitly, in which case it usually follows the verb and is in the nominative case. The verb agrees in gender with its subject.

(iii)A transitive verb, in addition to having a subject, also takes a direct object in the accusative case.

(iv)The basic word order is VSO.

(v) The word order may vary to SVO, VOS or even OVS under certain conditions.

5.2.4 Sources of Syntactic Ambiguity in Arabic

Broadly speaking, ambiguity is a linguistic phenomenon that is not restricted to a particular language (Hamada and Al Kufaishi, 2009). In other words, ambiguity is an inherent characteristic of any natural language, occurring at all levels of representation (Diab, 2003). Ambiguity prevails at different linguistic levels in

Arabic: lexical, structural, semantic and anaphoric. We will focus our discussion in this section on the structural (or syntactic ambiguity) in Arabic.

Syntactic ambiguity poses a major problem for any syntactic parser. The resolution of structural ambiguity is a central topic in NLP. A sentence is structurally ambiguous if it can have more than one syntactic representation. According to Daimi (2001), the problem of ambiguity in Arabic has not received enough attention by researchers, due to the particular characteristics of Arabic including its high syntactic flexibility. There are some sources that result in structural ambiguity in Arabic. We will discuss three ambiguity-generating areas in the Arabic language. These are ‘lack of diacritics’, ‘Arabic nature of pro-drop’ and ‘word order variation’.

5.2.4.1 Lack of Diacritics

It has been pointed out earlier in the thesis that modern Arabic is written without diacritics or short vowels. This, consequently, makes morphological and subsequently syntactic analysis highly ambiguous (Attia, 2008). We have pointed out in chapter 1 that a word in Arabic can have different pronunciations without any change of spelling due to the absence of diacritics. This results in many Arabic homographs which can have different POS categories and morphological features. Thus, the same homograph can be interpreted as a verb or noun. Also, a verbal form of a word can be either in the active or passive voice, and declarative or imperative form. In addition, some verbal forms have the middle letter doubled to make the verb in question causative, which does not appear in orthography. Even more some agreement morphemes on the verbs are ambiguous with regard to person and gender differences. All this can best be illustrated through the following examples.

5.32 أكل *Okl*	(verb vs. noun)
أكَلَ	أكْلٌ
Oakala	*OakolN*
“ate”	“eating”

5.33 ضرب *Drb*	(active vs. passive)
ضَرَبَ	ضُرِبَ
Daraba	*Duriba*

“hit”	“was hit”

5.34 راسل *rAsl*	(declarative vs. imperative)
رَاسلَ	رَاسلْ
raAsala	*raAsil*
“corresponded with”	“correspond with!”

5.35 كتب *ktb*	(non-causative vs. causative)
كتبَ	كتَّبَ
kataba	*kat~aba*
“wrote”	“made (someone) to write”

5.36 درست *drst*	(person and gender differences)		
درسْتُ	درسْتَ	درسْتِ	درسَتْ
darasotu	*darasota*	*darasoti*	*darasato*
studied.1.sing	studied.2.masc.sing	studied.2.fem.sing	studied.3.fem.sing
“I studied”	“You studied”	“You studied”	“She studied”

It is frequently the case that a single word-form can have a combination of all the above-mentioned types of ambiguities, as illustrated in figure (1.1) in chapter 1, which results in a higher level of ambiguity.

5.2.4.2 Arabic Pro-drop

We have observed in our discussion of Arabic sentence structure that some examples have an explicit NP in the subject position. However, sometimes the subject is not explicitly mentioned but implicitly understood as an elliptic personal pronoun (or a pro-drop). Arabic is, thus, a pro-drop language. The pro-drop theory (Baptista, 1995; Chomsky, 1981) stipulates that a null category (pro) is allowed in the subject position of a finite clause if the agreement features on the verb are rich enough to enable its content to be recovered. This pro-drop phenomenon, which is referred to as الضمير المستتر *AlDamiyr Almustatir* “elliptic pronoun”, is frequent in Arabic due to the rich agreement morphology that verbs have. In Arabic verbs conjugate for number, gender and person. This, in turn, enables the reconstruction of the missing subject.

As Ryding (2005) points out, “the subject pronoun is incorporated into the verb as part of its inflection.” The following example sheds light on this point.

5.37 يكتبون الدرس

V	(PRO)	NP
yakotubuwna	(*hum*)	*Aldarosa*
write.pres.pl.masc.3		the-lesson.acc

“They write the lesson”

It is worth noting that when an elliptic pronoun is present in an Arabic sentence it gives rise to a major syntactic ambiguity, leaving any syntactic parser with the challenge to decide whether or not there is an elliptic pronoun in the subject position (Chalabi, 2004b). According to Attia (2008), pro-drop ambiguity originates from the fact that many verbs in Arabic can be both transitive and intransitive. Thus, in case such verbs are followed by only one NP the ambiguity arises, as shown in example (5.38).

5.38 أكلت الدجاجة

V	NP
Oklt	*AldjAjp*
ate.fem	the-chicken

In the absence of diacritics, as pointed out by Attia (2008), we are not sure whether the NP following the verb in this example is the subject (in this case the meaning is ‘the chicken ate’) or the object and the subject is an elliptic pronoun meaning *she* and understood by the feminine mark on the verb (in which case the meaning will be ‘she ate the chicken’). This ambiguity is caused by two facts.

(i) There is a possibility for a pro-drop subject following Arabic verbs.

(ii) The verb أكل *Oakala* “to eat” can be both transitive and intransitive.

These two interpretations exhibit two different possible syntactic structures and could be represented in two different PS trees as follows.

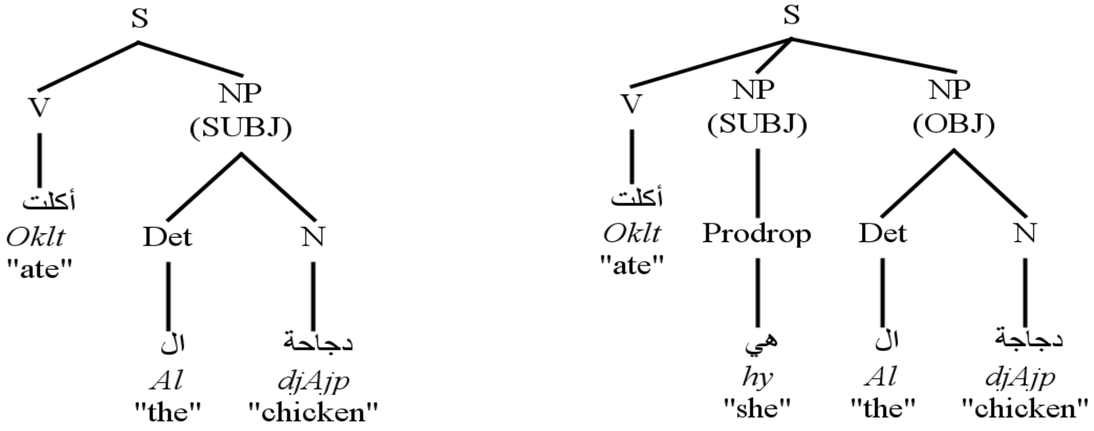


Figure 5.2: Different phrase structure trees for a possible pro-drop sentence

5.2.4.3 Word Order Variation

In section 5.2.3 we have shown the flexible nature of Arabic word order. Arabic word order is comparatively free, where a range of word orders is possible. Although the canonical order of Arabic sentences is VSO, Arabic allows also SVO, VOS and OVS orders. However, the final word order, i.e. OVS, is restricted to CA, and normally does not occur in MSA. This relatively free word order in Arabic causes many structural ambiguities. A parser does not find it easy to detect which order is meant in a given sentence, since all these different word orders are possible in a given sentence. This is because the distinction between nominative subject and accusative object is made through diacritics which are missing in MSA. Thus, whereas SVO order is easily detected by the parser, VOS gets mixed up with VSO. This means that every VSO sentence has a VOS reading, which causes a serious ambiguity problem (Attia, 2008). The following two undiacritized Arabic examples show the VSO and VOS orders that can cause this sort of structural ambiguity.

5.39 كتب الطالب الدرس (VSO sentence)

ktb	*AlTAlb*	*Aldrs*
write.past	the-student.nom	the-lesson.acc

"The student wrote the lesson."

5.40 كتب الدرس الطالب (VOS sentence)

ktb *Aldrs* *AlTAlb*

write.past the-lesson.acc the-student.nom

"The student wrote the lesson"

The variation in word order is also vivid in zero copula constructions. In zero copula constructions the subject normally comes before the predicate as in (5.41) below.

5.41 الرجل في البيت (Subj-Pred zero copula)

Alrjl *fy Albyt*

the-man in the-house

"The man is in the house"

However, this word order can be inverted, where the predicate precedes the subject. This occurs under certain constraints as in (5.42), where the subject is indefinite and the predicate is a prepositional phrase.

5.42 في البيت رجل

fy Albyt *rjl*

in the-house man

"In the house there is a man"

Unless the inversion of subject and predicate is constrained, it will lead to many ambiguities. In fact, zero copula constructions cause an ambiguity problem in our lexicon-free dependency parser, as will be illustrated later.

5.3 Main Approaches to Syntactic Analysis

Following the descriptive account of Arabic syntax in the previous section, we are going to shed light on the two main approaches to syntactic analysis in this section. As pointed out at the beginning of this chapter, the first approach is phrase structure analysis, which makes use of the notion of constituency, and the second one is dependency analysis, which underlies our syntactic framework. Since both approaches make use of different syntactic information, it is expedient to make a comparison between both approaches so as to fully grasp each of them. Then we will explore in detail the dependency framework which we adopt in our syntactic analysis.

Syntactic preprocessing can be differentiated with regard to the type of syntactic analysis it produces (Kermes, 2008). In this respect there are normally two main types of syntactic analysis. The first type is a phrase-structure or constituent-based analysis. The other type of analysis is a dependency structure analysis. According to Mel'čuk (1979), there is no other essentially divergent possibility. We consider it useful to start with throwing light on the phrase-structure analysis as a way of comparing it with the dependency structure analysis that we will describe later.

5.3.1 Phrase Structure Grammar (PSG)

This type of grammar has been introduced by Chomsky in a number of his writings. He initiated his theory in *Syntactic Structures* (1957) and then incorporated a number of modifications in his *Aspects of the Theory of Syntax* (1965). Chomsky (1965) clearly makes a fundamental distinction between two approaches to looking at language: a theory of language system and a theory of language use. These two approaches are what he refers to as **competence** and **performance** respectively[11]. Competence can be defined as "the speaker-hearer's knowledge of his language", whereas performance is "the actual use of language in concrete situations" (Chomsky 1965). He then proceeds to describe what a grammar of a language should be. According to Chomsky (1965), a grammar of a language is supposed to be a description of the ideal speaker-hearer's intrinsic competence. Thus, a fully adequate grammar is simply a system of rules that assign to each of an infinite number of sentences a structural description indicating how this sentence is understood by the ideal speaker-hearer. Karlsson (2008), quoting Chomsky (1965), points out that one way to test the adequacy of a grammar is to determine whether or not the sentences it generates "are actually grammatical, i.e. acceptable to the native speaker".

Language is not a mere sequence of words occurring next to each other in an unordered way. In other words, words are not strung together as a sequence of parts of speech, like beads on a necklace, but are organized into phrases to form a

[11] This distinction is related to the langue-parole distinction proposed by Saussure (1955).

sentence, following some constraints on word order. One basic notion in this regard is that certain groupings of words behave as constituents (Manning and Schutze, 1999). This notion is illustrated by Chomsky (1957) when he emphasizes that linguistic description on the syntactic level is formulated in terms of constituent analysis. The basic idea of constituency is that groups of words may behave as a single unit or phrase, which is called a constituent. Thus, a noun phrase (NP) may be defined as a sequence of words surrounding at least one noun (Jurafsky and Martin, 2009). Similarly, a verb phrase (VP) is a sequence of words that contain at least one verb. The following example illustrates this point.

5.43 The man ate the apple

In the sentence above, the constituents *the man* and *the apple* are noun phrases, while the constituent *ate the apple* is a verb phrase.

A set of rules has been devised to model the relationship between these phrases (constituents) called phrase structure rules. They are also referred to as rewrite rules. Each rule of the form X → Y is interpreted as "rewrite X as Y" (Chomsky, 1957). These rules are sometimes called productions. Here are some of these productions for English (adapted from Jurafsky and Martin, 2009 and Manning and Schutze, 1999):

S → NP VP
NP → (Det) Noun
NP → Proper Noun
NP → NP (PP)
PP → Prep NP
VP → V NP (PP)
Det → the
Det → a
Noun → man
Noun → butterfly
Noun → net
Verb → caught
Preposition → with

There are two types of symbols in these productions. The symbols that correspond to lexical items are called terminal symbols, while the symbols that express clusters of these are called non-terminal symbols. In the above simplified version of rules, the item to the right of the arrow is an ordered list of one or more terminals and non-terminals, while the one to the left of the arrow is a single non-terminal symbol.

Phrase structure trees are normally used to graphically illustrate the structure of a given sentence. In such trees one node dominates another when you can trace a path from the first node to the second one moving only downward through the tree (Poole, 2002). The previous productions can account for the following sentence.

5.44 The man caught the butterfly with a net.

The phrase structure tree (PST) of this sentence can be given a parse tree that looks as follows.

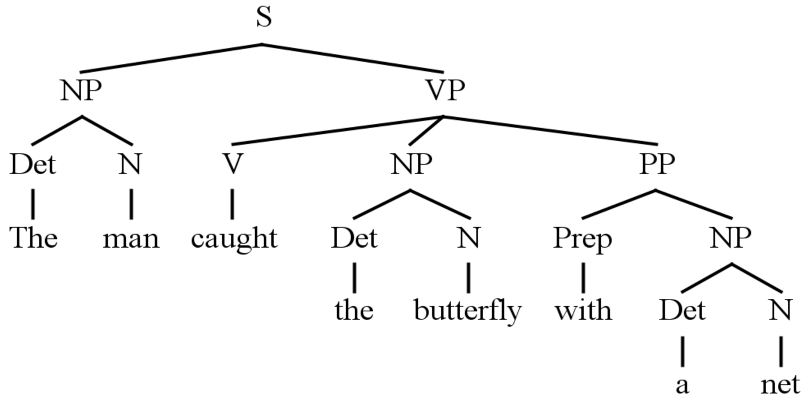


Figure 5.3: A phrase structure tree of an English sentence

As a matter of fact, constituency analysis comes from the structuralist tradition represented by Bloomfield (1933) and was formalized, as noted above, in the model of phrase structure grammar (PSG), or context-free grammar (CFG) (Chomsky 1957, 1965). A wide range of different theories about natural language syntax are based on constituency representations. PSG has been developed extensively since Chomsky's early work, (e.g. Chomsky, 1981, 1995). Within Computational Linguistics (CL) it has led to a new family of grammars termed 'unification grammar'. This includes frameworks that are prominent in CL, such as LFG (Kaplan and Bresnan, 1982), GPSG (Gazdar et al., 1985) and HPSG (Pollard and Sag, 1994).

5.3.2 Dependency Grammar (DG)

Besides PSGs another sort of grammar evolved, called dependency grammar (DG), which considered the concept of phrase unnecessary and embraced the view that linguistic structure is said to arise through the dependencies between words (Daniels, 2005). DG was developed by Tesnière (1959). It is distinct from PSGs, as it lacks phrasal nodes, i.e. all nodes are lexical. As we have seen before, in a constituency-based phrase-structure analysis, the focus is on the syntactic structure of language. This syntactic structure, according to generative theories, can be studied independently of meaning. This is best shown in Chomsky's (1957) famous example in (5.45) below.

5.45 Colorless green ideas sleep furiously.

Thus, we can judge a sentence to be syntactically good (i.e. well-formed), but semantically odd (i.e. meaningless). Mel'čuk (1988), a proponent of dependency analysis, describes this approach as "generate structures first, and ask questions about meaning later". Nonetheless, in a dependency-based analysis, there is a closer relationship between syntax and semantics. This is manifested in the dependency representation, since relations between pairs of words in a sentence are represented in terms of predicate-argument relations, or head-modifier relations. This use of lexical dependencies is an important aid to parsing (Jurafsky and Martin, 2009).

In dependency analysis structure is determined by the relation between a word (a head) and its dependents. DGs are not defined by a specific word order, unlike constituency-based analysis which is more heavily dependent on word order. Dependency analysis is, thus, well suited to languages with free word order, such as Arabic, Czech, etc. A dependency grammar is defined as a set of dependency rules, each of the form 'category X may have category Y as a dependent' (Daniels, 2005). Thus, within the context of dependency grammar, the above PST in figure (5.3) can be re-drawn to give the dependency tree (DT) in figure (5.4) below. In that way the difference between both types of grammar can be made clear.

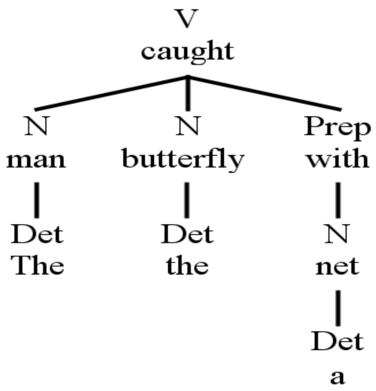


Figure 5.4: A dependency tree of an English sentence

In the previous diagram, the verb is the root node in the DT. Looking at this diagram, a number of dependency rules can be deduced to cover the sentence *the man caught the butterfly with a net* as follows.

V (N * N Prep)
N (Det *)
Prep (* N)
Det (*)
V: caught
N: man, butterfly, net
Det: the, a
Prep: with

The first three rules are called dependency rules, whereas the remaining rules are called assignment rules. The star * is used to indicate the place for the head of the whole construction.

5.3.2.1 The Notion of Dependency

The fundamental notion of dependency is broadly based on the idea that the syntactic structure of a sentence consists of binary asymmetrical relations between the words of the sentence (Nivre, 2006). The idea is expressed in the following way in the opening chapters of Tesnière (1959):

> "The sentence is an *organized whole*, the constituent elements of which are *words*. Every word that belongs to a sentence ceases by itself to be isolated as in the dictionary. Between the word and its neighbors, the mind perceives *connections*, the totality of which forms the structure of the sentence. The structural connections establish *dependency* relations between the words. Each connection in principle unites a *superior* term and an *inferior* term. The superior term receives the name *governor*. The inferior term receives the name *subordinate*. Thus, in the sentence *Alfred parle* [. . .], *parle* is the governor and *Alfred* the subordinate." [English translation by Nivre (2006)]

It is clear that a dependency relation (DR) holds between a **head** and a **dependent** or governor and modifier. In this respect, some words are habitually used with certain constructions, which in a sense they control or govern (Earl, 1973). These governing words impose syntactic constraints on words surrounding them (Robison, 1970). Thus, in a DR words depend on (or are governed by) other words. Generally speaking, the dependent is a modifier, object or complement; the head plays a more important role in determining the behaviours of the pair (Wu et al., 2009).

Criteria for establishing DRs, and for distinguishing the 'head' and the 'dependent' in such relations, are obviously of central importance for dependency grammar. Here are some of the criteria that have been proposed for identifying a syntactic relation between a head H and a dependent D in a construction C (Hudson, 1990):

1. H determines the syntactic category of C and can often replace C.
2. H determines the semantic category of C; D gives semantic specification.
3. H is obligatory; D may be optional.
4. H selects D and determines whether D is obligatory or optional.
5. The form of D depends on H (agreement or government).
6. The linear position of D is specified with reference to H.

We can notice that this list contains a mix of different criteria, some syntactic and some semantic.

Dependency grammar postulates rules for describing a given language. According to Hays (1964), a dependency rule is a statement about the valence of one kind of syntactic unit. The following notation (due to Gaifman, 1965) illustrates a dependency rule:

(5.1)

$$X (Y_1, Y_2, \ldots., *, \ldots., Y_n)$$

This means that $Y_1 \ldots . Y_n$ can depend on X in this given order, where X is to occupy the position *. The symbol *n* here refers to number. The following figure illustrates this dependency rule:

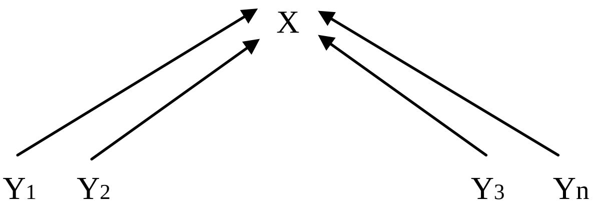


Figure 5.5: An illustrative figure of a dependency rule

Hays (1964) gives the following hypothetical English rule as an example:

(5.2)

$$V_\alpha \ (N_{p1}, \ *, \ N, \ D_\beta),$$

Where V_α is a class of verb morphemes, N_{p1} a class of plural nouns, N a noun class, and D_β a class of adverbs – say, of manner. This rule could be used in connection with utterances such as *children eat candy neatly*. However, we would like to point out that the position of the governing element in the previous notations is true of English. But when it comes to a language with a relatively free word order like Arabic, the governing element can occupy different positions. This means that it can precede all dependents, come in the middle or come after them. This is because word order in Arabic is more flexible than in English. The notation in (5.1) can be reinterpreted as providing a specification of constituency structure and canonical word order, where the left-to-right order in the rule need not be strictly enforced. This makes it possible to cope with the fact that Arabic allows a range of possible word orders, e.g. VSO, SVO, VOS and OVS.

Before elaborating on the dependency theory, we will draw a comparison between PSG and DG as far as rules are concerned. Robinson (1967) shows the difference between both theories with regard to rules and representation as follows:

PSG

Axiom: # S #

Rewriting Rules:

1- S → NP VP

2- VP → V NP

3- NP → D N

4- D → the

5- D → some

6- N → boys

7- N → girls

8- V → like

9- V → admire

DG

Axiom: * (V)

Dependency Rules:

1- V (N * N)

2- N (D *)

3- D (*)

Assignment Rules:

1- D: the, some

2- N: boys, girls

3- V: like, admire

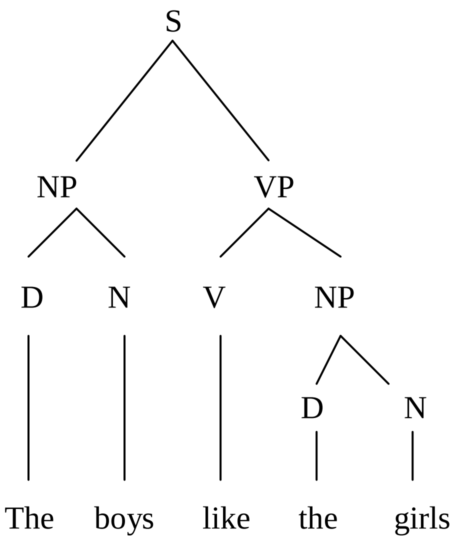


Figure 5.6: A phrase structure tree

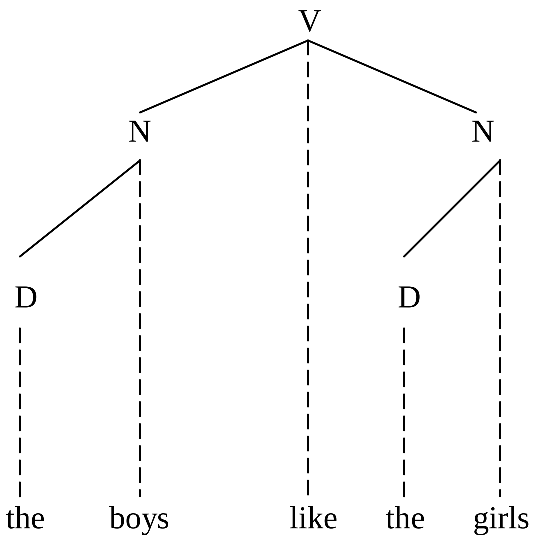


Figure 5.7: A dependency tree

In the dependency tree in the figure above solid lines represent dependency, while the dotted lines show the projection of each lexical item.

5.3.2.2 DG Notational Variants

As pointed out above, there are two ways to describe sentence structure in a given language: by breaking up the sentence into constituents (or phrases), which are then broken into smaller constituents, or by drawing links connecting individual words. These links refer to DRs between heads and dependents. Dependents that precede their heads are usually called ‘predependents’, whereas those which follow their heads are called ‘postdependents’ (Covington, 2001).

DRs can be analyzed in various notations. Notational variants of DG can be illustrated through the following example.

5.46 Tall people sleep in long beds.

Here are different notations for describing DRs. These representations can be shown in tree-like diagrams as in figure (5.8) or through using arrows as in figure (5.9) below.

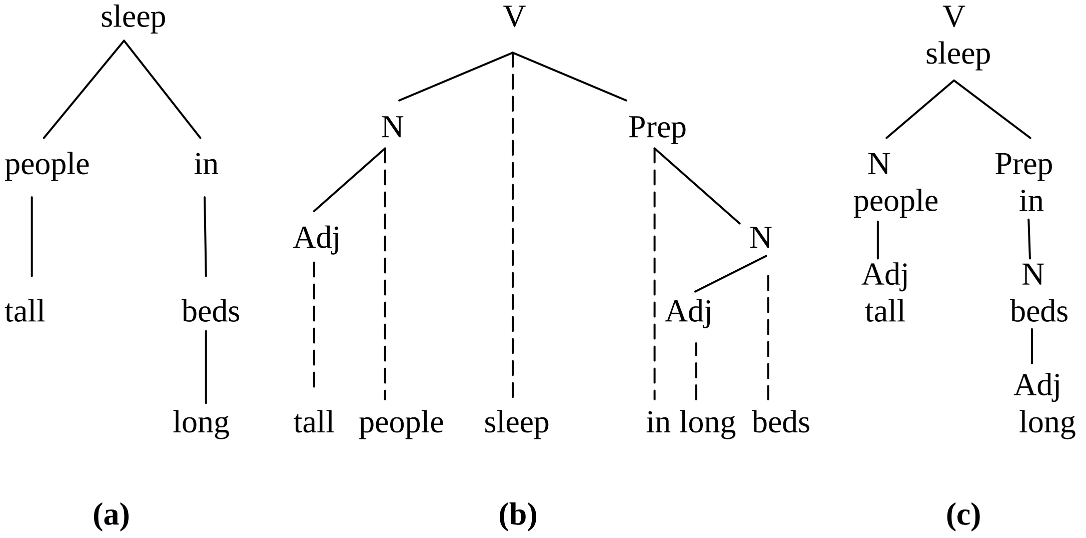


Figure 5.8: Different tree-like representations for dependency relations

The three previous diagrams show different ways of representing the DRs of a given sentence. Notably, we will use the type of diagram in (c) to represent the DRs when we discuss the shallow parsers for both Arabic and English. DRs can be also illustrated through graphs as shown in figure (5.9) below, where arrows point from head to dependent.

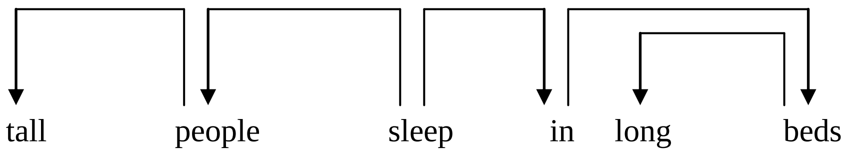


Figure 5.9: Dependency relations expressed through arrows (direction of arrow from head to dependent).

As a matter of fact, dependency relations can be labelled with grammatical functions (subject, object …etc.). This can be done using tree-like or arrow-like diagrams. Nivre (2006) gives example (5.47) below with an arrow-like diagram to show DRs along with labels for grammatical functions.

5.47 Economic news had little effect on financial markets.

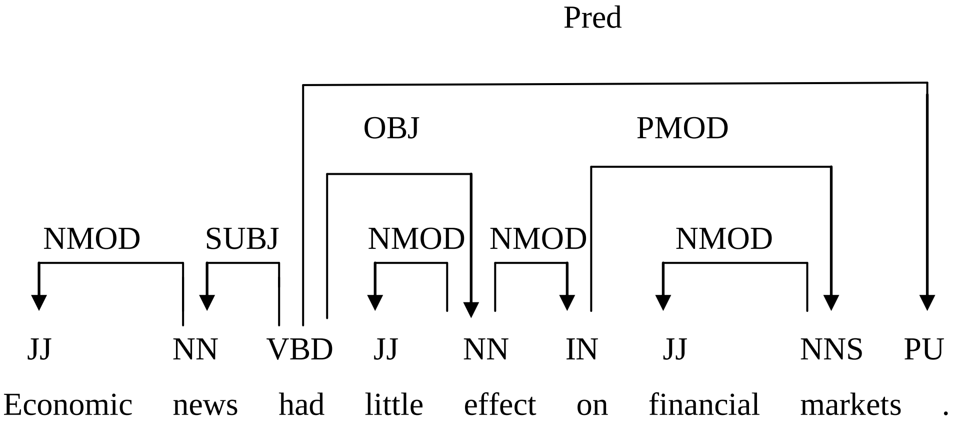

Pred
OBJ
PMOD
NMOD
SUBJ
NMOD
NMOD
NMOD
JJ NN VBD JJ NN IN JJ NNS PU
Economic news had little effect on financial markets .

Figure 5.10: Dependency relations along with grammatical functions

The above diagram illustrates the dependency relations, i.e. the heads and their dependents, for the above-mentioned sentence, where arrows point from head to dependent. Moreover, the grammatical functions are given in the structure. Thus, the word *news* is the subject (SUBJ), whereas the word *effect* is the object (OBJ). The abbreviation NMOD refers to a nominal modifier, while PMOD refers to a post-modifier. As for the abbreviation Pred, it refers to the predicate, i.e. *had little effect on financial markets* is the predicate of the whole sentence.

Various theories have emerged under the DG framework. These theories use different representations for DRs. Among the well-known theories in the field are the MTT (Mel'čuk, 1988) and WG (Hudson, 1984; 1990). In general, Mel'čuk (1988) describes a multistratal dependency grammar, i.e. one that distinguishes between several types of DRs (morphological, syntactic and semantic) (Buchholz and Marsi, 2006). As regards the syntactic dependency, which concerns us here, Mel'čuk (1988) points out the following:

(i) Syntactic dependency is universal. There is no language that does not have syntactic dependency.

(ii) Syntactic dependency cannot be bilateral. If, in a sentence, the word-form *w1* depends syntactically on *w2*, then in this sentence, *w2* can never be syntactically dependent on *w1*.

(iii) In a sentence, any word-form can depend syntactically on only one other word-form; this is the uniqueness of the syntactic governor. Thus, **w2* → *w1* ← *w3* cannot occur.

As for WG theory, as the name suggests, it recognizes words as basic elements of syntactic structures. It makes no reference to any unit longer than the word (except for the unit 'word-string' which is only used when dealing with coordinate structures) (Hudson, 1990; 2007). With regard to syntactic dependency, WG focuses on the 'companion' relation between words which occur together. As a matter of fact, the relation of companion is more than mere co-occurrence: it is a matter of co-occurrence sanctioned explicitly by the grammar. This point is shown in the following example.

5.48 She has brown eyes.

There will be entries in the grammar that specifically allow *she* and *has* to co-occur, but none which allows *has* and *brown* to co-occur; rather, *brown* is allowed to occur with words like *eyes*, and the latter are allowed to occur with words like *has*. Thus, each of these pairs are 'companions' of one another, but *has* is not a companion of *brown*. It is customary to add 'directionality' to the companion relation, so that one companion in each relation is labelled as 'head' to distinguish it from the other. In this way such relations are described in terms of a dependency structure that can be shown in the following diagram, with arrows pointing from head to modifiers (or dependents).

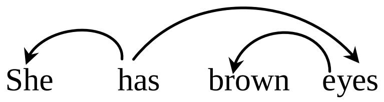


Figure 5.11: Dependency relations as described by WG

For more details about MTT and WG, the reader is referred to Mel'čuk (1988) and Hudson (1984; 1990).

It should be pointed out that different DG-based theories differ with regard to their analysis or representation of DRs. All the main differences between these theories are explained by Nivre (2006). It is not our concern here to discuss the differences between these theories.

It is worth noting that the DG tradition can reasonably be described as the indigenous syntactic theory of Europe. It was adopted as the basis for the European machine translation system EUROTRA (Hudson, 1990). It should be noted that dependency-based grammars and constituency-based grammars converge on some issues, e.g. the distinction between valency-bound and valency-independent constructions. This distinction is implemented in terms of subcategorization in HPSG theory (Bröker, 1997). Furthermore, according to Mel'čuk (1988), a number of PSG-oriented theories employ grammatical or dependency relations. Thus, in LFG (Kaplan and Bresnan, 1982) grammatical roles are expressed in the functional-structure or 'f-structure'. In HPSG (Pollard and Sag, 1994) subcategorization frames are used, and in Case Grammar (Fillmore, 1968) semantic dependencies are used. Generally speaking, as Gaifman (1965) points out, both DGs and PSGs supply the sentences that they analyze with additional structure; there is a very close relationship between these structures.

5.3.2.3 Conversion from DG Analysis to PSG Analysis

Converting a set of dependency structure (DS) annotations into phrase structure (PS) or vice versa means that we want to obtain a representation which expresses exactly the same content (Rambow, 2010). This is frequently done these days as there is a growing interest in dependency parsing but many languages only have PS treebanks. It is normally possible to convert a dependency structure analysis into a constituent analysis so long as the 'Head' of the structure in question is known. Thus, the following sentence is first given its dependency structure then transferred into constituent structure.

5.49 The boy put a book on the table.

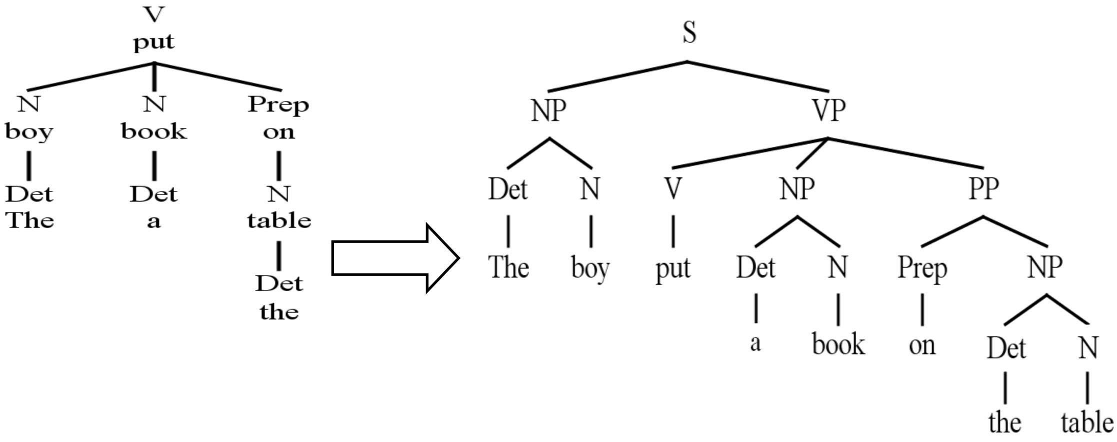


Figure 5.12: Conversion of a dependency tree into a phrase structure tree

Notably, the opposite way of converting a constituent structure into a dependency structure is also possible if the head is known in the constituent structure in question. Xia and Palmer (2001) express the same idea as they point out that once the heads in phrase structures are found, the conversion from phrase structures to dependency structures is straightforward. Conversion from **projective** dependency trees to phrase structure trees is also easy.

5.3.2.4 The Notion of Valency

The theory of **valency** (or valence) was first presented by Tesnière (1959) to capture the observation that the verb can be said to determine the basic structure of its clause. In other words, valency is seen as the capacity of a verb to combine with other sentence constituents, in a way similar to that in which the valency of a chemical element is the property to combine with a fixed number of atoms of another element (Platzack, 1988). Tesnière (1959) developed the notion of valency within the dependency grammar framework, where the verb was seen as the item on which the rest of the sentence depends. According to Platzack (1988), the elements occurring together with the verb in a clause are divided by Tesnière (1959) into two types: **actants** and **circonstants**. The circonstants are typically adjuncts, referring to the different aspects of the setting of the action or state of affairs referred to by the verb. They are not directly dependent on the verb, and therefore lie outside valency. It is thus the number of actants, which are immediately dependent on the verb, that constitute the valency of the individual verb.

It has been pointed out above that valency is a central notion in the theoretical tradition of DG. It is similar to the subcategorization notion in HPSG. In general, valency is based on the idea that in the lexicon each word specifies its daughters. According to Nivre (2005; 2006), the idea is basically that the verb imposes requirements on its syntactic dependents that reflect its interpretation as a semantic predicate. The terms valency-bound and valency-free are used in the DG literature to make a rough distinction between dependents that are more or less closely related to the semantic interpretation of the head. In figure (5.10), which describes the dependency structure of the sentence (5.47)**,** the subject and the object would normally be treated as valency-bound dependents of the verb *had*, while the adjectival modifiers of the nouns *news* and *markets* would be regarded as valency-free. The prepositional modification of the noun *effect* may or may not be treated as valency-bound, depending on whether the entity undergoing the effect is supposed to be an argument of the noun *effect* or not.

As far as heads and dependents are concerned, there is some agreement on some relations but not on others. Thus, while most head-complement and head-modifier structures have a straightforward analysis in dependency grammar, there are also many constructions that have a relatively unclear status. Such constructions include grammatical function words, such as articles, complementizers and auxiliary verbs, as well as structures involving prepositional phrases. For these constructions, there is no general consensus in the DG tradition as to whether they should be analyzed as ‘head-dependent’ relations at all and, if so, what should be considered the head and what should be considered the dependent. For example, some theories regard auxiliary verbs as heads taking lexical verbs as dependents; other theories make the opposite assumption; and yet other theories assume that verb chains are connected by relations that are not dependencies in the usual sense (Nivre 2005; 2006).

With respect to valency, verbs are classified into zero-valent if they have no actants that depend on them, mono-valent if they have one actant, di-valent if they have two or tri-valent if they have three actants (Somers, 1987). This applies to both English and Arabic. Examples for the types of valency for English verbs can be given as follows:

(1) **Zero-valent**: rain, snow.

(2) **Mono-valent**: come, laugh, cry.

(3) **Di-valent**: see, love, hate.

(4) **Tri-valent**: give, send, buy.

Allerton (1982) points out that at first sight it appears that English has no tetra-valent verbs. However, he (ibid.) claims that there are some exceptions where a verb can be said to be tetra-valent as the case with the verbs *charged* and *paid* as shown in the following example.

5.50 The firm {charged} Oliver a large sum for the job.
{paid}

In (5.50) *Oliver* seems to be the indirect object, and *a large sum* the direct object. In this way the verbs *charge* and *pay* would fit a standard type of trivalent pattern, with the exception of the final prepositional phrase *for the job*. According to Allerton (ibid.), it is still debatable whether this prepositional phrase should be recognized as a class of adverbial or be assigned to the valency of the verb and in this case such verbs will be described as tetra-valent.

Arabic verbs, on the other hand, can be classified as follows:

(1) **Zero-valent:** تمطر *tumoTir* “to rain”

(2) **Mono-valent**: جاء *jaA'* “to come”, ضحك *DaHika* “to laugh”, بكى *bakaY* “to cry”.

(3) **Di-valent**: رأى *raOaY* “to see”, أحب *OaHab~a* “to love”, كره *kariha* “to hate”.

(4) **Tri-valent**: أعطى *OaEoTaY* “to give”, أرسل *Oarosala* “to send”, اشترى *A$otaraY* “to buy”.

(5) **Tetra-valent**: أعلم *OaEolama* “to let (someone) know”, خبَّر *xab~ara* “to let (someone) be informed”.

According to Herslund (1988), valency should be stated in terms of ‘Grammatical Relations’ (GRs). However, he argues, not all kinds of GRs belong to the valency of verbs. He follows Tesnière (1959) in his classification of the elements that occur with the verb in a clause into clausal complements (i.e. adjuncts) and verbal complements (i.e. subjects and objects). Adjuncts are called ‘circonstants’ according to Tesnière or circumstants according to Halliday (1970). According to Herslund (1988), only verbal complements belong to a verb’s valency. As for the adjuncts, which are typically adverbial phrases of place and time, they belong to the entire clause and do not subcategorize any verb.

5.3.2.5 Dependency Parsing & Free Word Order Languages

Broadly speaking, dependency parsing of natural language texts may be either grammar-driven or data-driven. In a grammar-driven approach, sentences are analyzed by constructing a syntactic representation in accordance with the rules of the employed grammar. Most of the modern grammar-driven dependency parsers parse by eliminating the parses which do not satisfy the given set of constraints. Some of the known constraint based parsers are Karlsson et al. (1995), Tapanainen and Järvinen (1997), and more recently, Debusmann et al. (2004) which provides a multi-stratal (or multi-dimensional) paradigm to capture various aspects of a language. Data-driven parsers, in contrast, use a corpus of pre-analyzed texts (e.g. a treebank) to induce a probabilistic model for proposing analyses for new sentences (Nivre 2005, 2006).

Although dependency analysis is particularly useful for dealing with free word order languages, it is applicable to any language. For example, dependency parsers have been developed for a number of languages led by English. For instance, Nivre and Scholz (2004) describe a deterministic dependency parser for English based on memory-based learning. In addition, Nivre et al. (2006) present a language-independent system for data-driven dependency parsing, called MaltParser, where it has been evaluated empirically on Swedish, English, Czech, Danish and Bulgarian. Afterwards, MaltParser has been updated and evaluated on ten different languages: Bulgarian, Chinese, Czech, Danish, Dutch, English, German, Italian, Swedish and Turkish. These ten languages represent fairly different language types, ranging from Chinese and English, with poor morphology and relatively inflexible word order, to languages like Czech and Turkish, with rich morphology and relatively flexible word order, and with Bulgarian, Danish, Dutch, German, Italian and Swedish somewhere in the middle (Nivre et al., 2007). In addition, partial dependency parsers have been developed for a number of languages. As a case in point, a partial parser based on dependency analysis has been carried out for Irish (Uí Dhonnchadha, 2008).

Parsing morphologically rich, free word order languages is a challenging task. It has been maintained that free word order languages can be handled better using the dependency based framework than the constituency based one (Hudson, 1984, Mel’čuk, 1988). As it was made clear above, the basic difference between a constituency-based representation and a dependency representation is the lack of non-terminal nodes in the latter. Dependency analysis has been developed for the Arabic language. Ramsay and Mansour (2004) developed a rule-based syntactic parser for Arabic called PARASITE. This parser is based on a combination of Head-driven Phrase Structure Grammar and Categorial Grammar but outputs dependency trees for Arabic sentences. Dependency treebanks have also been developed for Arabic. Among the well known in the literature are Prague Arabic Dependency Treebank (PADT) (Smrž and Hajič, 2006; Smrž et al., 2008) and Columbia Arabic Treebank (CATiB) (Habash et al., 2009). There are some differences between these two dependency treebanks. Some of the differences are highlighted by Habash et al. (2009) in the following example.

5.51 خمسون الف سائح زاروا لبنان وسوريا في ايلول الماضي

xmswn Alf sA}H zArwA lbnAn wswryA fy Aylwl AlmADy

"50 thousand tourists visited Lebanon and Syria last September"

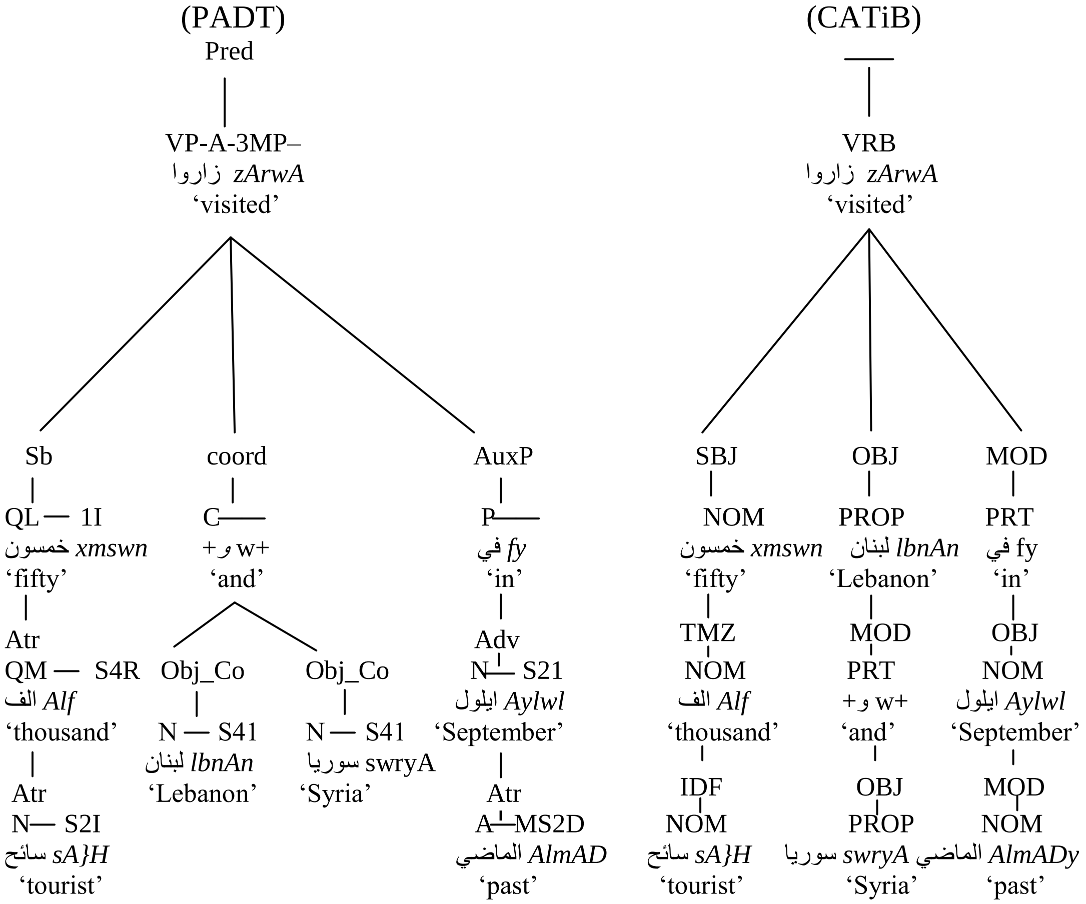


Figure 5.13: Dependency representation in both the Prague Arabic Dependency Treebank (PADT) and the Columbia Arabic Treebank (CATiB)

It should be noted that the above PADT's representation refers only to PADT's analytical level and not PADT's deeper tectogrammatical level. There are some differences between PADT and CATiB representations as shown in the above figure. First, Habash et al. (2009) indicate that PADT analytical labels are generally deeper than CATiB labels. This is because the analytical labels in PADT are intended to be a bridge towards the PADT tectogrammatical level. For instance, we can notice that dependents of prepositions are marked with the relation they have to the node that governs the preposition. Thus, in the above figure we can see that ايلول *Aylwl* "September" is marked as Adv (i.e. Adverbial) of the main verb زاروا *zArwA* "visited". Likewise, the coordinated elements لبنان و+سوريا *lbnAn w+swryA* "Lebanon and Syria" are both marked as Co (i.e. coordinated) and as Obj (i.e. object) with their relationship to the governing verb. Second, CATiB distinguishes different types of nominal modifiers, such as adjectives, idafa (i.e. annexation) and tamyiz (i.e. specification). PADT, on the other hand, does not make this distinction and marks all types as Atr (i.e. Attribute). Third, the other main difference that we can notice between both PADT and CATiB is that in PADT the coordination conjunction heads over the different elements it coordinates. CATiB adopts a different approach in this regard, as the conjunction is treated as a modifier for the first conjunct and as a head of the second conjunct.

We can then conclude that dependency structures could be represented differently, where some representations may be deeper than others. Every representation is done to meet a specific requirement. This conclusion points to the way we have done our Arabic dependency parser, which produces shallow unlabelled (or unnamed) dependencies owing to the fact that it is lexicon-free. But it is, still, fairly satisfactory for our current purpose of finding syntactically related words to improve the proposer.

It is worth noting that a dependency treebank is being developed for the Qur'an to allow researchers interested in the Qur'an to get as close as possible to the original Arabic text and understand its intended meanings through grammatical analysis (Dukes and Buckwalter, 2010; Dukes et al., 2010). The Qur'anic Arabic Dependency Treebank (QADT)[12] provides two levels of analysis: morphological annotation and syntactic representation using traditional Arabic grammar known as إعراب *IiEoraAb*.

[12] The Qur'anic Arabic Corpus is an online resource which is available at: http://corpus.quran.com/

The syntactic representation adopted in the treebank is a hybrid dependency/constituency phrase structure model. This is motivated by the fact that the treebank follows traditional grammar and this type of representation is flexible enough to represent all aspects of traditional syntax. Thus, the syntactic representation in QADT is done using dependency graphs to show relations between words, but relations between phrases are also shown by non-terminal nodes (Dukes et al., 2010). Figure (5.14) below gives an example of QADT representation.

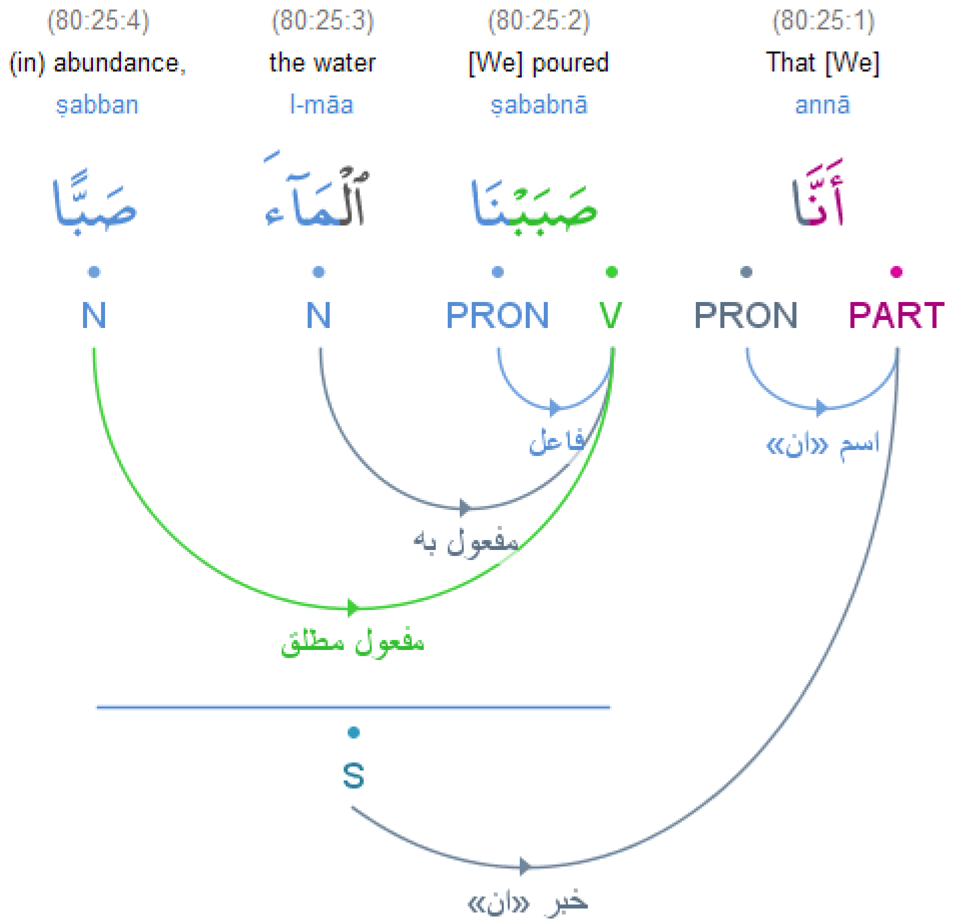


Figure 5.14: A hybrid dependency graph for verse (80:25) of the Qur'an in the Qur'anic Arabic Dependency Treebank (QADT) project

It is noteworthy that a dedicated team of Qur'anic Arabic experts have reviewed the morphological and syntactic annotation in the QADT. Moreover, the project is verified online via collaborative annotation through volunteer corrections (Dukes and Buckwalter, 2010).

5.4 Arabic Lexicon-Free Dependency Parser

We now present our dependency parser for Arabic. We should recall that the purpose is to find syntactically related words to guide the proposer. The Arabic parser produces unlabelled DRs, because of the lack of fine-grained morphology and the absence of a lexicon. This means that the parser outputs the dependency 'head-dependent' attachment without labelling the grammatical function in question, such as subject, object, modifier…etc. We obtain the DRs from the Arabic corpus which is now tagged by the Arabic tagger. We will begin with introducing the advantages of dependency parsing then describe the parser below.

5.4.1 Introduction

Parsing is an important preprocessing step for many NLP applications and thus of considerable practical interest (Buchholz and Marsi, 2006). Parsers use different grammatical formalisms: some of them exploit constituency grammar, while others use dependency grammar. Dependency parsing offers some advantages, which Covington (2001; 1990) outlines as follows:

- Dependency relations are close to the semantic relationships needed for the next phase of analysis. In fact, it is not necessary to read off 'head-dependent' relations from a tree that does not show them directly.
- The dependency tree contains one node per word. Because the parser's job is only to connect existing nodes, not to posit new ones, the task of parsing is, in a way, more straightforward. In this way the task is easier to manage.
- Dependency parsing lends itself to word-at-a-time operation. This means that parsing is carried out by accepting and attaching words one at a time rather than by waiting for complete phrases.
- Dependency parsing is advantageous in languages where the order of words is free.

According to Abney (1989), most top-down PS parsers introduce spurious local ambiguity. This is absent in dependency parsing, which attaches words one at a time, and does not wait for complete phrases (Covington, 2001). Consider, for example, a grammar which includes the rules VP $\longrightarrow$ V NP PP and VP $\longrightarrow$ V NP. When encountered with a sentence beginning *John found*, a top-down constituency parser builds the following structure, and attempts to expand VP:

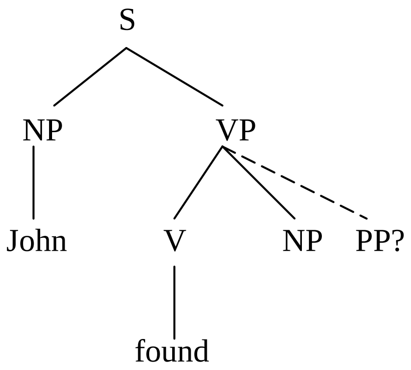


Figure 5.15: A top-down constituency parser's attempt to expand VP

Nevertheless, the parser has insufficient information to determine whether it will accept the first or second expansion. Therefore, it must guess, and backtrack if it guesses wrong; that is spurious local ambiguity. This means that it will be forced to backtrack on a sentence like *Mary found the book* or *Mary found the book on the table*. Similarly, a bottom-up constituency parser cannot construct the verb phrase until all the words in it have been encountered. Dependency parsers, in contrast, accept words and attach them with correct grammatical relations as soon as they are encountered, without making any presumptions in advance (Covington, 2001).

Our syntactic framework is a shallow rule-based one, which is conceptualized by using dependency grammar, in which linguistic structure is described in terms of dependency relations among the words of a sentence; it does so without resorting to units of analysis smaller or larger than the word. We utilize some information about syntax in both Arabic and English without requiring a full parse in either language. There are some advantages of not relying on full parses, which include, according to Schafer and Yarowsky (2003), the following:

(1) Unavailability of parsers for many languages of interest.

(2) Parsing time complexity represents a potential difficulty for both model training and testing.

As regards the Arabic parser, we have written a number of rules that cover the basic sentence structure in Arabic. These rules are represented in the following section. It is worth noting that we make use of regular expressions in our parser, since they can be applied quickly.

5.4.2 Arabic Dependency Relations

As indicated before, the main goal of writing this dependency parser for Arabic is to find syntactically related words in the parallel corpus to be used as **translation pairs** to resegment the corpus and bootstrap the selection process. The basic idea behind this activity is that having shorter verses could improve the proposer's accuracy.

Syntactic annotation in the dependency framework involves two types of interrelated decisions: attachment and labelling (Žabokrtský and Smrž, 2003; Habash and Rambow, 2004; Tounsi et al., 2009). First, the attachment of one word to another indicates that there is a syntactic relationship between the head (or governing) word and the dependent (or governed) word. Second, the labels (or relations) specify the type of attachment (Habash et al., 2009).

We have mentioned at the beginning of this thesis that we deliberately took a decision at the very outset of our project not to have a lexicon of words. Accordingly, in our approach to Arabic dependency parsing we deal only with attachment. In other words, we get the syntactic relationship between the head word and the dependent word. Thus, we say that a given word is the head in a given construction and the other words are dependents on this head. But we do not specify the type of such relationship. Thus, we do not specify whether a given dependent is a subject, object or modifier of a given head in a certain construction. The following example makes this point clearer.

5.52 كتبت الطالبة الدرس

ktbt	*AlTAlbp*	*Aldrs*
write.past.sing.fem.3	the-student.sing.fem.nom	the-lesson.acc

"The student wrote the lesson."

In this previous example the head word is the verb كتبت *ktbt* "wrote" and the dependents are the two nouns الطالبة *AlTAlbp* "the student" and الدرس *Aldrs* "the lesson". The Arabic dependency parser gets this relationship without specifying that the noun الطالبة *AlTAlbp* is the subject and the noun الدرس *Aldrs* is the object in this sentence. Furthermore, it should be made clear that the definite article ال *Al* "the" is dependent on the noun attached to it. But since we do not have a separate tag for the definite article, we treat it as part of the following noun. The dependency relation

(DR) in this sentence can be illustrated with the following dependency tree (DT), using the tagset we used to tag the Arabic corpus.

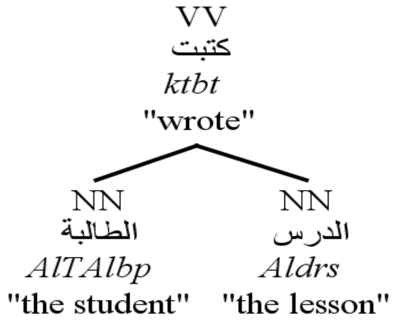


Figure 5.16: An unlabelled dependency tree for an Arabic sentence

The Arabic parser gets only the dependency attachment without labelling it because it is so difficult to label the grammatical functions in the current project for the following reasons.

(i) There is no use of a lexicon that includes the subcategorization frames or valency for verbs. When such frames are identified, we can know whether a given verb is intransitive, transitive, ditransitive or tritransitive (as some verbs in Arabic take three objects). In this way we can label grammatical functions such as ‘subject’, ‘object’, etc. However, we do not have this information to incorporate into the parser.

(ii) Arabic is a relatively free word order language, where objects can precede subjects. This is complicated with the absence of a lexicon, which makes deep parsing more difficult.

(iii) Arabic is a pro-drop language. This means that zero subjects are common in Arabic sentences. The subject of a sentence is normally omitted when it can be reconstructed due to the rich agreement morphology that characterizes verbs in Arabic. It has been shown that Arabic verbs conjugate for number, gender and person. In our analysis we cannot detect zero subjects, as it is extremely difficult to do this owing to the constraints under which the current research is being conducted.

(iv) Arabic is morphologically rich, and often a single word will consist of a stem with multiple fused affixes and clitics. Each of these morphological segments is assigned a part-of-speech, which makes syntactic dependencies between morphological word-segments a unique complexity not found in other languages such as English. This is particularly clear in the Qur'anic language (Dukes et al., 2010), which is the corpus we are using.

(v) We have discussed in chapter 2 the nature of the text we are dealing with, i.e. the Qur'an. We have referred to the specific nature of this text and the difficulties involved for our project. These features which characterize the Qur'anic text also affect the way we do parsing, since it is very difficult to deeply parse unpunctuated text, let alone under the lack of a lexicon.

Accordingly, the dependency analysis of Arabic is shallow and partial. We have discussed the way it is both shallow and partial at the beginning of this chapter.

In the coming sections we describe the work carried out on shallow parsing of Arabic using rule-based dependency analysis. We deal with various types of syntactic relations. We will discuss every type with illustrative examples in the following lines. We have used a number of patterns in our dependency parser to cover the major syntactic constructions. We use regular expressions (REs) to compile patterns. We will throw light on the use of REs to compile verbal and nominal patterns in the parser. The same REs are used to compile other patterns of interest. Thus, we will start with discussing the REs that are used to compile verbal and nominal patterns then discuss every pattern in detail below.

5.4.2.1 Regular Expressions (REs) Patterns

As stated above, we use REs to compile patterns for a number of constructions to be matched with the Arabic tagged corpus, which is now undiacritized. The following figure shows an excerpt from the Arabic tagged corpus against which we match our patterns.

```
(::,newverse,13099)(*lk,DEMO,13100)(AlktAb,NN,13101)(lA,PART,13102
)(ryb,NN,13103)(fyh,PREP+PRO,13104)(hdY,NN,13105)(llmtqyn,PREP+NN,
13106)(::,newverse,13107)(Al*yn,RELPRO,13108)(yWmnwn,VV,13109)(bAl
gyb,PREP+NN,13110)(wyqymwn,CONJ+VV,13111)(AlSlAp,NN,13112)(wmmA,CO
NJ+PREP+RELPRO,13113)(rzqnAhm,VV+PRO,13114)(ynfqwn,VV,13115)(::,ne
wverse,13116)(wAl*yn,CONJ+RELPRO,13117)(yWmnwn,VV,13118)(bmA,PREP+
RELPRO,13119)(Onzl,VV,13120)(Ilyk,PREP+PRO,13121)(wmA,CONJ+PART,13
122)(Onzl,VV,13123)(mn,PREP,13124)(qblk,PREP+PRO,13125)(wbAl|xrp,C
ONJ+NN,13126)(hm,PRO,13127)(ywqnwn,VV,13128)(::,newverse,13129)(Ow
l}k,DEMO,13130)(ElY,PREP,13131)(hdY,NN,13132)(mn,PREP,13133)(rbhm,
NN+PRO,13134)(wOwl}k,CONJ+DEMO,13135)(hm,PRO,13136)(AlmflHwn,NN,13
137)
```

Figure 5.17: A portion of the Arabic tagged corpus

The above portion of the corpus shows that we use tuples that include the word, its POS tag and the number of the word's position in the corpus. It also shows that we start every verse with a tuple, `(::,newverse,13099)`, which consists of the double colon, the word indicating a new verse and the number corresponding to its position in the corpus.

The REs that we use to compile different patterns to match against the tagged corpus are illustrated below. We start with shedding light on verbal and nominal patterns then discuss patterns for other POS categories. Verbal and nominal patterns are firstly discussed as follows.

(A) **`verbPattern="(?:[^,]*)VV(?:[^,]*)"`**
(B) **`nounPattern="(?:[^,]*)NN([^,]*)"`**
(C) **`freeVerbPattern="VV"`**
(D) **`freeNounPattern="NN"`**

The previous REs are classified into two main groups. Each group consists of two patterns for verbs and nouns. The first group deals with verbs and nouns that may or may not have proclitics and enclitics attached to them as in (A) and (B), whereas the second group is concerned with free verbs and nouns that have no clitics at all as in (C) and (D). The first group of patterns needs more explanation. Patterns (A) and (B) start with a query (or question mark) followed by a colon. In Python this is called a non-capturing group. It is used when one wants to collect a part of a regular expression, but is not interested in retrieving the group's contents. Then the expression [^,]* is used to state the condition that the pattern in question should include any number of characters which are not commas. Then comes the tag **VV** or **NN** to specify that we match only verbs or nouns. The remainder of the pattern does the same like the beginning part of it. Thus, the verbal pattern (A) above matches any verb preceded by any number of proclitics and/or followed by any number of enclitics. For example, فلنقصن *flnqSn* "then indeed we will definitely narrate" consists of two proclitics, the conjunction ف *f* "then" and the emphatic complementizer ل *l* "indeed", besides the verbal stem نقص *nqS* "(we) narrate" and the emphatic suffix ن *n* (translated here as "definitely") which is called نون التوكيد *nuwn Altawkiyd* in Arabic. In fact, we do not regard prefixes such as the definite article ال *Al* "the" or suffixes such as the previous one as clitics. Clitics are confined only to conjunctions, prepositions and complementizers that are attached to the beginning of a word as well as pronouns that are attached to the end of a word. A verb can be attached to an enclitic pronoun such as جاءك *jA'k* "has come to you", where the verb جاء *jA'* "has come to" is attached to the second person singular pronoun ك *k* "you". Similarly, nouns can have both proclitics and enclitics. For example, وكتبه *wktbh* "and his books" is composed of the conjunction و *w* "and", the nominal plural stem كتب *ktb* "books" and the possessive pronoun ـه *h* "his". The verbal pattern (C), in contrast, matches free verbs that have no clitics. The same applies to the nominal pattern (D) where it matches free nouns that have no clitics.

Second, a number of patterns are compiled for other POS categories, such as determiners, particles, prepositions, pronouns, etc. These patterns are described as follows.

```
(E) numdetdemoPattern="((?:[^,]*)(NUM|DET|DEMO)(?:[^,]*))?"
(F) particlePattern="(?:[^,]*)PART(?:[^,]*)"
(G) auxPattern="(?:[^,]*)AUX(?:[^,]*)"
(H) prepPattern="(?:[^,]*)PREP(?:[^,]*)"
(I) freePrepPattern="PREP"
(J) pronounPattern="PRO"
(K) relativePronounPattern="RELPRO"
```

Since the REs have been explained under the previous patterns for verbs and nouns, we will refer to the categories only in these patterns. The first of these patterns, pattern (E), is concerned with the POS categories **NUM** for ‘number’, **DET** for ‘determiner’ and **DEMO** for ‘demonstrative’. It tries to capture zero or more of such categories. This means that if it finds one or more of these categories, it will match and return them. Otherwise, it will not return any value for them if not found. The other patterns work in the same way for particles, auxiliaries and prepositions as in (F), (G), (H), respectively. There is one thing that should be noted here. We have two patterns for prepositions. The first pattern deals with the cliticised prepositions that can be attached to following nouns or pronouns as mentioned in (H). This pattern captures cases such as بالغيب *bAlgyb* “in the unseen” as well as فيه *fyh* “in it”. As for the second one as in (I), it deals with the free prepositions that occur independently, such as من *mn* “from”. Similarly, the final two patterns (J) and (K) are concerned with free pronouns and relative pronouns respectively.

Due to the lack of a lexicon, as pointed out above, we bracket together heads and their dependents without labelling the grammatical functions, i.e. subject, object, modifier…, etc. in a given construction. From now on we will refer to this bracketing as DRs. The generation of a full parse tree for a sentence is beyond the scope of the current work. We will use a tree diagram to represent DRs with the head being the root node at the top then the dependents on the leaves of the tree. It is also noteworthy that we do not care about any clitics, whether they are proclitics that are attached to the beginning of words or enclitics that are attached to the end of words. Thus, proclitic conjunctions and prepositions as well as enclitic pronouns are not tackled in the parser. We focus only on the open-class items or the full words, i.e. verbs and nouns. This is because we aim for finding syntactically related verbs and nouns in the Arabic corpus and then similarly finding related verbs and nouns in the English corpus so as to map between them and extract a number of seeds to be used for bootstrapping the proposer. Accordingly, the Arabic parser gets the dependency attachment between verbs and nouns whether or not they include clitics.

5.4.2.2 Syntactic Constructions

We now discuss the DRs between heads and dependents in the parser. We classify these relations into a number of classes that cover major syntactic constructions in

Arabic. These classes focus on verbal constructions, copula constructions, nominal constructions, and prepositional phrases. We start with discussing verbal constructions in the following section. We have taken a portion of the corpus to make it a Gold Standard to evaluate the parser’s accuracy. As clarified before, the Arabic POS tagger has an error rate of about 9%. So, we have corrected the mistakes in the POS tags for this chosen portion. We will cite various examples from this Gold Standard as well as other parts of the corpus to illustrate the DRs that we use to parse the POS-tagged corpus.

5.4.2.2.1 Verbal Constructions

The basic sentence structure in Arabic has been discussed earlier in this chapter. It has been pointed out that traditional grammarians classify a sentence as verbal if it starts with a verb and as nominal if it starts with a noun. Nonetheless, in the current framework we cannot distinguish between verbal and nominal sentences, since we are dealing with unpunctuated text as stated earlier. Thus, there are no sentence boundaries that tell us where a sentence ends and another one begins. Consequently, the different constructions we deal with in the parser may constitute a complete sentence or part of a sentence. They are rather chunks, as noted earlier. For these reasons we will describe those constructions or chunks that start with a verb as verbal and those in which the noun comes before the verb or those that have no verb at all as nominal. We begin with illustrating the verbal constructions then proceed to show the other constructions that are covered by the dependency parser.

The basic verbal sentence in Arabic is composed of a verb followed by a subject, object and other complements. The subject can be explicitly stated as an NP or implicitly understood as an elliptic personal pronoun (or a pro-drop). The notion of pro-drop in Arabic has been explained earlier in this chapter. In fact, we do not make any dependency representation for pro-drop cases. We deal only with the explicit items, but elliptic pronouns, which mostly function as subjects, are not represented in our framework. The DRs in a verbal construction are represented through a number of dependency rules in the parser that are described below.

```
1- [('HD', verbPattern), numdetdemoPattern, ('dp',
   nounPattern)]
```

The previous pattern is for the first dependency rule in the parser. The rule says that if a verb occurs in a given construction, it is the head of the whole construction and the remaining elements are dependent on this head. Thus, the current rule begins with the ‘verb’ pattern that may stand alone or be attached to clitics as noted above. The categories we are interested in are enclosed between two brackets. Thus, we are interested only in verb and noun patterns. The ‘noun’ pattern covers both free nouns and nouns with clitics. In this construction the ‘noun’ is dependent on the verb. Notably, **`'HD'`** refers to the head of a particular construction, while **`'dp'`** refers to the dependent. As for the intervening pattern **`'numdetdemoPattern'`**, it refers to the three categories of ‘number’, ‘determiner’ and ‘demonstrative’ that can come between a verb and a following noun. This intervening pattern will recur in some other rules. Here is part of the output for this dependency rule, which also highlights the way the parser outputs DRs for all rules.

```
[[('HD',('yWmnwn','VV','13109')),('dp',('bAlgyb','PREP+NN','13110
'))],[('HD',('wyqymwn','CONJ+VV','13111')),('dp',('AlSlAp','NN','
13112'))],[('HD',('kfrwA','VV','13141')),('dp',("swA'",'NN','1314
2'))],[('HD',('xtm','VV','13151')),('dp',('Allh','NN','13152'))],
[('HD',('|mnA','VV','13168')),('dp',('bAllh','PREP+NN','13169'))]
,[('HD',('yxAdEwn','VV','13176')),('dp',('Allh','NN','13177'))],[
('HD',('fzAdhm','CONJ+VV+PRO','13190')),('dp',('Allh','NN','13191
'))],[('HD',('|mn','VV','13225')),('dp',('AlnAs','NN','13226'))],
[('HD',('|mn','VV','13230')),('dp',("AlsfhA'",'NN','13231'))],[('
HD',('A$trwA','VV','13267')),('dp',('AlDlAlp','NN','13268'))]]
```

Figure 5.18: Part of the output of the Arabic dependency parser for the first dependency rule

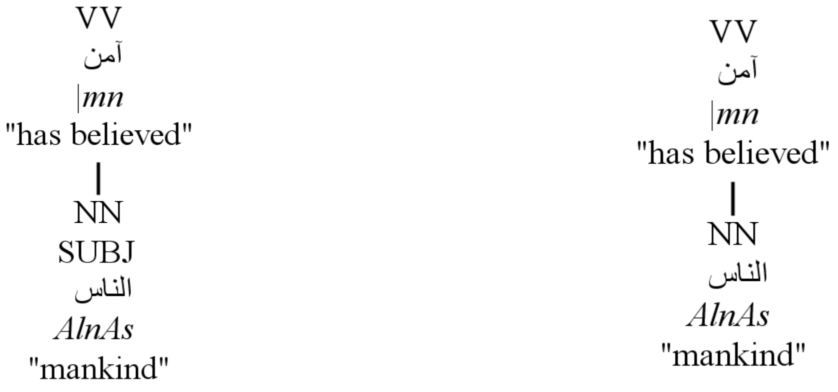
The current pattern covers both cases of transitive and intransitive verbs in the active and passive voice. If the verb is in the active voice, there may be two possibilities for the following explicit noun. First, the explicit noun may be the subject and the verb in this case is intransitive. Second, this explicit noun may be the object and the subject in this case is a pro-drop and thus the verb is transitive. According to Dukes and Buckwalter (2010), this case of a pro-drop subject is more frequent in Qur’anic Arabic. But if the verb is in the passive voice, the explicit noun is then the subject of the passive transitive verb. This subject is called نائب فاعل *naA}ib faAEil* “passive subject” in traditional Arabic grammar. As discussed above, what concerns us here is to get the DRs between the governor and dependent so as to find related words to be used in guiding the proposer. These possible structures for this pattern can be represented through DTs as shown in the following examples.

As shown in the previous figure, the parser outputs DRs between brackets. However, for the purpose of illustration, we will present the DRs in tree diagrams. We will give the ideal DT for a given construction as well as the output of the Arabic parser for such a construction represented also in a tree diagram. In fact, the Arabic input to the parser is written in Buckwalter transliteration and so its output is transliterated Arabic. But we will add the Arabic script as we do throughout the thesis. The following DT shows a DR between an active verb and its related noun which functions as its explicit subject.

5.53 وإذا قيل لهم آمنوا كما آمن الناس

*wI*A qyl lhm |mnwA kmA |mn AlnAs*[13]

[And when it is said to them, "Believe just as mankind has believed,"] (Qur'an, 2:13)



(a) A dependency tree for an active verb with an explicit subject

(b) The parser's output for this construction

Figure 5.19: A dependency tree for an active verb with an explicit subject

As it is clear in diagram (b) of figure (5.19), the parser gets the DR between a given verb and its dependent without labelling the grammatical function of the verb's dependent, which is the subject in the current example[14]. As mentioned earlier, we treat the definite article *Al* "the" as part of the following noun, not as a separate

[13] The constructions on which we focus in a given verse are underlined.

[14] Some of the labels for grammatical functions in the dependency trees are borrowed from the Qur'anic Arabic Corpus (Dukes and Buckwalter, 2010; Dukes et al., 2010), some others are from Columbia Arabic Treebank (CATiB) (Habash et al., 2009), while others are our own.

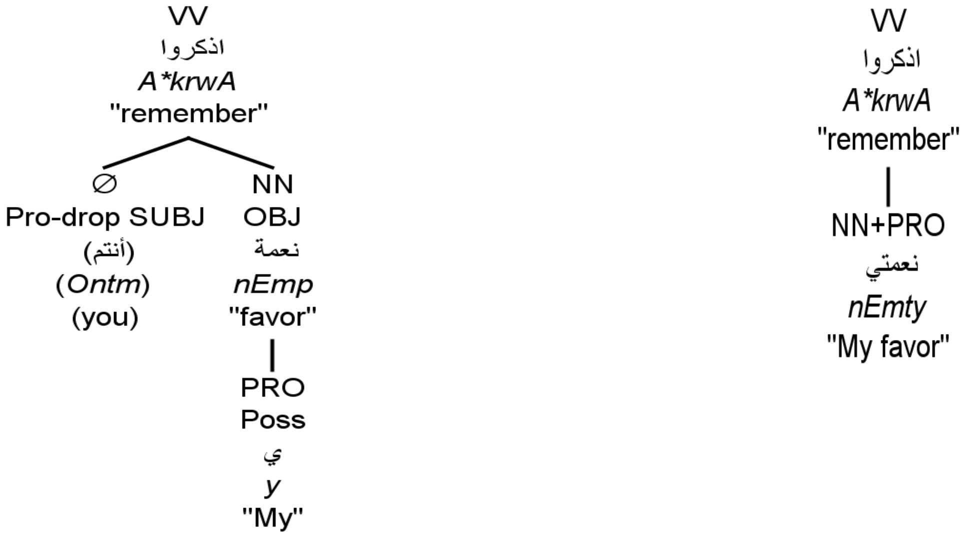
element. Another DT for a verb and its related noun which functions as its object is shown in figure (5.20). It is worth noting that plural agreement markers such as ون *wn*, وا *wA* which are attached to verbs in the plural form are regarded in traditional Arabic grammar as the explicit subject of the whole construction. However, in our ideal analysis we would regard the subject as pro-drop signified by such markers, as shown below. However, the parser's actual output does not handle such pro-drop or zero subjects, as noted earlier. Both the ideal analysis and the parser's actual analysis are given below.

5.54 يا بني إسرائيل اذكروا نعمتي التي أنعمت عليكم

*yA bny IsrA}yl A*krwA nEmty Alty OnEmt Elykm*

[O Seeds (or: sons) of Israel remember My favor wherewith I favored you,]

(Qur'an, 2:40)



(a) A dependency tree for an active verb with a pro-drop subject

(b) The parser's output for this construction

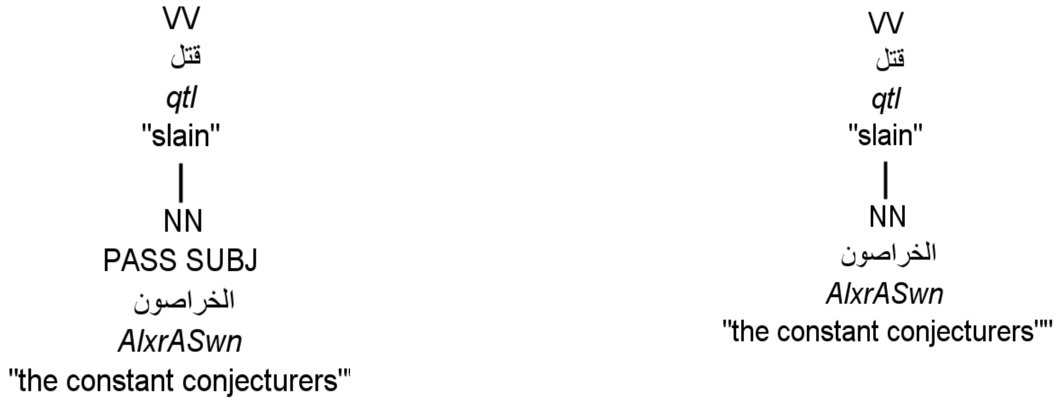
Figure 5.20: A dependency tree for an active verb with a pro-drop subject

Similarly, in this example the parser attaches the noun to the verb in a DR without identifying the noun as the object, as is shown in diagram (b) of the previous figure. In addition, as explained when we discussed the Arabic POS tagger, we do not make word segmentation due to the lack of a lexicon. Thus, a noun with attached clitics is given a complex tag, as the case in this example. The word نعمتي *nEmty* "My favor", which is a possessive construction, is POS tagged as **NN+PRO**. The parser, therefore, treats it as one element and attaches it as a whole with the head verb in a DR, as shown above. The following DT shows a passive verb with its passive subject in a DR.

5.55 قتل الخراصون

qtl AlxrASwn

[Slain are the constant conjecturers,] (Qur'an, 51:10)



(a) A dependency tree for a passive verb with a passive subject

(b) The parser's output for this construction

Figure 5.21: A dependency tree for a passive verb with a passive subject

In this DT the verb قتل *qtl* "slain" is in the passive voice and so the following noun الخراصون *AlxrASwn* "the constant conjecturers" is the passive subject of the verb.

As a matter of fact, the parser sometimes makes the wrong decision and attaches unrelated verbs and nouns in a DR. Such errors mostly occur as a result of attaching a noun dependent on the main verb to the verb of a subordinate clause, which may be an adjectival relative clause, as shown in the following example.

5.56 فبدل الذين ظلموا قولا غير الذي قيل لهم

*fbdl Al*yn ZlmwA qwlA gyr Al*y qyl lhm*

[Then the ones who did injustice exchanged a saying other than that which had been said to them] (Qur'an, 2:59).

The dependency parser attaches the noun قولا *qwlA* "a saying" to the verb ظلموا *ZlmwA* "did injustice" in a DR, which is totally wrong. This has been done because the noun *qwlA* is immediately preceded by the verb *ZlmwA* and thus the parser attaches both of them in a DR. In fact, the noun *qwlA* is related to the main verb بدل *bdl* “exchanged” in a verb-object relation, not to the verb *ZlmwA* which functions as the صلة *Slp* “subordinate clause” of the relative pronoun الذين *Al*yn* “who”. The shallow parser could not get the relation between the verb *bdl* and its object *qwlA*, since it cannot handle long-distance dependencies, but deals only with adjacent head and dependents.

```
2- [('HD', verbPattern), numdetdemoPattern, ('dp1',
   nounPattern), numdetdemoPattern, ('dp2', nounPattern)]
```

This pattern is for the second dependency rule in the parser. In this second rule the head verb is followed by two dependent nouns. In fact, there is overlapping between the rules in the parser, so that the relations in this pattern are definitely included in the previous pattern. This overlapping could be avoided by cascading the rules so as to apply them in order. This way of ranking the dependency rules is not carried out in the current framework but can be a topic for future work. Nevertheless, the current framework is sufficient for the main goal of the parser, which is to find related words to improve the proposer, as noted before.

In our implementation of this pattern to extract the ‘head-dependent’ pairs from the parallel corpus, we deal only with two elements, i.e. the head verb and one following dependent. Thus, we subdivide this pattern into two patterns as follows:

```
[('HD', verbPattern), numdetdemoPattern, ('dp',
nounPattern), numdetdemoPattern, nounPattern]
[('HD', verbPattern), numdetdemoPattern, nounPattern,
numdetdemoPattern, ('dp', nounPattern)]
```

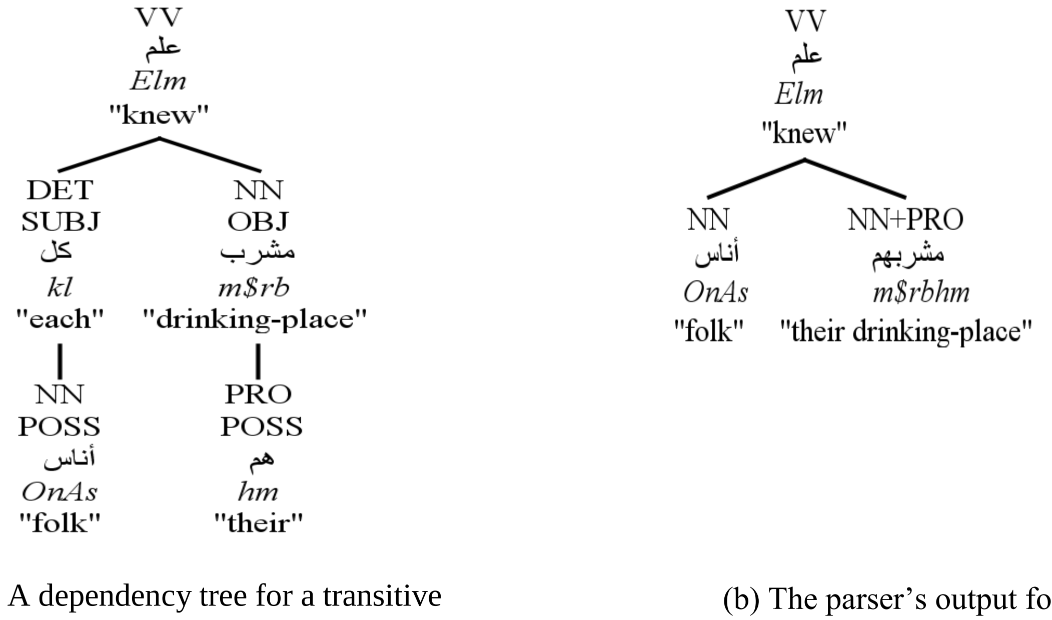
In the first pattern above we focus on the head verb and only the first noun that follows a given verb, whereas in the second pattern we focus on the second noun and leave out the first. However, in our discussion of the DRs between words in different constructions that are covered by this rule we will describe the relations between the three words: the verb and the two following nouns.

Basically when two explicit nouns follow a verb, these two arguments can have different interpretations. First, they may be both the subject and object. Sometimes the first noun may be in a possessive relation to a previous determiner which is the actual subject as shown below. Second, the subject may be a pro-drop which is more common in the Qur'an, as noted above, and the first noun is the object and the second noun is a cognate accusative, which is called مفعول مطلق *mafoEuwl muTolaq* "absolute object". The verb in these two cases is transitive. Third, they may be the indirect and direct objects with a pro-drop subject and the verb is ditransitive in this case. A fourth possibility is that the first noun may be an object and the second noun is a modifier for the previous noun. This modifier may be functioning as an adjectival modification or it may be the satellite noun of a construct NP. This last possibility applies with intransitive verbs also, where the first noun may be an explicit subject and the second noun is an adjectival modifier or part of a construct NP. Otherwise, it may be that the first noun is an explicit subject as the case before and the second noun is an adverbial phrase or "adjunct" called حال *HaAl* in traditional Arabic grammar. We will show DTs for some of these possible structures that comprise three elements: the verb and the two following nouns.

5.57 قد علم كل أناس مشربهم

qd Elm kl OnAs m$rbhm

[Each folk already knew their drinking-place.] (Qur'an, 2:60)



(a) A dependency tree for a transitive verb with an explicit subject and object

(b) The parser's output for this construction

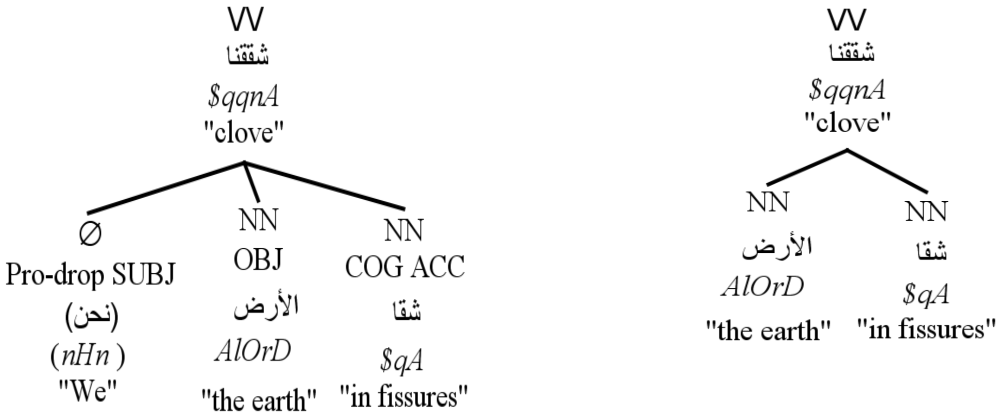
Figure 5.22: A dependency tree for a transitive verb with an explicit subject and object

From the parser’s output for the above DT we note that it obtains the DR between the verb and the two following nouns and ignores the intervening determiner, that is the quantifier كل *kl* “each”, which functions as the subject in this construction. The determiner is annexed to the noun أناس *AnAs* “folk”, thus forming a construct phrase.

5.58 ثم شققنا الأرض شقا

vm $qqnA AlOrD $qA

[Thereafter We clove the earth in fissures,] (Qur’an, 80:26)



(a) A dependency tree for a transitive verb with a direct object and a cognate accusative

(b) The parser’s output for this construction

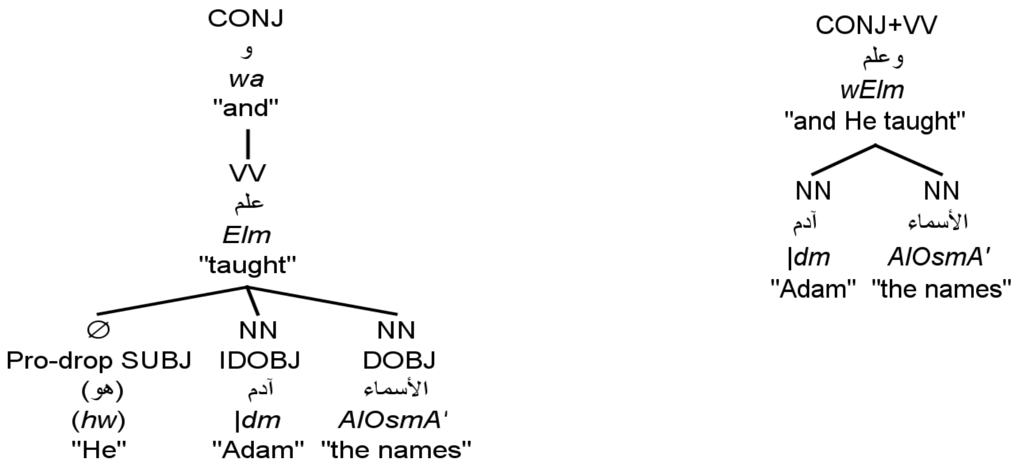
Figure 5.23: A dependency tree for a transitive verb with a direct object and a cognate accusative

The first noun in this DT is the direct object and the second noun is a cognate accusative that emphasizes the verb. The subject here is a prodrop indicated by the inflectional agreement suffix and is estimated as “we”.

5.59 وعلم آدم الأسماء كلها

wElm |dm AlOsmA' klhA

[And He taught Adam all the names;] (Qur’an, 2:31)



(a) A dependency tree for a ditransitive verb

(b) The parser's output for this construction

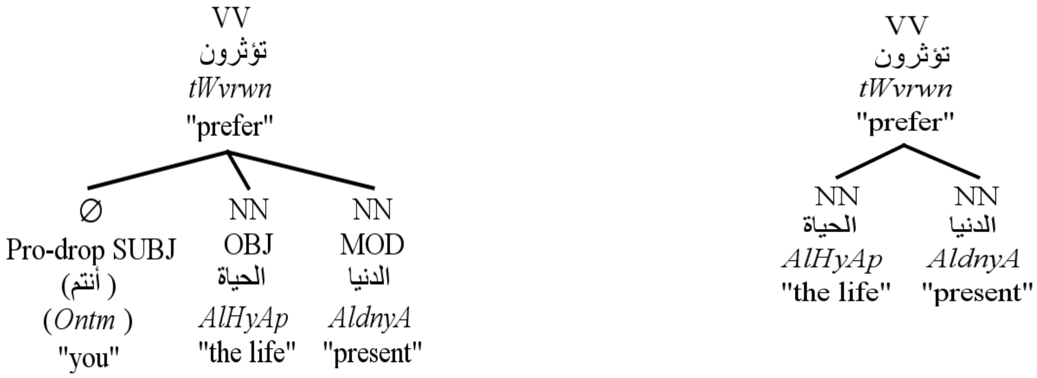
Figure 5.24: A dependency tree for a ditransitive verb

The first noun in the current DT is the indirect object and the second noun is the direct object. The subject here is a pro-drop estimated as "he".

5.60 بل تؤثرون الحياة الدنيا

bl tWvrwn AlHyAp AldnyA

[No indeed, you prefer the present life,] (Qur'an, 87:16)



(a) A dependency tree for a transitive verb with a direct object and a nominal modifier

(b) The parser's output for this construction

Figure 5.25: A dependency tree for a transitive verb with a direct object and a nominal modifier

In the above DT the head verb is followed by the object and a nominal that modifies that object. As for the subject, it is pro-drop in this example.

In the previous patterns we discussed the DR between a head verb and either one or two following dependent nouns. In the following lines we will discuss one more pattern for verbal constructions that we use to parse the Arabic tagged corpus with DRs.

3- `[('HD', verbPattern), ('dp1', freePrepPattern), ('dp2', nounPattern)]`

In the above pattern we obtain the DR between a head verb and a following prepositional phrase (PP). Nevertheless, in our implementation we focus only on the head verb and the following dependent noun as shown in the following pattern:

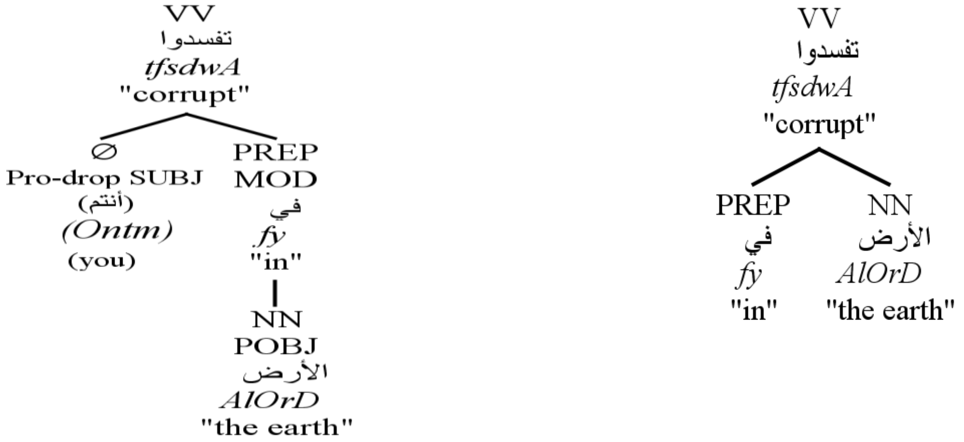
```
[('HD', verbPattern), freePrepPattern, ('dp', nounPattern)]
```

As pointed out above, we will describe the DRs between the three elements as shown in the following example.

5.61 وإذا قيل لهم لا تفسدوا في الأرض قالوا إنما نحن مصلحون

*wI*A qyl lhm lA tfsdwA fy AlOrD qAlwA InmA nHn mSlHwn*

[And when it is said to them, "Do not corrupt in the earth," they say, "Surely we are only doers of righteousness." (i.e. reformers, peacemakers)] (Qur'an, 2:11)



(a) A dependency tree for a verb followed by a prepositional phrase

(b) The parser's output for this construction

Figure 5.26: A dependency tree for a verb followed by a prepositional phrase

As shown in the above output of the parser the prepositional phrase as a whole, i.e. the preposition and the following noun which is the object of preposition (POBJ), is in a DR to the head verb. But when we parse the POS-tagged corpus we focus on the head verb and the following noun and leave out the preposition, as noted above. Although the output of the parser, which is given in DT (5.26b), does not agree with the ideal representation in (5.26a), it is considered a correct alternative analysis by some dependency grammarians (Nivre, 2006), where both the preposition and the noun are dependents of the verb.

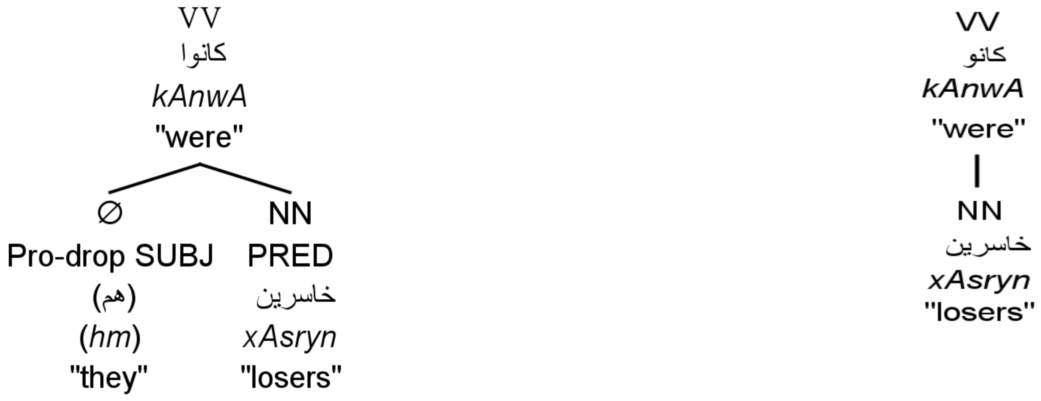
5.4.2.2.2 Copula Constructions

At the beginning of this chapter we pointed out that there is a type of nominal sentence in Arabic that consists of a subject and predicate in the nominative case. This type of sentence is called equational. Normally these sentences are referred to as zero copula, since the verb كان *kaAna* "to be" is not used in the present tense indicative, but simply understood. Nevertheless, when such sentences are used in the past or future tense, the verb كان *kaAna* is explicitly used. It is also explicitly used when the sentence is negated in the present. In this case these verbs can both serve as auxiliary or copulative. The Arabic tagset that we use to tag the Arabic text does not include a separate tag for the verbs that serve as a copula. We give them the same tag as main verbs, i.e. **VV**. However, we use the tag **AUX** when such verbs are used as auxiliary verbs. The patterns for such copulative verbs are the same like the previous patterns that were discussed under the verbal constructions above. This means that copulative verbs could be followed by two explicit nouns, with the first noun functioning as the subject and the second noun as the predicate. The subject of a copula can be sometimes a pro-drop and in this case one explicit noun, i.e. the predicate, follows the copulative verb. We will give one example below that shows a DT for a construction that starts with a copulative verb.

5.62 أولئك الذين حق عليهم القول في أمم قد خلت من قبلهم من الجن والإنس إنهم كانوا خاسرين

*Owl}k Al*yn Hq Elyhm Alqwl fy Omm qd xlt mn qblhm mn Aljn wAlIns Inhm kAnwA xAsryn*

[Those are they against whom the Saying has come true among nations that already passed away even before them, of the jinn and humankind (alike); surely they were losers.] (Qur’an, 46:18)



(a) A dependency tree for a copula verb with a pro-drop subject and predicate

(b) The parser’s output for this construction

Figure 5.27: A dependency tree for a copula verb with pro-drop subject and predicate

This DT describes the DR between a copula verb and its pro-drop subject and predicate. The parser obtains the DR between the head verb and the explicit noun which is the predicate, but it does not deal with pro-drop or zero items, as noted before. As pointed out above, we do not use a separate tag for copulative verbs; we use the **VV** tag for all types of verbs except auxiliary verbs for which we use **AUX** tag. Sometimes the pro-drop subject can be explicitly mentioned to emphasize the subject. This occurs with verbs in general, whether they are copula or not.

5.4.2.2.3 Nominal Constructions

It has been repeatedly mentioned throughout the thesis that the Arabic text that we are using is unpunctuated. This makes it extremely difficult to obtain reasonably accurate DRs for nominal constructions, i.e. a construction that starts with a noun. This is because in the absence of punctuated sentences a noun in a given construction can belong either to the following verb, which we aim for here, or to a preceding verb which has no relation with such a noun. Going through the Gold Standard, which is only 1100 words, it has been found out that using REs to obtain DRs between a given noun and a following verb results in the wrong dependency attachment in most cases. This is expected to result in a bigger number of wrong DRs in the entire corpus. The following example shows a wrong DR outputted by the parser for the following nominal construction pattern.

```
1- [('dp', nounPattern), ('HD', verbPattern)]
```

5.63 وإذا قيل لهم آمنوا كما آمن الناس قالوا أنؤمن كما آمن السفهاء ألا إنهم هم السفهاء ولكن لا يعلمون

*wI*A qyl lhm |mnwA kmA |mn AlnAs qAlwA OnWmn kmA |mn AlsfhA' OlA Inhm hm AlsfhA' wlkn lA yElmwn*

[And when it is said to them, "Believe just as mankind has believed, " they say, "Shall we believe just as the fools have believed?" Verily, they, (only) they, are surely the fools, but they do not know.] (Qur'an, 2:13)

The above-mentioned verse contains the noun الناس *AlnAs* "mankind" preceded by the verb آمن *|mn* "has believed" and followed by the verb قالوا *qAlwA* "they say". When we execute the above pattern to obtain the DR between nouns and following verbs, it makes a wrong dependency attachment between the current noun *AlnAs*, which is dependent on the preceding head verb *|mn*, and the following verb *qAlwA*. This wrong attachment occurs frequently when this nominal pattern is implemented. This occurs also with cliticized nouns as the following example shows.

5.64 قد علم كل أناس مشربهم كلوا واشربوا من رزق الله ولا تعثوا في الأرض مفسدين

qd Elm kl OnAs m$rbhm klwA wA$rbwA mn rzq Allh wlA tEvwA fy AlOrD mfsdyn

[Each folk already knew their drinking-place. "Eat and drink of the provision of Allah, and do not perpetrate (mischief) in the earth, (as) corruptors."] (Qur'an, 2:60)

The above-mentioned verse contains the cliticized noun مشربهم *m$rbhm* "their drinking-place" followed by the verb كلوا *klwA* "eat". When the above pattern is carried out to get the DR between nouns and following verbs, it obtains a wrong dependency attachment between the current noun *m$rbhm*, which is actually dependent on the preceding verb علم *Elm* "knew", and the following verb كلوا *klwA*.

To reduce the bad impact of this pattern, we have introduced the condition that the noun in question should be preceded by a determiner, demonstrative or numeral. The pattern is thus modified to be as follows:

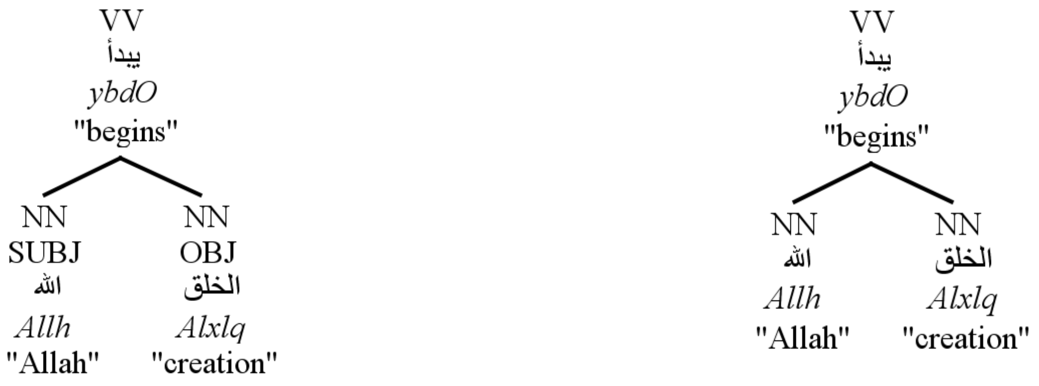
2- **`[numdetdemoPattern, ('dp', nounPattern), ('HD', verbPattern)]`**

In this pattern the first constituent **`numdetdemoPattern`** refers to the three POS categories **NUM**, **DET**, and **DEMO**. This condition is a way of constraining the noun that starts the pattern so that it might be linked to the following verb. The constrained pattern still has DR mistakes, but when we examined portions of the corpus, including the Gold Standard, we found that the number of wrong DRs has decreased, though the number of outputted DRs is now much fewer. This means that the precision increased without greatly affecting the recall. However, as regards the task of finding 'head-dependent' pairs in the parallel corpus to improve the proposer, and for which this shallow parser has been developed, we found out that applying this constrained pattern to obtain DRs between words in the parallel corpus results in very few translational pairs that are then filtered to produce ultimately only one pair. Therefore, using this pattern for the task of finding 'seeds' is not useful for the unpunctuated text that we are handling. So, we will not evaluate this pattern as the other used patterns. The following example shows a DT for a nominal construction.

5.65 الله يبدأ الخلق ثم يعيده ثم إليه ترجعون

Allh ybdO Alxlq vm yEydh vm Ilyh trjEwn

[Allah begins creation; thereafter He brings it back again; thereafter to Him you will be returned..] (Qur'an, 30:11)



(a) A dependency tree for a nominal construction

(b) The parser's output for this construction

Figure 5.28: A dependency tree for a nominal construction

In traditional Arabic grammar there is a pro-drop subject estimated as هو *hw* “he” after the verb يبدأ *ybdO* “begins”, which refers back to the explicit subject الله *Allh* “Allah”. Also, traditional Arabic grammarians consider the verb and the following object as the predicate of the explicit subject in this construction. It should be noted that there is overlapping between different rules. So, the noun immediately following the verb is in a DR to the verb in the current rule and it is also so in the first rule that is discussed under the verbal constructions above. Undoubtedly, better dependency parsing could have been achieved by ranking the rules as mentioned before.

At the beginning of this chapter we have discussed the different types of nominal sentences. We have touched upon a specific type of nominal sentence, namely equational sentences. These equational constructions are verbless because the copula verb كان *kaAna* “to be” is not used in the present tense indicative. Thus, such sentences are called zero copula. The copula, however, is used when a sentence is in the past or future tense or if the sentence is negated in the present. Different examples for zero copula constructions have been given in that part of the thesis. As far as the current framework of dependency parsing is concerned, it is extremely hard to obtain the DRs between subjects and predicates in the zero copula constructions using this specific type of unpunctuated text and under the lack of a lexicon. This is because most often the subject of a zero copula construction is attached to a previous unrelated noun and in this way it is treated as the predicate and the previous noun as the subject. This mostly happens with the current framework because the parser has not got enough information to detect where a given construction ends and the other begins because of the lack of punctuation in the text under analysis. Second, we do not have fine-grained morphological cues about words, which can handle case markers that distinguish between nominative and accusative case. Third, we do not have a separate POS tag for the definite article that could have been used as a cue to distinguish between construct phrases in which the first noun is always indefinite and modified nominal phrases in which the first noun may be definite or indefinite.

Thus, if we try to get DRs for two nouns that follow each other, these two nouns can be subject and predicate of a zero copula construction, subject and object of a verbal construction (or could be indirect and direct objects) or a head noun followed by a nominal modifier which may be an adjectival modifier or a satellite noun in a construct phrase. Accordingly, DRs are not compatible in these various constructions. So, the first noun, i.e. the subject, in a zero copula construction is dependent on the second noun, i.e. the predicate. However, the modifier following a head noun or the satellite noun in a construct phrase depends on the first noun. Even more, the noun which is the object in a verbal construction does not depend on the preceding noun, i.e. the subject, but both the object and subject depend on the verb in the construction in question. We will shed more light on this point by giving examples for each of these syntactic construction types below.

We start by giving an example for a 'construct phrase' then discuss other types of two following nouns that are problematic under the current framework.

5.66 ثم توليتم من بعد ذلك فلولا فضل الله عليكم ورحمته لكنتم من الخاسرين

*vm twlytm mn bEd *lk flwlA fDl Allh Elykm wrHmth lkntm mn AlxAsryn*

[Thereafter you turned away even after that, so had it not been for the Grace of Allah towards you and His mercy, indeed you would have been of the losers.] (Qur'an, 2:64)

The underlined words فضل الله *fDl Allh* "the Grace of Allah" are an example of a construct phrase where the first noun is the head noun which has neither the definite article nor nunation because it is annexed to the second noun. But it can take any case mark: nominative, accusative or genitive depending on the function of the whole construct phrase in a sentence structure. As for the second noun, which is called the satellite, it is marked either for definiteness or indefiniteness, and is always in the genitive case.

5.67 قالوا ادع لنا ربك يبين لنا ما لونها قال إنه يقول إنها بقرة صفراء فاقع لونها تسر الناظرين

qAlwA AdE lnA rbk ybyn lnA mA lwnhA qAl Inh yqwl InhA bqrp SfrA' fAqE lwnhA tsr AlnAZryn

[They said, "Invoke your Lord for us that He make evident to us what color she is." He said, "Surely He says that surely she is a yellow cow, bright (is) her color, pleasing to the onlookers".] (Qur'an, 2:69)

The underlined words above are a 'modified NP', where the second noun functions as an adjectival modifier. Both nouns in this NP are in the indefinite case. They can be also used in the definite case. However, if the first noun is indefinite and the second noun is definite, it will be a construct phrase as shown above.

5.68 الله لطيف بعباده يرزق من يشاء وهو القوي العزيز

Allh lTyf bEbAdh yrzq mn y$A' whw Alqwy AlEzyz

[Allah is Ever-Kind to His bondmen; He provides whomever He decides; and He is The Ever-Powerful, The Ever-Mighty.] (Qur'an, 42:19)

Here the underlined words are a 'zero copula' construction. The first noun is called مبتدأ *mubtadaO* "that starts the construction", i.e. the subject, and the second noun is called خبر *xabar* "predicate". Both nouns are always in the nominative case. Notably, the predicate in this example has the prepositional phrase بعباده *bEbAdh* "to his bondmen" as a complement. As can be observed, the copula appears in the English translation.

5.69 وعلم آدم الأسماء كلها

wElm |dm AlOsmA' klhA

[And He taught Adam all the names;] (Qur'an, 2:31)

In this example the two underlined nouns function as the indirect and direct objects respectively. The subject is pro-drop estimated as هو *hw* "he".

We have seen that two successive nouns can have different interpretations under the lack of fine-grained morphological information: lack of case marking, lack of separate tags for definite articles in addition to the lack of a subcategorization lexicon. Therefore, in the current framework we could not have a rule for DRs between two successive nouns because such DRs are not consistent, and are problematic for the current unpunctuated text, as pointed out above.

5.4.2.2.4 Prepositional Phrases

The final pattern that we use in the Arabic dependency parser is that for prepositional phrases (PPs). It is worth noting that we have written a large number of patterns to cover a good deal of DRs between words in given constructions, but on examination it has been found that many of them produce wrong DRs owing to the constraints under which we are conducting the current parser. Thus, we have chosen only those patterns that we trust. It has been made clear throughout the thesis that we are mainly interested in open-class words, i.e. verbs, nouns, adjectives and adverbs, in the current task of lexical selection. We do not attempt to handle closed-class words. However, as far as the parser is concerned, we have used a pattern that gets the DR between prepositions and following nouns so as to match it with the corresponding PPs in the English corpus to obtain translational pairs that can be used as anchor points for bootstrapping the selection process. We do not aim at selecting equivalents for prepositions; we only use them in the collection of ‘head-dependent’ pairs. We will give the pattern that we use for PPs and an example from the Gold Standard that shows this DR.

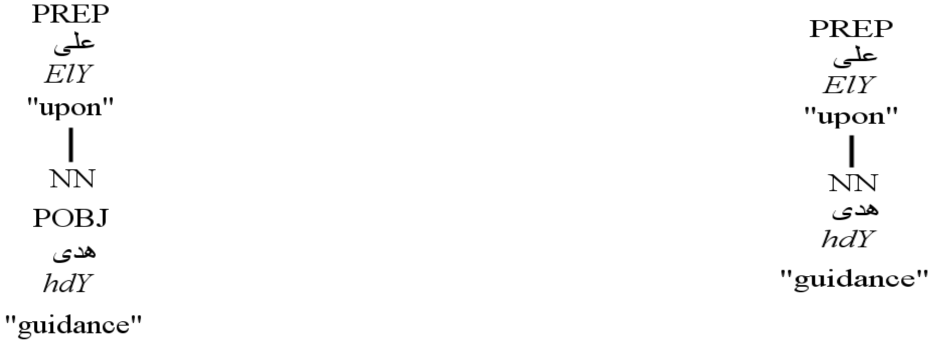
```
[('HD', freePrepPattern), ('dp', nounPattern)]
```

We hold the view that the preposition is the head in a PP. This pattern is used to collect DRs between head prepositions and dependent nouns. This pattern is composed of two sub-patterns for both prepositions and nouns. The preposition pattern **`freePrepPattern`** focuses only on those free prepositions that do not have any clitics. This has been done to exclude those prepositions that have cliticized items such as particles and pronouns. As for the noun pattern **`nounPattern`**, we get both free nouns and cliticized nouns. This is because nouns may have cliticized genitive pronouns in the final position. The following example shows two PPs, where the first PP has a free noun and the second PP contains a cliticized noun.

5.70 أولئك على هدى من ربهم وأولئك هم المفلحون

Owl}k ElY hdY mn rbhm wOwl}k hm AmflHwn

[Those are upon guidance from their lord, and those are they who are the prosperers.] (Qur’an, 2:5)

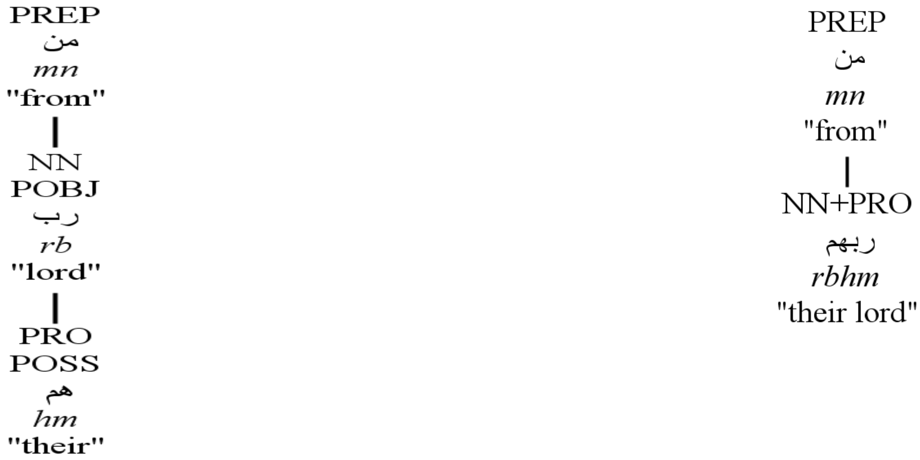


(a) A dependency tree for a PP including a free noun

(b) The parser’s output for this construction

Figure 5.29: A dependency tree for a prepositional phrase including a free noun

In the above DT the noun which is the object of preposition (POBJ) has no clitics. The following DT shows a PP that contains a cliticized noun.



(a) A dependency tree for a PP including a cliticized noun

(b) The parser's output for this construction

Figure 5.30: A dependency tree for a PP including a cliticized noun

5.4.3 Evaluation

We now evaluate the accuracy of the Arabic shallow parser for the dependency rules that we implemented to obtain the 'head-dependent' pairs. We are interested in the preciseness of the used rules, since we ultimately seek to obtain a number of trusted seeds. So, what matters is to get the seeds right, even if they are few in number. Therefore, the recall issue is not of concern to us here. The following table shows the precision score for each one of the used rules when they are applied to the Gold Standard.

Rules	**Head**	**Dependent**	**Accuracy**
Verb>Noun	Verb	Noun	0.954
Verb>Noun>(Noun)	Verb	First Noun	0.971
Verb>(Noun)>Noun	Verb	Second Noun	0.857
Verb>(Prep)>Noun	Verb	Noun	1.0
Prep>Noun	Prep	Noun	1.0

Table 5.3: Accuracy Scores for dependency rules

The sign > is used in the table to mean that the current POS category, which is normally the head in these rules, is followed by other categories which are often its dependents. Also, the brackets ( ) are used to mean that what is between them is not matched in a given rule. So, we apply the condition that there should be two nouns following the verb in the second and third rules, but we match only the first and leave out the second in the second rule and escape the first and match the second in the third rule. The third rule in the previous table is worth consideration. It has scored lower than the second rule. This is because in a number of cases the second noun that follows a verb is not a dependent of that verb but of some other item. We will show an example of a wrong DR made by this third rule and explain the reason for that.

5.71 قال إنه يقول إنها بقرة لا ذلول تثير الأرض ولا تسقي الحرث مسلمة لا شية فيها

*qAl Inh yqwl InhA bqrp lA *lwl tvyr AlOrD wlA tsqy AlHrv mslmp lA $yp fyhA*

[He said, "Surely He says that surely she is a cow not tractable (Literally made subservient) to stir the earth or to water the tillage, with no blemish in it.]

(Qur'an, 2:71).

The parser has introduced a wrong DR between the verb تسقي *tsqy* “water” and the second following noun مسلمة *mslmp* “sound”. This is because this second noun functions as an adjective for the noun بقرة *bqrp* “cow” mentioned earlier in the verse. Thus, it is not dependent on the current verb. The average score for the used five rules is 0.956.

5.5 English Lexicon-Free Dependency Parser

Having discussed the different dependency rules that we use in the Arabic parser to obtain DRs between lexical items in the Arabic corpus, we now set out to discuss the English parser and the dependency rules that we use to obtain DRs between lexical items in the English corpus. It should be emphasized that the English POS tagger and dependency parser are not part of the contributions of this thesis. We are just using them as a black box in support of our main task. So, we are not going to evaluate their performance. Nonetheless, looking at a random part of the English POS-tagged corpus shows that despite the errors made by the English tagger, using it is useful in tagging the English text of the parallel corpus to achieve the main task of lexical selection. As regards the English dependency parser that depends on the output of the tagger, it is also useful in labelling DRs between open-class words in the corpus. As noted above, we aim ultimately to map between both Arabic and English DRs to extract ‘head-dependent’ translational pairs to be used in our attempt to improve the proposer.

As was the case with Arabic, the English dependency parser is lexicon-free. A number of patterns have been used to obtain the DRs between head words and their dependents in the English translation in the parallel corpus. We focus on the major DRs in English, as we did with Arabic. REs are used also here to compile patterns. However, in the English parser we use REs to compile patterns for the dependency rules; we do not use it to make patterns for nouns, verbs or any other category, as was the case with Arabic. This is because English lexical items are not cliticized like Arabic, with the exception of the possessive form and some abbreviated forms such as *‘ll* for “will” and *’ve* for “have”. Anyway, we do not take interest in the possessive marker, i.e. the apostrophe and also these abbreviated forms do not occur in the English corpus that we use. Thus, we do not need to distinguish patterns for free words and cliticized words as we did in Arabic. We use a given POS tag immediately inside the patterns for the dependency rules, as will be shown below. These rules cover a number of syntactic constructions. So, we will discuss each construction with illustrative examples. First, we will show a portion from the English tagged corpus against which we match the used patterns.

```
(::,PU,26496)(that,CJ,26497)(is,VB,26498)(the,AT,26499)(book,NN,26
500)(there,EX,26501)(is,VB,26502)(no,AT,26503)(suspicion,NN,26504)
(about,PR,26505)(it,PN,26506)(a,AT,26507)(guidance,NN,26508)(to,PR
,26509)(the,AT,26510)(pious,NN,26511)(::,PU,26512)(who,PN,26513)(b
elieve,VV,26514)(in,PR,26515)(the,AT,26516)(unseen,AJ,26517)(and,C
J,26518)(keep,VV,26519)(up,AV,26520)(the,AT,26521)(prayer,NN,26522
)(and,CJ,26523)(expend,NN,26524)(of,PR,26525)(what,DT,26526)(we,PN
,26527)(have,VH,26528)(provided,VV,26529)(them,PN,26530)(::,PU,265
31)(and,CJ,26532)(who,PN,26533)(believe,VV,26534)(in,PR,26535)(wha
t,DT,26536)(has,VH,26537)(been,VB,26538)(sent,NN,26539)(down,AV,26
540)(to,TO,26541)(you,PN,26542)(and,CJ,26543)(what,DT,26544)(has,V
H,26545)(been,VB,26546)(sent,NN,26547)(down,AV,26548)(before,PR,26
549)(you,PN,26550)(and,CJ,26551)(they,PN,26552)(constantly,AV,2655
3)(have,VH,26554)(certitude,NN,26555)(in,PR,26556)(the,AT,26557)(h
ereafter,NN,26558)(::,PU,26559)(those,DT,26560)(are,VB,26561)(upon
,PR,26562)(guidance,NN,26563)(from,PR,26564)(their,DP,26565)(lord,
NN,26566)(and,CJ,26567)(those,DT,26568)(are,VB,26569)(they,PN,2657
0)(who,PN,26571)(are,VB,26572)(the,AT,26573)(prosperers,NN,26574)
```

Figure 5.31: A portion of the English tagged corpus (with incorrect tags underlined)

The above portion of the corpus is the translation of the Arabic corpus portion in figure 5.17 above. The English tagset, as pointed out in the previous chapter, is based on the BNC basic (C5) tagset with some modifications. We have illustrated earlier what the tuples in the corpus mean. In this portion of the corpus some words are wrongly tagged, e.g. the verb *expend* is wrongly tagged as **NN**. These mistakes that are introduced by the English tagger actually affect the accuracy score for the current task of dependency parsing as well as the ultimate task of lexical selection. Definitely, a better score could have been achieved for both tasks if the English corpus had had fewer wrong tags. It should be noted that the above POS-tagged portion of the corpus includes 8 wrong tags out of 75. This portion could be used for evaluating the English tagger's accuracy, which then stands at 89 %. Nevertheless, it should be affirmed that 4 out of the 8 wrong tags are introduced because the BNC itself generally gets these words wrong. We used this English tagger despite its known problems because it is lexicon-free and we aim for carrying out the entire project without any hand-coded information.

5.5.1 English Dependency Relations

Now we will discuss the DRs that we use in the English parser. We pointed out above that in the implementation stage we focus only on the head and one dependent element. This is for both Arabic and English. Thus, we will discuss here the implemented rules that handle two elements only, i.e. the head and dependent. We should recall that it has been mentioned at the early part of this chapter that the DRs for Arabic are not labelled with grammatical functions such as subject, object, etc. But for English we label the DRs with grammatical functions such as subject, object, etc. We will classify these DRs into a number of categories as follows.

1- 'subject-verb' relation

2- ‘verb-object’ relation

3- ‘verb-PP’ relation

4- ‘prep-noun’ relation

It has been pointed out before that English verbs differ with regard to their valency, i.e. the number of arguments that a verb can have. Thus, verbs may be ‘zero-valent’, in which case they take no argument such as *rain*. Other verbs are ‘mono-valent’, where they subcategorize for only one argument, namely their subject (e.g. *laugh*). A third category of verbs, i.e. ‘di-valent’, can have two arguments, that is the subject and object (e.g. *see*). Finally, tri-valent verbs take three arguments which are the subject and both indirect and direct objects such as *give* (Allerton, 1982). The DRs that we exploit in the English parser are bound to cover these valency-bound verbs and most of their arguments. We will discuss each of these DRs and the rules that are used to obtain them in the following lines with illustrative examples.

5.5.1.1 ‘Subject-Verb’ Relation

This is the first English DR that is used in the English parser. This DR is between head verbs and their subjects. The rule for this DR is written as follows:

```
['(DT|DP|AT)?','(AJ|NN)*',('subj','(NN|NP)'),'V(B|D|H|M)*',
'XX*',('HD','VV')]
```

The pattern for this rule comprises a number of components. These components are either obligatory or optional. The first two components, i.e. `'(DT|DP|AT)'`, `'(AJ|NN)*'` are optional. The first component refers to the three POS tags: **DT** for ‘general determiner’ which typically occurs either as the first word in an NP or as the head of an NP such as *this* in both sentences *this is my book* and *this book is mine,* **DP** for ‘possessive determiners’ (e.g. *your*, *his*, *their*) and **AT** for ‘articles’ (e.g. *the, a*, *an*). The ‘|’ operator between these categories means match either **DT**, **DP** or **AT** at the beginning of the whole pattern, if any of them is found. In other words, it means match zero or only one of such categories. That is why the query ? is added at the end of it. Then match the second component, i.e. `'(AJ|NN)*'` which is composed of the two POS tags **AJ** for any type of ‘adjectives’ and **NN** for any type of ‘nouns’. The star * is used in this pattern to mean match zero or more of these tags. This is because there may be a number of modifying adjectives or nouns that precede head nouns in a given NP. The first two optional components are followed by the first obligatory component that should be matched in this pattern. This obligatory component contains two sub-components; the first one `'subj'` for the grammatical label ‘subject’, and the second one `'(NN|NP)'` for the POS tags **NN** or **NP**. It means that in case any ‘general noun’ or ‘proper noun’ immediately precedes a verb in a given sentence, it is the subject of this verb and thus depends on it. As for the second obligatory component, i.e. `('HD', 'VV'),` it refers to head verbs. It is preceded by two optional sub-patterns. The first one is `'V(B|D|M|H)*'` which is used to match auxiliary and modal verbs that may come before a main verb. It ends with the star to mean that zero or more of auxiliary or modal verbs can precede main verbs. Thus, VB refers to the verb ‘to be’, whether in the present tense (e.g. *am*, *is*, *are*) or in the past tense (e.g. *was*, *were*). The other tags are used for other types of auxiliary or modal verbs, such as the different forms of the verb *do*, *have* and *will*. The details of the used English tagset have been described in the previous chapter. The other sub-pattern that precedes head verbs is `'XX*'` which stands for the negative particle “not” or “n’t”. In a nutshell, this first rule says that a noun that precedes a main verb in a given construction is the subject of this verb on which it depends. The following figure shows a portion of the output of this rule. It should be noted that the parser outputs the head then the dependent. Thus, verbs will come before their subjects, as shown below.

```
[('HD',('believed','VV','26755')),('subj',('mankind','NN','26753'
))],[[('HD',('believed','VV','26766')),('subj',('fools','NN','267
64'))],[('HD',('gained','VV','26846')),('subj',('commerce','NN','
26845'))],[('HD',('said','VV','27382')),('subj',('lord','NN','273
81'))],[('HD',('knowing','VV','27489')),('subj',('ever','NN','274
88'))],[('HD',('recompense','VV','27928')),('subj',('self','NN','
27926'))],[('HD',('took','VV','28150')),('subj',('thunderbolt','N
N','28149'))],[('HD',('indeed','VV','28572')),('subj',('mercy','N
N','28571'))],[('HD',('pleasing','VV','28736')),('subj',('color',
'NN','28735'))]]
```

Figure 5.32: Part of the output of the English dependency parser for the first dependency rule

In the above portion of the corpus there are wrong DRs due to the wrong POS tags that are assigned to words in the corpus. For example, the adverbs “ever” and “indeed” are wrongly tagged as **NN** and **VV** respectively.

The above DRs can be best illustrated through DTs. We will give an example from the English corpus with a DT for a DR between the English words in the example. Unlike the Arabic DTs, we will give one DT for the output of the parser, since it is labelled with grammatical functions.

5.72 [And when it is said to them, "Believe just as mankind has believed,”] (Qur’an, 2:13)

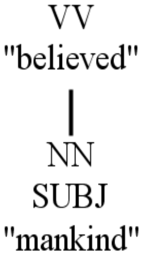


Figure 5.33: A dependency tree for ‘subject-verb’ relation

This DT illustrates the parser’s output for a DR between a head verb and its dependent subject in part of a sentence in the English corpus. As shown in this DT, the parser focuses on the main verb and leaves out the auxiliary verb.

5.5.1.2 ‘Verb-Object’ Relation

The second DR that we use in the parser is that between head verbs and their objects. The rule for this DR is written in the following pattern:

```
[('HD','VV'),'(DT|DP|AT)?','(AJ|NN)*', ('obj','NN|NP')]
```

This pattern collects DRs between a head verb and its dependent object which is the final noun in a given NP or a proper noun. The adjectives or nouns that precede the final noun in an NP are optional. Also, all types of determiner may occur at the beginning of an NP. The following example from the English corpus shows this ‘verb-object’ relation.

5.73 [The likeness of them is as the likeness of one who set to kindle a fire;] (Qur'an, 2:17)

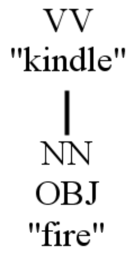


Figure 5.34: A dependency tree for 'verb-object' relation

This DT shows the parser's output for a DR between a head verb and its dependent object. As can be seen in this DT, the parser focuses on the main noun and leaves out the indefinite article.

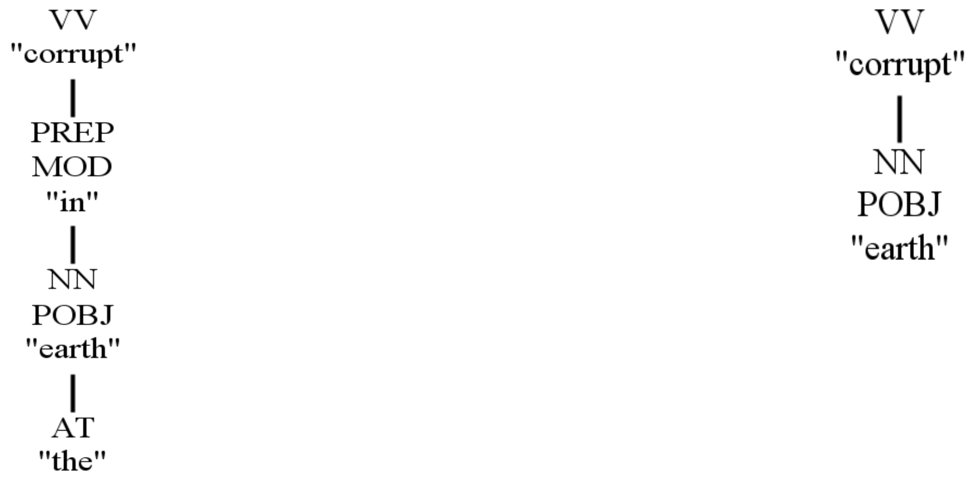
5.5.1.3 'Verb-PP' Relation

This DR focuses on the head verb and the following dependent PP. We are interested in the verb and the noun which is the object of preposition. So, we will exclude the preposition from the output of the parser in this relation. The rule for this DR is captured in the following RE pattern:

```
[('HD','VV'),'PR','(DT|DP|AT)?','(AJ|NN)*',('pobj','(NN|
NP)')]
```

This pattern is similar to the previous one with only two exceptions; the first is the addition of the tag PR for 'preposition' (e.g. *in, at*, *with, of*) and the second one is that the dependent noun is labelled as **pobj**, i.e. "object of preposition". Here we will show a DT for this relation and another one for the parser's output in which the preposition and any nominal determiners or modifiers are excluded.

5.74 [And when it is said to them, "Do not corrupt in the earth, " they say, "Surely we are only doers of righteousness." (i.e. reformers, peacemakers)] (Qur'an, 2:11)



(a) A dependency tree for a verb-PP relation

(b) The parser's output for this construction

Figure 5.35: A dependency tree for 'verb-PP' relation

As can be noticed, the parser selects only the head verb and the main noun in the PP to get the DR.

5.5.1.4 'Preposition-Noun' Relation

The final rule that is used in the English parser is that between any preposition and a following noun. Unlike the previous rule which has the condition that a verb should precede the PP, here we do not write this condition, but collect any preposition and the following noun in a PP. The pattern for this rule is written as follows.

```
[('HD','PR'), '(DT|DP|AT)?', '(AJ|NN)*',('pobj', '(NN|NP)')]
```

Our main interest in this pattern is the noun not the preposition, since we focus on the open-class words only in this study. So, when we implement the process of matching between Arabic and English DRs in the parallel corpus, we filter both the preposition and noun pairs and end up with a very small number of preposition pairs that do not contribute much to the task of bootstrapping the selection process. The following DTs show the output of the parser for this relation.

5.75 [Those are upon guidance from their lord, and those are they who are
the prosperers.] (Qur'an, 2:5)

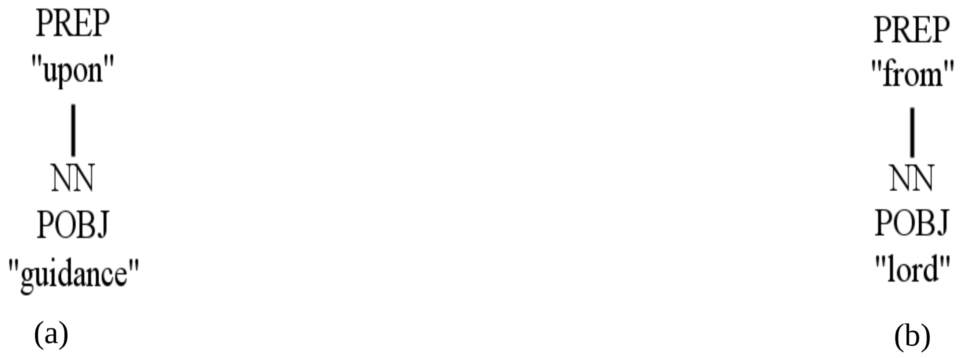


Figure 5.36: Two dependency trees for 'prep-noun' relation

The previous example includes two PPs following each other, but the noun in the first PP has no modifier, while the noun in the second one has a possessive modifier.

Having labelled the Arabic-English parallel corpus with DRs, we proceed to extract a number of 'head-dependent' translational pairs which we filter to obtain the seeds that are used as anchor points to resegment the corpus and bootstrap the selection process once more. The details of this stage of seed extraction will be discussed in the next chapter.

5.6 Summary

In this chapter we have given a brief account of Arabic syntactic structure and the various complexities which Arabic exhibits, thus making Arabic NLP a difficult task. The challenges that we face in handling Arabic computationally, especially MSA, lie in the nature of the language that has the following main characteristics: dropping vowels from the written language, rich and complex morphology, syntactic flexibility and possibility of a pro-drop subject. All these features result in a great deal of ambiguity in the Arabic written form.

In addition, we have discussed the two main approaches to syntactic analysis, namely PSG and DG, throwing more light on the DG theory in which our framework is based. PSG is formulated in terms of constituent analysis. The basic idea of constituency is that groups of words may behave as a single unit or phrase, which is called a constituent. These constituents have a specific element as a head for the whole grouping or constituent. A set of rules has been devised to model the relationship between these phrases (or constituents) called phrase structure rules. Different theories about natural language syntax, which are based on constituency representations, have evolved recently. Among the prominent theories in computational linguistics are Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982), Generalized Phrase Structure Grammar (GPSG) (Gazdar et al., 1985) and Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994). DG, on the other hand, regards the concept of phrase as unnecessary and holds the view that linguistic structure is said to arise through the dependencies between words. DG, which was developed by Tesnière (1959), is distinct from PSGs, as it lacks phrasal nodes, i.e. all nodes are lexical. Thus, a DR holds between a ‘head’ and a ‘dependent’. There are criteria for establishing dependency relations, and for distinguishing the ‘head’ and the ‘dependent’ in such relations. These criteria have been discussed earlier in this chapter. DRs can be analyzed in different notations, i.e. through trees, graphs.etc. A number of theories have emerged recently that are based on DG framework. Among the well-known ones in the field are MTT (Mel’čuk, 1988) and WG (Hudson, 1984). DG-based theories differ with regard to the way they represent DRs in a given construction. Each theory adopts a specific approach in its dependency analysis. As far as Arabic is concerned, we have referred to a number of dependency treebanks in the field, such as PADT, CATiB and QADT.

In conclusion, we have presented the Arabic lexicon-free dependency parser that we developed for obtaining DRs between lexical items in the corpus. The Arabic parser is a shallow one that outputs unlabelled DRs for certain constructions that we are interested in. Despite being shallow, it has proven to be useful, as it has been used to improve the proposer and resulted in a higher F-score after using the DRs in the bi-texts, as will be shown in the coming chapter. Likewise, we have described the similarly English lexicon-free dependency parser that we used to get the DRs between the lexical items in the English corpus. The output of both parsers will be used as input to the proposer or rather ‘parsed proposer’ that deals with DR-labelled corpus when we talk about the bootstrapping techniques in the following chapter.

Chapter 6

Translation of Lexical Items

6.1 Introduction

In this chapter we will describe our attempt at extracting translational equivalents from the Arabic-English parallel corpus. We will call the program for selecting the translational equivalents of lexical items ‘the proposer’. We apply this proposer to the parallel corpus in different stages. First, we apply it to the parallel corpus in its raw unannotated form, and in which case it will be called ‘raw proposer’. Then, we apply it to the corpus after annotating it with POS tags in both Arabic and English, and in which case it will be called ‘tagged proposer’. As indicated earlier in the thesis, we use the Arabic tagger that we have described in chapter 4 to tag the Arabic text. As for the English text, we use the English tagger that was described also in the same chapter to POS tag the English translation. Finally, we annotate the parallel corpus with DRs. The DRs in Arabic and English are carried out using the Arabic and English shallow parsers that were presented in chapter 5. Having annotated the parallel corpus with the basic DRs of interest, we then proceed to extract a number of trusted ‘head-dependent’ Arabic-English pairs which we filter to obtain one-word translation seeds. We use these seeds to resegment the corpus which is basically composed of unpunctuated verses that are mostly long. This procedure of bootstrapping is carried out to enhance the performance of the tagged proposer. All the stages of implementing the proposer on these different types of texts will be discussed in detail in the following sections. We will start with throwing light on the types of lexical relations that hold between words then present the proposer with its different stages. In conclusion, we discuss a proposed algorithm for automatically detecting ambiguous words in a given extracted bilingual lexicon.

6.2 Lexical Relations

There are a number of relations that hold among words and their meanings or senses. Generally speaking, there are two types of lexical relations among words: **syntagmatic** and **paradigmatic**. This distinction, according to Palmer (1981), was first made by De Saussure. The relationship that a lexical item has with other elements inside the sentence is called syntagmatic (Cruse, 2000). This is mainly a syntactic relation. Let us consider the following example.

6.1 The story is exciting.

The word *story* in (6.1) is syntagmatically related to the definite article *the*, the copulative verb *is* is related to the adjective *exciting*, and the noun *story* to the adjective *exciting*. Broadly speaking, when someone comes across a word like *story*, a number of words may occur to their mind. If such words are *is*, *does*, *writer*, etc., it is called a syntagmatic reply because it provides the phrase or the sentence with a required syntactic form; it is the next word in the phrase or the sentence. But if such words are like *tale* or *narrative*, it is called a paradigmatic reply because it chooses another word from a set of semantically related words, not mentioned in the sentence in question. We believe that both types of lexical relations, i.e. syntagmatic and paradigmatic, are complementary to each other because words acquire their meanings from both axes. It is worth mentioning that there is a third type of relation called derivational (Elewa, 2004). This relation is realized if the same word is used but in a different form, e.g. *stories* in the plural form. These types of relations can be illustrated in the following diagram:

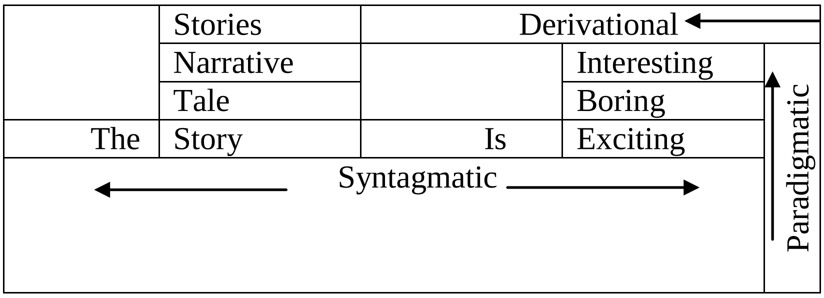


Figure 6.1: Types of lexical relations

Sometimes these lexical relations are referred to as linguistic contexts. It has been pointed out that there are two contrasting ways to think about linguistic contexts, one based on syntagmatic or co-occurrence approach and the other on paradigmatic or substitutability approach (Miller and Charles, 1991). Thus, the syntagmatic relation deals with co-occurrence patterns. Such patterns can be observed on both lexical and structural levels. In other words, lexical items can be combined with each other lexically or syntactically. One of the relationships that holds between words on the syntagmatically lexical level is **collocation**. The phenomenon of collocation can be broadly defined as the 'co-occurrence of words'. Collocation was first introduced by Firth (1957) who defined it by his statement "you shall know the word by the company it keeps". So, collocation refers to the co-occurrence of words more often than by chance. Firth (ibid.) gives the following example to illustrate this point: "One of the meanings of *ass* is its habitual collocation with an immediately preceding *you silly*". When it comes to translation from one language to another, collocations play a great role in this regard. Thus, one word in an SL can be translated into different words in a TL when it collocates with a number of different words. For instance, the English word *heavy* could collocate with a number of words that have different translations in Arabic. The different collocations for this word and their translation into Arabic can be illustrated in the following table.

English Collocations	**Arabic Equivalents**
Heavy rain	مطر غزير *mTr gzyr*
Heavy fog	ضباب كثيف *DbAb kvyf*
Heavy sleep	سبات عميق *sbAt Emyq*
Heavy seas	بحار هائجة *bHAr hA}jp*
Heavy meal	وجبة دسمة *wjbp dsmp*
Heavy smoker	مدخن مفرط *mdxn mfrT*
Heavy industry	صناعة ثقيلة *SnAEp vqylp*

Table 6.1: English collocations and their Arabic equivalents

It is noticeable in the previous table that the English word *heavy* has been translated into different Arabic words according to the word it collocates with. Likewise, an

Arabic word can be translated differently into English according to the word it collocates with. Thus, the word غليظ *glyZ* has been found in the Qur'anic corpus to collocate with the word عذاب *E*Ab* to mean "harsh torment" or with the word ميثاق *myvAq* to mean "solemn compact". As for the structural level, it is concerned with the syntax-based co-occurrence patterns. Thus, as we pointed out earlier, the verb *solve* appears frequently with the noun *problem* in the 'verb-object' relation. In our study we exploit the co-occurrence or syntagmatic approach to linguistic context, where associations are formed between a word and the other words that occur with it in the same phrases and sentences (Miller and Teibel, 1991).

The paradigmatic relations, on the other hand, deal with such relations as 'synonymy', 'hyponymy', 'antonymy', 'homonymy', 'polysemy', etc. It should be noted that the meaning of a word is determined in part by its relations with other words in a given language (Saeed, 2003). It is not the concern of the present study to discuss all these different relations. However, we will shed light on the last two relations mentioned above, namely 'homonymy' and 'polysemy', as they in a way have a bearing on the current task of lexical selection. This is because the task underway seeks to select the translational equivalents of lexical items in the parallel corpus. Some of these lexical items are homonymous or polysemous, which increases ambiguity of Arabic lexical items and consequently makes the selection process more challenging. This will be made clearer when we discuss the way to automatically detect ambiguous words at the end of this chapter. As a matter of fact, handling ambiguous words is not carried out in the current research but we have discussed the way to detect them automatically in a given translation lexicon. Resolving them will be pursued in future research.

What is traditionally described as **homonymy** can be illustrated through the word *bank*. It can have two unrelated meanings, i.e. "financial institution" and "sloping side of a river". Thus, homonyms are, according to Palmer (1981), several words with the same shape. In other words, homonyms are different words which happen to have the same phonological and graphic properties (Cruse, 2000). Lyons (1995) divides homonymy into absolute and partial. According to him, absolute homonyms will satisfy the following three conditions:

(i) They will be unrelated in meaning;

(ii) All their forms will be identical;

(iii) The identical forms will be grammatically equivalent.

The previous example of *bank* is included under absolute homonymy, since it meets all the conditions. However, there are also many kinds of what Lyons (1995) calls partial homonymy. This occurs when there is identity of one form but not all three of the above conditions are satisfied. As a case in point, the verbs *find* and *found* share the form "found", but not "finds", "finding", or "founds", "founding", etc. In addition, "found" as a form of *find* is not grammatically equivalent to "found" as a form of *found*. In this case conditions (ii) and (iii) are not satisfied and this example is thus a case of partial homonymy.

The phenomenon of homonymy is widespread in the Arabic language. The following table shows an example for a nominal homonym with its different meanings.

Homonym	**Meanings**
عين *Eyn*	eye
	spring
	spy
	overseer
	guard
	elite
	notable
	master
	essence

Table 6.2: An Arabic homonym

The word عين *Eyn* has occurred in the Qur'anic corpus with the two meanings of "eye" and "spring". This, clearly, poses a problem for the proposer to choose the right meaning in its proper context. For most of this chapter we are concerned only with finding the commonest translation. We will discuss issues of multiple translations briefly in 6.3.7.

Let us now move on to **polysemy**. Whereas homonymy is a relation that holds between two or more distinct lexemes, polysemy, i.e. multiple meaning, is a property of single lexemes (Lyons, 1995). In other words, the term polysemy is used to describe a single word-form with several different but closely related meanings.

Lyons (1977) cites the word *mouth* as an example of polysemy. This word is polysemous because it can mean either “organ of a body” or “entrance of a cave”. Similarly, we can talk about the ‘head’ of a person or the ‘head’ of an organization. However, it should be noted that a single word may denote a particular set of things in one language but does not denote the same set of things in another language (Rouhani, 1994). In Arabic, for instance, the two meanings of the word *head* are rendered differently as رأس *raOos* “head” and رئيس *ra}iys* “president” respectively.

As a matter of fact, the difference between homonymy and polysemy is not always clear-cut in particular instances (Lyons, 1995). A similar view is expressed by Kilgarriff (2007) with regard to word senses when he points out that “there are no decisive ways of identifying where one sense of a word ends and the next begins”. According to Lyons (ibid), there are two criteria that are usually used to judge words to be polysemes. These are etymology (the historical source of the words) and relatedness of meaning. For example, most native speakers of English would probably think that *bat*$_1$ “a furry mammal with membranous wings” and *bat*$_2$ “an implement for striking a ball in certain games” are different lexemes. In fact, these two words differ in respect of their historical source, *bat*$_1$ being derived from a regional variant of Middle English “bakke” and *bat*$_2$ from Old English ‘batt’ meaning “cudgel”. However, it sometimes happens that lexemes which native speakers of the language classify as being semantically unrelated have come from the same source. An example of this is the homonyms *sole*$_1$ “bottom of foot or shoe” and *sole*$_2$ “kind of fish”. Similarly, the words *pupil*$_1$ “student” and *pupil*$_2$ “pupil of the eye” are historically from the same origin, but semantically unrelated and are thus homonyms.

As far as our parallel corpus is concerned, we have noticed that an Arabic word can have different related meanings, i.e. be polysemous, according to the following noun it qualifies. Thus, the same word can have different connotations according to the word it collocates with. This is reflected in the translation, where such a polysemous word is translated with different English words in different contexts. This, consequently, constitutes a problem for the proposer to select the right TL word that conveys the meaning of a polysemous Arabic word in a given context. A tangible example can make this point clear. The following table shows the different meanings for the polysemous word عظيم *EZym* “great”.

Collocation	Reference Translation	Comments
عذاب عظيم *E*Ab EZym*	a tremendous torment	The word *EZym* here has a bad connotation as it qualifies the noun *E*Ab* “torment” and so is translated as “tremendous”.
أجر عظيم *Ojr EZym*	a magnificent reward	The word *EZym* here has a good connotation as it qualifies the noun *Ojr* “reward” and so is translated as “magnificent”.
ظلم عظيم *Zlm EZym*	a monstrous injustice	The word *EZym* here has a very bad connotation as it qualifies the noun *Zlm* “injustice” which refers to the act of associating others with God (Allah). Thus, it is translated as “monstrous”.

Table 6.3: Different collocations for an Arabic polysemous word from the Qur’anic corpus with different translations

It is worth noting that we do not usually make the same distinctions between individual words in writing and speech. Thus, the words *lead$_1$* “metal” and *lead$_2$* “dog’s lead” are spelt in the same way, but pronounced differently. This case is normally termed **homography**. The words *rite* and *right*, on the other hand, are spelt differently but pronounced in the same way. They are, thus, an example of **homophony.** In fact, there are some homonyms and homophones that are nearly antonyms, e.g. the homonyms *cleave$_1$* “part asunder” and *cleave$_2$* “unite” and the homophones *raise* and *raze* (Palmer, 1981).

Homographs are very common in the modern form of Arabic due to the lack of diacritics. According to Elewa (2004), Arabic is full of homographs which are distinguished in pronunciation. Thus, change of diacritics (or short vowels) makes a different base form and ignoring these diacritics produces such homographs. Elewa (ibid.) gives the undiacritized form ورد *wrd* as an example to show that Arabic is full of homographs. This undiacritized form can be diacritized to give the following word-forms, وَردٌ *wardN* “flowers”, وِردٌ *wirdN* “portion”, وَرَدَ *warada* “came”, ورَّدَ *war~ada* “flowerize”, وَ رَدَّ *wa rad~a* “and replied” and وَ رُدَّ *wa rud~a* “and was replied”. In fact, these homographs are frequently found in the corpus which we use now in its undiacritized form.

We will discuss an algorithm for automatically detecting such ambiguous words in a translation lexicon at the end of this chapter. We have carried out this step with a view to handle these words. However, we managed only to do the automatic detection and did not have time to handle them. This will be dealt with in future work.

6.3 Extraction of Translational Equivalents

Now we present the proposer that extracts translational equivalents from the parallel corpus. As pointed out at the beginning of this chapter, there is a general algorithm that we use to extract translational equivalents from the parallel corpus in its unannotated form. Then we modify this algorithm to handle the corpus after being annotated with POS tags. Accordingly, in our attempt to translate lexical items in the parallel corpus we deal with both raw and linguistically annotated texts. Our general framework is applicable to both types of texts with slight modifications. We have carried out a number of experiments on both texts to show the difference between them using different constraints. Thus, the first experiment is to apply our framework to raw texts that have no linguistic annotations. The second and third experiments are concerned with linguistically annotated texts. The first of these tackles texts annotated with POS tags. As for the second one, it handles texts annotated with DRs. As noted earlier, we use this step to extract a number of seeds that we trust to resegment the corpus and execute a bootstrapping technique to enhance the proposer. As a preprocessing step, we have developed a stemmer for both Arabic and English. This has been done to test the three different types of proposer on both word-forms as they are in the corpus and stemmed forms after stemming the parallel corpus. We should make it clear that we are mainly concerned with the translation of open-class words, i.e. verbs, nouns and adjectives. We focus on these words because they bear the semantic load in a given sentence. So, we do not attempt to translate other categories such as particles, prepositions, conjunctions ...etc. However, in our initial attempt that is applicable to raw texts we cannot filter the parallel corpus and retain only the content words. This is because in this experiment the texts are not POS tagged. We do this filtering in the experiments that we apply to POS tagged texts and DR-labelled texts. As a reminder, we should point out that all our work is lexicon-free. We do not hand-code a bilingual lexicon. We automatically extract this lexicon from our parallel verse-aligned corpus.

We start with throwing light on the used parallel corpus. Second, we discuss the preprocessing steps that we have taken prior to carrying out the task of selecting lexical equivalents. Third, we explain our general algorithm to extract lexical equivalents from the parallel corpus. Fourth, we discuss and evaluate the three different types of proposer, as well as the bootstrapping techniques that we execute to improve the final proposer. Finally, we present an algorithm for automatically detecting ambiguous words in a given bilingual lexicon.

6.3.1 Parallel Arabic-English Corpus

As explained earlier in the thesis, the parallel corpus that we use in this study consists of the Qur'an and its English translation. We align every Arabic verse as a whole with its translated English verse on the same line in a text file. We should recall that the Qur'an is basically a diacritized text but we have used an undiacritized version in our framework. We have discussed the reasons for using this corpus in its undiacritized form earlier in the thesis.

The Arabic original text of the Qur'an contains 77,800 words. The diacritized version of the Qur'an contains 19,268 distinct word-forms (or word types), whereas the undiacritized version contains 14,952 distinct word-forms. As we can notice, the number of words has collapsed because many words share the same orthographic forms but are different with regard to diacritic marks. Thus, when diacritics have been removed, many different diacritized forms have been conflated to fewer forms. As for the English translation that we use, it contains 162,252 words after normalization (i.e. lowercasing all words and removing what is between brackets as will be illustrated below). However, it unexpectedly contains only 5,531 distinct words. We have noticed that the Arabic corpus contains 77,800 words or rather tokens, whereas the English translation contains 162,252. This difference in number of words between English and Arabic may be due to the fact that Arabic is characterized by its rich morphology where clitics are attached to words, thus forming complex words that need to be decomposed into a number of words when translating into English. This is not the case with English. Therefore, the English words are bigger in number than their Arabic counterparts. Paradoxically, the English distinct word-forms, which consist of only 5,531, are fewer than the Arabic distinct word-forms which either contain 19,268 in the diacritized version or 14,952 in its undiacritized version. This may be also due to the fact that Arabic free and cliticized closed-class words, e.g. prepositions, conjunctions and pronouns, are translated into separate function words in English, which have high frequency in the corpus. The conjunction *and* and the definite article *the*, for instance, have occurred 9072 and 9068 times respectively, as will be shown below.

It has been stated at the beginning of the thesis that verses in the Qur'an vary in length. Some of them are short, while many of them are very long. Accordingly, a verse may contain one sentence or more. In fact, a Qur'anic verse could contain up to 129 words. In addition, we have pointed out that there are no punctuation marks in the Qur'anic verses, and thus there are no sentence boundaries. This, in turn, presents a difficult challenge for the proposer in its attempt to get the right equivalent. Here is a sample of our parallel corpus from a short chapter (the start of سورة العلق Surat[15] *Al-Alaq* "The Clot").

*AqrO bAsm rbk Al*y xlq*	Read: In the Name of your Lord Who created
xlq AlInsAn mn Elq	Created man from clots.
AqrO wrbk AlOkrm	Read: And your Lord is The Most Honorable
*Al*y Elm bAlqlm*	Who taught by the pen.
Elm AlInsAn mA lm yElm	He taught man what he did not know.
klA In AlInsAn lyTgY	Not at all! Surely man does indeed (grow) inordinate
On r|h AstgnY	That he sees himself becoming self-sufficient.
In IlY rbk AlrjEY	Surely to your Lord is the Returning.
*OrOyt Al*y ynhY*	Have you seen him who forbids
*EbdA I*A SlY*	A bondman when he prays?
OrOyt In kAn ElY AlhdY	Have you seen in case he is upon guidance

[15] The word 'surat' refers to one of the chapters of the Qur'an.

Ow Omr bAltqwY	Or he commands (people) to piety?
*OrOyt In k*b wtwlY*	Have you seen in case he cries lies and turns away?
Olm yElm bOn Allh yrY	Does he not know that Allah sees?

Figure 6.2: A sample of the parallel corpus (Qur'an, 96:1-14)

6.3.2 Preprocessing and Data Preparation

We have taken a number of preprocessing steps prior to writing our proposer of lexical equivalents. These steps range from simple procedures such as normalizing Arabic and English texts as well as generating a frequency list of all word-forms in the corpus to more complicated procedures such as carrying out a lexicon-free corpus-based stemming for both Arabic and English. These three steps will be discussed in the following sections.

6.3.2.1 Text Normalization

Text normalization "is a process by which text is transformed in some way to make it consistent in a way which might not have been before" (Mubarak et al., 2009a). We have normalized the English corpus so that it can be similar to the original Arabic corpus. The English translation initially contains all forms of punctuation that are lacking in the Arabic text. Due to this inconsistency between the Arabic and English texts we had to remove the punctuation marks from the English text. The English words have been also lowercased so that there is no distinction between *The* and *the*. Moreover, as stated in chapter 2, the translation we are using contains some explanatory parentheses as a way of explanation or for grammatical reasons. We have removed these parentheses so as to have a word-to-word matching, if possible, between the SL and TL texts. We have used regular expressions to do all these types of text normalization. Here is an example with its translation before and after normalization.

Arabic Verse	**English Translation**	**Translation after Normalization**
وما يدريك لعله يزكى *wmA ydryk lElh yzkY*	And what makes you (The prophet) realize whether he (The blind man Abdullah ibn Umm Maktûm) would possibly (try) to cleanse himself.	and what makes you realize whether he would possibly to cleanse himself

Table 6.4: An example of text normalization

As we can notice, the first word, the conjunction *and*, has been lowercased so as not to differentiate between the uppercase and lowercase forms of this conjunction. In addition, all words and clauses between round brackets, i.e. the parentheses, have been deleted from the translation. Sometimes the outputted translation is not a perfect one after removing parentheses. For example, the clause *would possibly (try) to cleanse himself* has been rendered as *would possibly to cleanse himself*, which is not grammatically right, because the infinitive *to* has been retained, while it should be removed. Nevertheless, this will not have much effect on our research purposes, since we are mainly concerned with the translation of lexical items and not of whole clauses or sentences. Finally, the period, which marks the end of a sentence and of a verse in this case, has been deleted from the English translation.

6.3.2.2 Frequency List Generation

A frequency list shows the words which make up the texts in the corpus, together with their frequencies of occurrence. Such a frequency list is produced by identifying each word-form found in the text, counting identical forms and listing them with their frequencies in a chosen sequence (Barnbrook, 1996). It is noteworthy that a distinction is often made between tokens and types. A token is an individual occurrence of any word-form (or word type). Thus, the word *the* may occur 100 times, for instance, in a corpus. It is thus a word-form (or type) but with 100 tokens. To do this frequency list we have generated two dictionaries for both Arabic and English texts. Each dictionary contains a given lexical item along with the number of times it has occurred in the corpus (i.e. the frequency list) as well as the actual numbers of verses in which it has occurred. To show this procedure, we will give some examples from the Arabic and English parallel corpus. The following table shows Arabic word types with their frequency in the Arabic corpus, their possible POS tag and their percentage with regard to the total number of tokens in the corpus, which is, as mentioned above, 77,800.

Word Type	**Freq.**	**Possible POS Tags**	**Freq./Total Tokens**
من *mn*	2764	PREP/RELPRO/VV	0.03552
الله *Allh*	2153	NN	0.02767
في *fy*	1185	PREP	0.01523
إن *In*	966	PART	0.01241
آمنوا *|mnwA*	263	VV	0.00338
كفروا *kfrwA*	189	VV	0.00242
الكتاب *AlktAb*	163	NN	0.00209
الصلاة *AlSlAp*	58	NN	0.00074
الزكاة *AlzkAp*	26	NN	0.00033
كتب *ktb*	23	NN/VV	0.00029

Table 6.5: Examples of Arabic word types and their frequency

The previous Arabic words are listed in descending order with regard to their frequency. We can notice that the function words such as من *mn*, في *fy* and إن *In* have the highest score of frequency besides the word الله *Allh* (Allah) which occurs more frequently in the Qur'anic corpus. The previous function words can be diacritized to give different interpretations. Thus, the word *mn* can be either the preposition مِن *min* "from", the relative pronoun مَن *man* "who", or the perfective verb مَنَّ *man~a* "has been bounteous/conferred a favour upon". The first two parts of speech are the most frequent in the corpus. Similarly, the word *In* can be diacritized to give either the emphatic particle إنَّ *In~a* "surely/indeed" or the conditional particle إنْ *Ino* "if". Other words are either verbs or nouns that have different scores of frequency. The last word in the previous table, namely كتب *ktb* can be diacritized to give different full underlying forms. Thus, it can be the perfective verb كَتَبَ *kataba* "wrote", the passive of the same verb كُتِبَ *kutiba* "was written", the causative verb كَتَّبَ *kat~aba* "cause (someone) to write/dictate”, the passive of this verb كُتِّبَ *kut~iba* “(someone) was made to write”, or the plural noun كُتُب *kutub* “books”. It should be noted that the frequency of occurrence for the above examples are for the mentioned word-forms not the lemmas. Thus, the word الله *Allh* has occurred 2153 times in the corpus in this form. However, it has also occurred in different forms, i.e. with attached clitics such as بالله *bAllh* “with Allah”, والله *wAllh* “and Allah”, etc. Every form has a different frequency. In actual fact, we could not do a frequency list for lemmas since we cannot do lemmatization without having a lexicon of words.

As for the English examples, the following table shows the same statistics as shown with regard to Arabic above. We should recall that the total number of English tokens in the corpus is 162,252.

Word Type	**Freq.**	**Possible POS Tags**	**Freq./ Total Tokens**
and	9072	CJ	0.05591
the	9068	AT	0.05588
in	3400	PR	0.02095
Allah	2703	NP	0.01665
book	250	NN	0.00154
believe	214	VV	0.00131
prayer	78	NN	0.00048
disbelieve	55	VV	0.00033
zakat (poor-dues)	30	NN	0.00018
write	13	VV	0.000080

Table 6.6: Examples of English word types and their frequency

The English words are also listed in descending order with respect to their frequency. We can notice here also that the function words have the highest score of frequency. In addition, the word *Allah*, which is left as it is in the English text without being translated into the word *God*, has the highest score among open-class words in the corpus. This is expected because the name of God is mentioned many times in the Qur’an.

6.3.2.3 Arabic & English Stemming

So as to experiment with both the word-forms as they are and the canonical forms in Arabic and English we have developed a stemmer for Arabic and English texts. We use a lexicon-free approach to stemming Arabic and English words, focusing on clustering similar words in the corpus that share at least three letters after stripping off affixes. We are thus more interested in grouping such words and reducing them under one stem, irrespective of whether the reduced form is the legitimate stem or not. In the following lines we give a general overview of what is meant by stemming and the difference between it and other related processes in the field. Then we describe our approach to stemming both Arabic and English words.

6.3.2.3.1 Introduction

First, we should distinguish between a number of terms that are usually related. These terms are tokenization, segmentation, stemming and lemmatization. The word **tokenization** refers to the process of “cutting a string into identifiable linguistic units that constitute a piece of language data” (Bird et al., 2009). In other words, a stream of characters in a natural language text must be broken up into distinct meaningful tokens before any processing beyond the character level can be performed (Kaplan, 2005). Here is an example that illustrates this point.

6.2 The cat sat on the mat.

Unlike humans, a computer cannot intuitively see that there are 6 words. To a computer this is only a series of 17 characters. A process of tokenization could be used to split the sentence into word tokens. Thus the above example can be tokenized as follows:

The	cat	sat	on	the	mat

Figure 6.3: A tokenized English sentence

As far as Arabic is concerned, tokenization is the process of segmenting clitics from stems. This is very common in Arabic since prepositions, conjunctions, and some pronouns are cliticized (orthographically and phonologically fused) onto stems (Diab et al, 2004). Clitics can be attached to different categories of words. They can be attached either to a closed-class word (function word) e.g. وفي *wfy* “and in” or to an open-class word e.g. والكتاب *wAlktAb* “and the book”. These clitics can be classified into the following types:

1- Proclitics, which occur at the beginning of words. These include mono-consonantal conjunctions (such as و *w*-, “and”, ف *f*-, “then”), prepositions (e.g. ب *b*-, “in, at” or “by”, ل *l*-“to, for”),…etc.
2- Enclitics, which occur at the end of words. In Arabic enclitics are complement pronouns, which include genitive (or possessive) pronouns (such as ه –*h*, “his”, ها -*hA* “her”) and object pronouns (such as ه –*h* “him”, ها *hA* “her”).

For example, the word-form وبصلها *wbSlhA* “and its onions” can be tokenized into the proclitic و *w* “and”, the stem بصل *bSl* “onions” and the enclitic ها *hA* “its”.

There is a limit on the number of clitics that can be attached to a word stem. In case of nouns, a given noun can comprise up to four tokens. For example, the lexical item وبحسناتهم *wbHsnAthm* “and with their virtues” can be tokenized as follows:

Conjunction	**Preposition**	**Stem with affixes**	**Genitive pronoun**
و *w*	ب *b*	حسنات *HsnAt*	هم *hm*

Table 6.7: A tokenized Arabic noun

Verbs can also comprise up to four tokens. The lexical item وليكتبهم *wlyktbhm* “and to write them” can be tokenized as follows.

Conjunction	**Complementizer**	**Stem with affixes**	**Object pronoun**
و *w*	ل *l*	يكتب *yktb*	هم *hm*

Table 6.8: A tokenized Arabic verb

Sometimes the term **segmentation** is interchangeable with tokenization. However, while tokenization delimits boundaries between syntactically functional units in a word, i.e. stems and clitics, segmentation is a method to determine the boundaries between all the word parts. This includes inflections, stems, or clitics (Mohamed and Kübler, 2010a; 2010b). The previous examples can be segmented as follows.

Conjunction	**Preposition**	**Stem**	**Feminine Plural Suffix**	**Genitive Pronoun**
و *w*	ب *b*	حسن *Hsn*	ات *At*	هم *hm*

Table 6.9: A segmented Arabic noun

As for the verb, it can be segmented as follows.

Conjunction	**Complementizer**	**Imperfect Verb Prefix**	**Stem**	**Object Pronoun**
و *w*	ل *l*	ي *y*	كتب *ktb*	هم *hm*

Table 6.10: A segmented Arabic verb

It is clear that the two previous examples have more segments than tokens.

In our framework we do not exploit segmentation or tokenization, since we have not manually constructed a lexicon. The third term is **stemming**, which we make use of in our framework. Stemming is the process for reducing a word to its stem, base or root form. This means that different morphological variants of a word can be conflated to a single representative form. For instance, *play*, *played*, *plays* and *playing* are grammatically conditioned variants of the same lexeme PLAY[16]. A lexeme, according to Lyons (1968), "refers to the more abstract units which occur in different inflectional 'forms' according to the syntactic rules involved in the generation of sentences." In other words, a lexeme is an abstract kind of word. It is not strictly speaking something that can be uttered or pronounced; only the word-forms that belong to it can be (Carstairs-McCarthy, 2002). Thus, a lexeme contrasts with the concrete word-shape (or word-form) which is defined only by its spelling or

[16] Lexemes are conventionally represented in capitals.

pronunciation (Hudson, 1995). Accordingly, word-forms are grammatical variants of a lexeme. This indicates that lexemes, as pointed out by Cruse (2000), can be thought of as groupings of one or more word-forms and are thus the headwords in a dictionary. It should be noted that lexemes can cover more than one word-form as can be noticed in the lexeme PLAY above. Nonetheless, one word-form can express two different lexemes. For example, CYCLE is used both as a noun and verb. In this case, the noun and verb *cycle* are two different lexemes (Hudson, ibid).

Stemming reduces all these variants (i.e. “played”, “plays” and “playing”) to its stem, namely *play*. However, stemming faces a problem with irregular forms. For example, the plural of such nouns as *man*, *tooth* and *wife* are “men”, “wives” and “teeth”. A stemmer for English strips word endings (suffixes). Thus, when it encounters word-forms such as “go”, “goes”, “went”, “gone”, “going”, it will strip all of them to the reduced form *go*, except the form *went* which is irregular. This is illustrated in table (6.11) below.

Base form	**3rd Person singular present tense**	**Past tense**	**Past participle**	**Progressive participle**	**Stemmed form**
go	goes	went	gone	going	go

Table 6.11: Different forms of an English verb with its stemmed form

As we have noticed in the previous table, all the morphological variants are reduced to the base form *go*, except the irregular past form *went*. The stemmed form here happens to be exactly similar to the base form. However, sometimes the stemmed form is not a legitimate form (a lexicon headword). In other words, a stemmer may chop off the end of a word but return a form that is not the base or dictionary form of it. For instance, the form *changing* may be stemmed to “chang”, which is an illegitimate form. Here comes the notion of **lemmatization**. It “is the process of reducing a set of word forms to a smaller number of more generalised representations” (Fitschen and Gupta, 2008). Both stemming and lemmatization share a common goal of reducing a word to its base form or root. However, lemmatization often involves usage of vocabulary and morphological analysis, as opposed to simply removing the suffix of a word. Thus, a lemmatizer reduces a number of word variants to its lemma (or lexicon headword). So, the above form *changing* is lemmatized as “change”, which is the lemma (or the legitimate form). The difference between stemming and lemmatization can be shown in the following table

Word-form	**Base form**	**Stemmed form**	**Lemmatized form**
walking	walk	walk	walk
changing	change	chang	change
better	good	_	good

Table 6.12: Difference between stemming and lemmatization

It is noticeable in the previous table that the stemmed and lemmatized forms of the first example, i.e. *walking*, match the base form. As for the second example *changing*, we find that the stemmed form “chang” is not similar to the base form, but the lemmatized form “change” is the same like the base form. The third example, however, shows a lemmatized form that matches the base form, but there is no stemmed form, since *better* is an irregular adjective. It is worth mentioning that the most common algorithm for stemming English is the Porter Stemmer (Porter, 1980; Willett, 2006).

6.3.2.3.2 Approaches to Arabic Stemming

It is time now to discuss Arabic stemming and the approaches adopted in this regard. Arabic, a highly inflected language with complex orthography, needs good stemming for effective text analysis (Thabet, 2004). Morphological change in Arabic results from the addition of prefixes, suffixes and infixes. Thus, simple removal of suffixes is not as effective for Arabic as it is for English (Taghva et al., 2005). In English and many other European languages, stemming is mainly a process of suffix removal (Lovins, 1968; Porter, 1980). Different approaches have been taken towards Arabic stemming (Larkey et al., 2002; 2007). They can be summarized as follows:-

- Manually constructed dictionaries of words. This approach involves developing a set of lexicons of Arabic stems, prefixes and suffixes, with truth tables indicating legal combinations. In other words, each word uses a unique entry in a lookup table. In this technique, words could be stemmed via a table lookup.

- Light stemmers, which remove prefixes and suffixes. This approach refers to a process of stripping off a number of prefixes and/or suffixes, without any attempt to handle infixes, or recognize patterns and find roots. Light stemming can correctly conflate many morphological variants of words into large stem classes. However, it can fail to conflate other forms that should be grouped together. For example, broken (or irregular) plurals for nouns do not get conflated with their singular forms. Similarly, some verbs in the past tense do not get conflated with their present tense forms (e.g. weak verbs). This occurs because of the internal differences.
- Morphological analyses which attempt to find roots. Several morphological analyzers have been developed for Arabic. Among those known in the literature are Xerox Arabic Morphological Analysis and Generation (Beesley, 1998a, Beesley, 2001), Buckwalter Arabic Morphological Analyzer (Buckwalter, 2002), Sakhr (Chalabi, 2004a). These analyzers find the root, or any number of possible roots for each word.
- Statistical stemmers, which group word variants using clustering techniques. In this technique, association measures between words are calculated based on shared unique N consecutive letters (i.e. the same shared root). Words that have a similarity above a predefined threshold are clustered and represented with only one word (Goweder et al., 2008). This statistical method can provide a more language-independent approach to conflation (Larkey et al, 2007).

6.3.2.3.3 Our Approach to Arabic Stemming

We apply light stemming, using a data-driven approach. The current approach groups similar word variants based on shared word-initial and word-final n-grams (i.e. supposed roots), which should be at least three letters. Having conditioned the existence of shared roots, it does light stemming by removing any letters that resemble a number of listed inflectional affixes (i.e. prefixes and suffixes). This is carried out on our corpus, which is the Qur’anic text in its undiacritized form. Thus, when there is a word with one form that has no similar forms in the corpus, it is left as it is without stemming. Thus, every word in our corpus is compared with other related words to get the stem. This is applied to both Arabic and English, but with some differences, as will be shown later. We should remember that we do not have a lexicon of words. Thus, what concerns us more, as noted earlier, is to group similar words and reduce them under one stem. In some cases we get the legitimate stem of the clustered words, but in other cases the reduced form is not the right stem. For our purpose of extracting translational equivalents from the parallel corpus we need to conflate similar words in the corpus into one reduced form so as to have a better chance of getting the right TL word. This is because Arabic is morphologically rich, where many morphological variants express the same semantic meaning of a lexical item. In addition, since we rely on statistical information about the co-occurrence of words in the corpus to obtain the lexical equivalents, grouping similar words under one stem will increase the frequency of occurrence for such a stem and thus increase the chance of getting the TL word right.

Our approach to Arabic stemming is illustrated in the following figure.

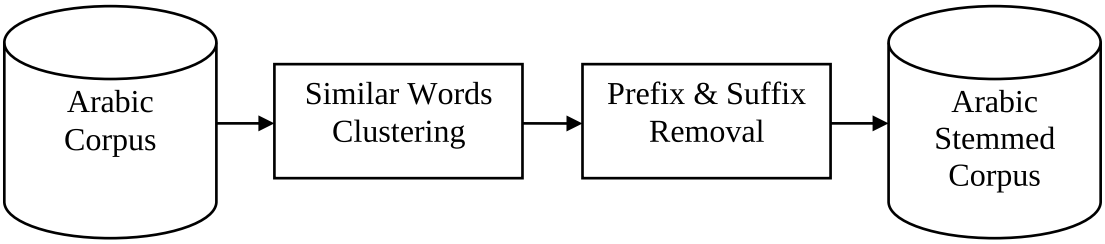


Figure 6.4: Our Approach to Arabic stemming

We try to obtain the stem of Arabic words by carrying out the following steps:

1- We try to get the Arabic stem based on comparing the beginning of words in our corpus. We impose the condition that the words in question begin with the same letter. If so, we go through the remaining letters and truncate the final letters that are similar to a group of listed suffixes, but on condition that there should be at least three letters before removing the suffixes. This step pertains to suffix removal. Here are some examples.

Clustered Words	**Gloss**	**Removed Suffixes**	**Possible Stems**
وجد *wjd*	found	——	وجد *wjd*
وجدتم *wjdtm*	you (pl.) found	تم *tm*	وجد *wjd*
وجدها *wjdhA*	(he) found her/it	ها *hA*	وجد *wjd*
وجدنا *wjdnA*	we found	نا *nA*	وجد *wjd*
وجدوا *wjdwA*	they found	وا *wA*	وجد *wjd*
وجدت *wjdt*	I/you found	ت *t*	وجد *wjd*
وجدكم *wjdkm*	(he) found you (pl.)/ your means (noun)	كم *km*	وجد *wjd*
أصاب *OSAb*	afflicted	——	أصاب *OSAb*
أصابها *OSAbhA*	afflicted (masc.) her/it	ها *hA*	أصاب *OSAb*
أصابت *OSAbt*	afflicted (fem.)	ت *t*	أصاب *OSAb*
أصابك *OSAbk*	afflicted you (sing.)	ك *k*	أصاب *OSAb*
أصابهم *OSAbhm*	afflicted them	هم *hm*	أصاب *OSAb*
أصابكم *OSAbkm*	afflicted you (pl.)	كم *km*	أصاب *OSAb*
أصابه *OSAbh*	afflicted him	ـه *h*	أصاب *OSAb*
مال *mAl*	wealth	——	مال *mAl*
مالك *mAlk*	your (sing.) wealth/ possessor/ Malik (proper noun)	ك *k*	مال *mAl*
ماله *mAlh*	his wealth	ـه *h*	مال *mAl*
مالا *mAlA*	wealth	ا *A*	مال *mAl*

Table 6.13: Examples of clustered stemmed words with suffixes removed

We should make it clear that the word وجدكم *wjdkm* in the previous table has occurred in the corpus as a noun whose full form is وُجْدِكُمْ *wujodikumo* "your means". Part of the verse in which this word-form has occurred is given in (6.3) below.

6.3 أسكنوهن من حيث سكنتم من وجدكم

Osknwhn mn Hyv skntm mn wjdkm

Make them dwell (in some part of the housing) where you are dwelling, according to your means. (Qur'an, 65: 6)

However, the other verbal meaning whose full form is وَجَدَكُم *wajadakum* "he found you" has not occurred in the corpus. Thus, the stemmer has mistakenly clustered this word along with the other unrelated words which refer to the verb "found". Nonetheless, the stem وجد *wjd* is correctly the same for all words.

Similarly, the word مالك *mAlk* is ambiguous because it can have three different interpretations when diacritics are restored to the word. The first sense is manifested in the diacritized surface forms مَالُكَ *maAluka*, مَالَكَ *maAlaka*, مَالِكَ *maAlika* (with different case markers) "your wealth". The second sense is manifested in the diacritized surface forms مَالِكُ *maAliku*, مَالِكَ *maAlika*, مَالِكِ *maAliki* "possessor". The third sense is used as "a proper noun" referring to the angel who is guarding the hell-fire. The second and third interpretations are the stem forms, whereas the first one is the stem with an attached possessive pronoun. The second and third meanings have occurred in the Qur'anic corpus as shown in (6.4) and (6.5) below, but the first meaning for this word-form did not occur in the corpus.

6.4 قل اللهم مالك الملك

ql Allhm mAlk Almlk

Say, "O Allah, Possessor of the Kingship (3: 26)

6.5 ونادوا يا مالك ليقض علينا ربك

wnAdwA yA mAlk lyqD ElynA rbk

And they will call out, "O Malik, (keeper of Hell) let your Lord decree upon us!" (43: 77)

We should point out that the word-form ليقض *lyqD* in (6.5) is translated as "let (your Lord) decree" in our corpus. However, some other translators render this word-form as "let (your Lord) make an end of / put an end to". The translations of the word-form are different because the translators have interpreted the meaning of the homonymous verb differently. But it is not the concern of our current study to study the differences and judge them as far as accuracy is concerned.

With regard to the stemmer's output, it has clustered the word-forms مال *mAl*, مالك *mAlk*, ماله *mAlh* and مالا *mAlA* in one category and reduced them to the stem مال

mAl. This stem is correct for the word-forms مال *mAl* "wealth", ماله *mAlh* "his wealth" and مالا *mAlA* "wealth (acc.)" as well as the first sense of the word-form مالك *mAlk* "your wealth". However, it is not the correct stem for the second and third senses of the same word-form, which mean "possessor" and "proper noun" respectively.

2- We apply the same previous technique but in the reverse way. We check the end of words in our corpus and make sure that the words in question end with the same letter. If so, we go through the remaining letters and truncate the first letters that are similar to a group of listed prefixes, but on condition that there should be at least three letters before removing the prefixes. This step pertains to prefix removal. Here are some examples.

Clustered Words	**Gloss**	**Removed Prefixes**	**Possible Stems**
ختم *xtm* يختم *yxtm* وختم *wxtm* نختم *nxtm*	sealed (he) seals and (he) sealed (we) seal	—— ي *y* و *w* ن *n*	ختم *xtm* ختم *xtm* ختم *xtm* ختم *xtm*
جمع *jmE* نجمع *njmE* فجمع *fjmE* الجمع *AljmE* يجمع *yjmE* وجمع *wjmE*	gathered (we) gather so/then (he) gathered the gathering (he) gathers and (he) gathered	—— ن *n* ف *f* ال *Al* ي *y* و *w*	جمع *jmE* جمع *jmE* جمع *jmE* جمع *jmE* جمع *jmE* جمع *jmE*
ماء *mA'* بماء *bmA'* وماء *wmA'* كماء *kmA'* الماء *AlmA'* سماء *smA'*	water with water and water as water the water heaven	—— ب *b* و *w* ك *k* ال *Al* س *s*	ماء *mA'* ماء *mA'* ماء *mA'* ماء *mA'* ماء *mA'* ماء *mA'*

Table 6.14: Examples of clustered stemmed words with prefixes removed

In the first example in this table the three verbal word-forms يختم *yxtm*, وختم *wxtm*, نختم *nxtm* were clustered with the word-form ختم *xtm* after the prefixes were removed from all of them. Then all of them were correctly stemmed to ختم *xtm*. As for the second example, all the word-forms were correctly grouped after the prefixes were truncated and were all stemmed correctly. However, all conflated word-forms are verbs with the exception of one word-form الجمع *AljmE* “the gathering” which is a noun, but it was anyway stemmed correctly. The final example in the table is about different word-forms for the noun ماء *mA’* “water”. The word-forms بماء *bmA’*, وماء *wmA’*, كماء *kmA’* and الماء *AlmA'* were clustered correctly with the word-form ماء *mA’* after prefixes were eliminated and thus stemmed correctly. But the final word-form سماء *smA’* “heaven” is not related to the other word-forms and thus was clustered wrongly and then also stemmed wrongly as ماء *mA’*, while its stem should be the same word-form سماء *smA’*. The reason for this is that the stemmer has identified the first letter in the word-form as a prefix because it resembles the future tense prefix, while it is a main letter of the stem and not a prefix in this case.

3- In the third step we combine between the first and second techniques. When there are variants for a given word, the stemmer conflates them to one form. As noted above, a word should have at least three letters before stripping off prefixes and suffixes. Prefixes include proclitics that are attached before prefixes and suffixes include enclitics that come after suffixes at the end of words. The following table shows a list of Arabic prefixes and suffixes that we remove from words.

Prefixes		**Suffixes**	
Conjunctions و *w* “and” or ف *f* “then”, Question Particle أ *O* “is it true that”, Prepositions ب *b* “with”, ل *l* “to”, ك *k* “as”	The Definite Article ال *Al*, Tense Markers ي *y*, ت *t*, ن *n*, س *s*, ا *A*	Feminine Marker ـة *p*, Dual Markers ان *An*, تان *tAn*, Plural Markers ون *wn*, ين *yn*, ات *At*, Agreement Markers ت *t*, تا *tA*, تما *tmA*, تم *tm*, تن *tn*, ن *n*, ا	Genitive Pronouns ي *y* “my”, نا *nA* “our”, ك *k*, كما *kmA*, كم *km*, كن *kn* “your”, ـه *h* “his”, ها *hA* “her”, هما *hmA*, هم *hm*, هن *hn* “their”, Object Pronouns ني *ny* “me”, نا *nA* “us”, ك *k*, كما *kmA*, كم

		A, تا *tA*, وا *wA*,	*km*, كن *kn* “you”, ـه *h* “him”, ها *hA* “her”, هما *hmA*, هم *hm*, هن *hn* “them”

Table 6.15: The Arabic truncated affixes

Here are some examples from the final output of the Arabic stemmer after combining both prefix and suffix removal.

Clustered Words	**Gloss**	**Removed Affixes**	**Outputted Stems**
وتولى *wtwlY* فتولى *ftwlY* يتولى *ytwlY* تولى *twlY*	and he turned away then he turned away he turns away he turned away	و *w* ف *f* ي *y* ت *t*	تولى *twlY* تولى *twlY* تولى *twlY* ولى *wlY*
شرح *$rH* اشرح *A$rH* نشرح *n$rH* يشرح *y$rH*	expanded expand we expand he expands	—— ا *A* ن *n* ي *y*	شرح *$rH* شرح *$rH* شرح *$rH* شرح *$rH*
ماكثون *mAkvwn* ماكثين *mAkvyn*	staying (nom.) staying (acc. & gen.)	ون *wn* ين *yn*	ماكث *mAkv* ماكث *mAkv*
مذعنين *m*Enyn*	compliant	——	مذعنين *m*Enyn*

Table 6.16: A sample of the output of the Arabic stemmer

The first example in the table shows that the Qur’anic text contains four variants of the base form *wlY*. The first three words have been stemmed to *twlY*, whereas the fourth variant has been stemmed to *wlY*. This example shows that a particular word is stemmed to some form and then this second form is stemmed to a third one. This is expressed through the following formula.

$$\begin{array}{l} X \longrightarrow X_1 \\ X_1 \longrightarrow X_{11} \end{array} \tag{6.1}$$

In this formula X refers to a given word-form, while X_1 refers to the possible stem for such a form which may be another word-form with X_{11} as its possible stem. We should make it clear that the gloss we provide for the Arabic examples is based on the English translation that we use. Some words are homonymous in nature, i.e. they can have more than one unrelated meaning. Thus, the word شرح *$rH* is translated in the Qur'anic corpus with the meaning of "expand". But it can be used in other contexts to mean "explain" as in شرح الدرس *$rH Aldrs* "(he) explained the lesson". It is noteworthy that the word-form مذعنين *m*Enyn* is the only form that has occurred in the Qur'anic text and so the stemmer kept it as it is because it has no similar forms.

Broadly speaking, according to Larkey et al. (2002; 2007), stemmers make two types of errors. Strong stemmers tend to form larger stem classes in which unrelated forms are erroneously conflated, while weak stemmers fail to conflate related forms that should be grouped together. Most stemmers fall between these two extremes and make both types of errors. As far as the accuracy of our Arabic stemmer is concerned, we are mainly interested, as mentioned earlier, in grouping semantically related words under one reduced form (or stem). This reduced form may be the right stem or not. Thus, we measure the accuracy of clustering related words without taking into account whether the stem is the legitimate form or not. In this regard, the Arabic stemmer has achieved a precision of 0.96 when tested on a set of 100 words. As for recall, it is difficult to measure it because we do not know how many other forms should have been conflated with the current outputted forms. The standard we use to measure the stemmer's accuracy can be illustrated in the following table.

Word-Forms	**Gloss**	**Possible Stems**	**Hypotheses & Scoring**
شاهد *$Ahd*	witness	شاهد *$Ahd*	
شاهدا *$AhdA*	a witness	شاهد *$Ahd*	A-B [1] B-C [1] C-E [1]
شاهدون *$Ahdwn*	witnesses	شاهد *$Ahd*	A-C [1] B-D [1] C-F [1]
شاهدين *$Ahdyn*	witnesses	شاهد *$Ahd*	A-D [1] B-E [1] D-E [1]

وشاهد *w$Ahd* الشاهدين *Al$Ahdyn*	and a witness the witnesses	شاهد *$Ahd* شاهد *$Ahd*	A-E [1] B-F [1] D-F [1] A-F [1] C-D [1] E-F [1]
الشر *Al$r* الشرك *Al$rk* بالشر *bAl$r*	the evil associating others (with Allah) with the evil	الشر *Al$r* الشر *Al$r* الشر *Al$r*	A-B [0] A-C [1] B-C [0]

Table 6.17: Arabic stemmer's accuracy standard

As the previous table shows, we set a number of hypotheses for scoring the relatedness of clustered words. So, the hypothesis A-B, for example, checks whether the first word and the second word are semantically related. If so, they are given [1] score. If they are unrelated they are given [0] score. Accordingly, in the first example all combinations are given [1] score because they are all related. However, in the second example the second word is unrelated to both the first and third words and thus A-B and B-C are given [0] score. The fact that the definite article ال *Al* "the" was kept in the stems of the word-forms in the second example is obvious but is not considered in the scoring as we stated earlier.

6.3.2.3.4 Our Approach to English Stemming

As for English stemming, we apply the same techniques as before with only one difference. We truncate only inflectional suffixes. We do not truncate prefixes as the case with Arabic. Our approach to English stemming can be shown in the following figure.

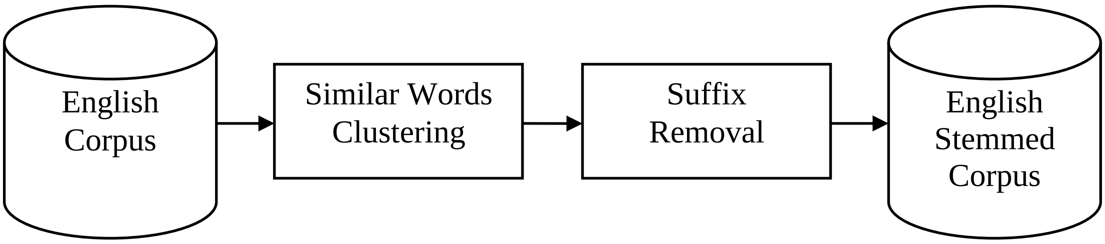


Figure 6.5: Our Approach to English stemming

The suffixes we remove from words are illustrated in the following table.

Plural or present tense	s
Plural or present tense	es
Possessive	‘s
Past tense or participle	ed
Progressive participle	ing
Past participle	en
Comparative Adjective	er
Superlative Adjective	est
A Special Case	e

Table 6.18: The English truncated suffixes

We should refer to something here concerning the suffixes in the above table. We remove the letter “e” from the end of words to capture cases such as “change” so that it can be conflated with other related variants such as “changes”, “changed” and “changing”.

Here are some examples from the English stemmer.

Clustered Words	**Removed suffixes**	**Possible Stems**
messengers	s	messenger
messenger’s	's	messenger
conceal	——	conceal
conceals	s	conceal
concealed	ed	conceal
concealing	ing	conceal
change	e	chang
changed	ed	chang
changing	ing	chang

Table 6.19: A sample of the output of the English stemmer

As noted earlier, the most common algorithm for English stemming is the Porter Stemmer. Since our stemmer, which is corpus-based, and Porter Stemmer, which is rule-based, use different techniques, they are not strictly comparable. As indicated before, the tools that process the English language are not part of the contribution of this thesis. Therefore, we will not evaluate the accuracy of the English stemmer. However, looking at the previous table gives us insight into the performance of the stemmer, which is accurate enough for the current task.

6.3.3 General Bilingual Proposer

As pointed out above, our first experiment towards target word selection, using our general framework, is applied to raw texts that have no linguistic annotations. We use both the Arabic and English stemmers to stem word-forms in the parallel corpus. This allows us to experiment with both stemmed and unstemmed texts. In this experiment translational equivalents are extracted for both content and function words as discussed before. When we come to the section on evaluating the proposer at this stage we will evaluate it with regard to all words and also content words only. We will discuss below our proposed method toward target word selection then talk about the different algorithms we use to achieve our goal.

6.3.3.1 Current Framework

The method underlying the general framework that we adopt to select the translational equivalent of an SL lexical item consists of two stages:

(i) Bilingual Lexicon Building

(ii) Target Word Selection

A similar approach has been carried out on an English-Greek annotated parallel corpus with the use of context vectors (Piperidis et al., 2005). They use only annotated corpora. They have not experimented with raw unannotated corpora. But our method is applied to both raw and annotated corpora. We have done that to show the difference in results when experimenting with different types of text. This method of comparison has not been applied before, to the best of our knowledge, to Arabic-English parallel corpora.

6.3.3.1.1 Bilingual Lexicon Building

The first goal is thus to automatically build a bilingual lexicon. The corpus-specific lexicon is extracted using unsupervised statistical methods, based on the following basic principle:

- For each sentence-pair, each word of the TL sentence is a candidate translation for each word of the parallel SL sentence.

This principle means that (*S*, *T*) is a candidate if *T* appears in the translation of a sentence containing *S*. This sentence-pair may be raw or POS tagged. Following the above principle we compute the frequency (the number of occurrences) of each word in the SL and TL sentences. We then compile a bilingual lexicon, giving preference to the target word that has the highest score in the TL sentences that correspond to the SL sentences. We call this method the 'baseline' algorithm, since we will introduce two other algorithms that are based on the same method but with some modifications, as will be discussed below. However, this procedure of extracting the lexicon considers all candidates for inclusion in the lexicon, and thus results in significantly low precision and recall. This is because the TL function words that occur more frequently in the corpus are suggested as possible translations for any SL word. Therefore, we have to filter the parallel corpus to exclude the function words from being suggested as possible translations for content words. The way of filtering the corpus is explained in the following section. Our automatic lexicon extraction is applied to both raw and POS-tagged texts. Figure (6.6) below depicts an overall view of lexicon building architecture, whether on unannotated or POS-tagged bi-texts.

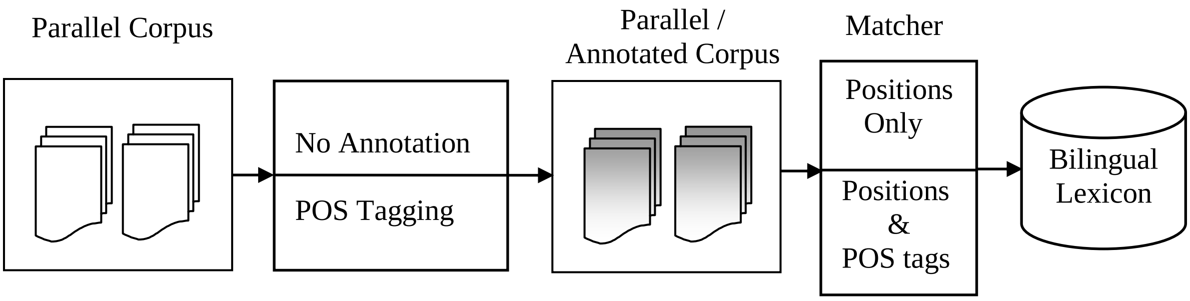


Figure 6.6: Automatic lexicon building architecture

As figure (6.6) shows, we build the bilingual lexicon either from raw unannotated texts or from POS-tagged texts. Then we apply the basic principle that we have mentioned above, which we will call a matcher. This matching algorithm either matches words in the raw parallel texts based on only frequency of occurrence and relative positions (which will be described under the section for 'scoring algorithms') or matches words based on these two notions besides the similarity of their POS tags in both languages.

6.3.3.1.1.1 Parallel Corpus Filtering

In order to filter out the highly frequent words, which are mostly function words, from being suggested as likely translations for every content word we have used the following constraint:

- The proposed TL word should not occur more than *n* times as often as the SL word or less than 1/*n* as often in the entire corpus for some value of *n*.

Normally we will refer to this constraint as 'the filter' in most places in this chapter. But when we want to distinguish it from another sort of filter such as the POS tags filter, we will specify it as 'the occurrence filter'. This distinction will be made when the tagged proposer is discussed later. We tried different values for *n*, i.e. 4, 5, 6... 10, all of which led to the same result as *n* = 3. However, trying it with *n* = 2 gave less accurate results.

This filter is bi-directional. So, it works if the SL is Arabic or English. But we will use the Arabic-English direction throughout the thesis for two reasons:

(i) Arabic-to-English translation is harder than English-to-Arabic translation, because of the translation problems involved on different linguistic levels: lexical, morphological, syntactic, .etc.

(ii) Moreover, Arabic-to-English direction is easier to evaluate. This is because we focus only on content words and ignore the function words which may be free words or clitics attached to other words. In fact, we constructed a Gold Standard for Arabic-to-English translation to compare it with the output of our system. Hence, when we refer to the SL we mean the Arabic language and the TL refers to English.

As regards the filter, when we use it as a constraint for selecting TL translational equivalents, the overall performance is considerably improved. This is best illustrated through the following examples in Table (6.20) below. We will give an example for an SL word that has low frequency in the corpus so that the extracted lexicon can best be accommodated in the table. This example is also given using one of the three scoring algorithms that we will present in the following section, namely the baseline algorithm.

Algorithm	**Filter**	**SL Word**	**Frequency**	**Extracted Lexicon**	**Correct Translations**
Baseline	-	عبس *Ebs*	2	(3, he), (2, **frowned**), (2, and), (1, turned), (1, thereafter), (1, **scowled**), (1, away)	frowned scowled
Baseline	+	عبس *Ebs*	2	(2, **frowned**), (1, **scowled**)	frowned scowled

Table 6.20: An example of the extracted lexicon using the baseline algorithm with and without the filter

We should make it clear that we use the plus sign (+) here to refer to using the filter and the minus sign (-) to refer to doing without the filter. As can be seen in the above table, when the filter is not used the extracted lexicon for the Arabic word *Ebs* contains seven words with the function word "he" scoring the highest number of occurrences and so is most likely to be chosen as the translation for the SL word in the final phase of the system. Moreover, there are other closed-class words that are suggested in the lexicon and are similarly wrong translations for the SL word. But when the filter is used the extracted lexicon contains the correct English word "frowned" and another synonymous word "scowled". The word "frowned" was given a higher score for occurrence, i.e. 2 times, than the other word "scowled" which occurred 1 time.

6.3.3.1.1.2 Scoring Algorithms

We have used three different algorithms to match Arabic words with their English equivalents. These three algorithms result in different scores that will be discussed in the evaluation sections. Thus, we will call them scoring algorithms. The first algorithm is the **baseline**. This algorithm embodies the general principle that we have mentioned above about our proposed method for learning bilingual equivalents. As a reminder, we have stated that according to this principle all words in a TL sentence (or verse) are considered as possible candidates for each word in the corresponding SL sentence. In this algorithm we do not take any other factors into consideration, such as the relative distance between words in a sentence that we will consider in the other scoring algorithms. In this way this baseline algorithm will extract all the TL words that have occurred in correspondence with each SL word in the entire parallel corpus and add them to the extracted lexicon. Therefore, this algorithm is expected to result in low accuracy of the extracted lexicon. Furthermore, the situation is made worse by the difference between both Arabic and English with regard to morphological nature and sentence structure.

In fact, it is commonly the case that one Arabic word may correspond to a number of English words, as has been shown in chapter 1. In this case all the English words are suggested as likely candidates for the Arabic word. Let us consider the following example.

6.6 فلنذيقن

*fln*yqn*

So indeed we cause to taste definitely (lit. trans.)

So indeed we will definitely cause (….) to taste (Qur'an, 41: 27)

The empty parenthesis used in the translation is a placeholder for the object of the whole sentence. It is clear from this example that the Arabic word corresponds to 8 English words. This is because of the rich morphological nature of Arabic where clitics are attached to words. Those clitics are translated as separate words in English. As a matter of fact, this problem could have been solved by tokenization, i.e. segmenting clitics from stems. But we could not do this because our approach is lexicon-free and it is difficult to tokenize words without using a lexicon of words. Under this algorithm the 8 English words will be proposed as possible translational candidates for the Arabic word in the bilingual lexicon. In this case this baseline algorithm is expected, as noted above, to result in low accuracy in the bilingual lexicon. However, when the filter constraint is used, the accuracy is improved.

The structural difference between Arabic and English is also another factor that aggravates the situation for matching Arabic words with their corresponding English words in the parallel corpus. We have emphasized in the preceding chapter that Arabic, unlike English, is characterized by word order variation. We have shown that one of the basic sentence structures in Arabic is the verbal sentence, where a sentence begins with a verb followed by a subject, object and other complements. This structure is more common in Arabic, especially in the CA corpus that we are using in this study. This verbal structure is usually described as the canonical word order in Arabic. The following figure shows this difference in word order between both languages through alignment of a short verse in the corpus (Qur'an, 29:1).

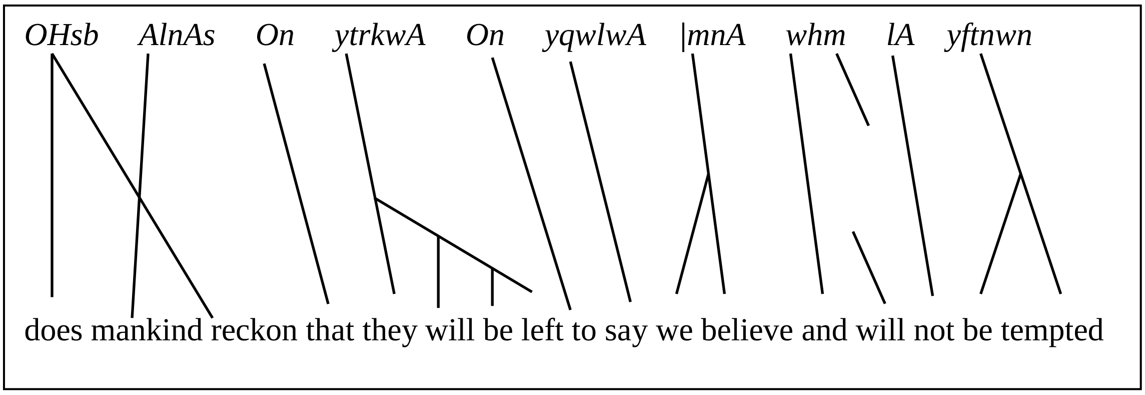

OHsb AlnAs On ytrkwA On yqwlwA |mnA whm lA yftnwn
does mankind reckon that they will be left to say we believe and will not be tempted

Figure 6.7: Alignment of a short verse showing word order differences between Arabic and English

As we can see in the previous figure the correspondences between Arabic and English words are different. Thus, the word order in the Arabic sentence and the English translation is different. Moreover, a single Arabic word may correspond to one or more English words. Furthermore, there may be an English word that has no equivalent in the Arabic sentence or an Arabic word that has no corresponding translation in the sentence.

The two other algorithms are based on the positions of SL words in a verse relative to the positions of corresponding TL words. The second algorithm is called **weighted 1**. In this algorithm the distance between the relative positions of the SL and TL words is taken into account. This can be illustrated in the following figure

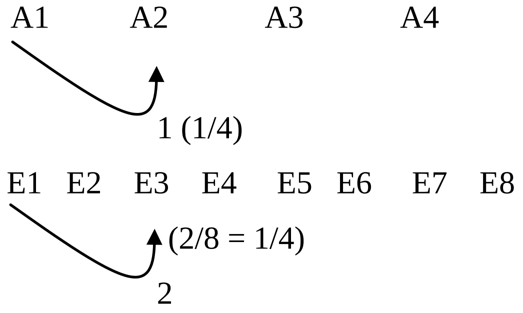


Figure 6.8: Distance between the relative positions of words

The previous figure shows that the relative distance between word A1 and word A2 is 1 compared to a total of 4 words and thus it constitutes ¼ of the total. But the relative distance between word E1 and E3 is 2 compared to a total of 8 words and similarly constitutes ¼ of the total. Thus, E3 is a likely equivalent for A2. This can be illustrated through giving the following example.

6.7 سبح اسم ربك الأعلى

sbH Asm rbk AlOElY

Glorify name lord-your the-most high (lit. trans.).

Extol the name of your Lord, the Most Exalted (Qur’an, 87: 1).

Disregarding the comma in the English translation, we have 9 words as equivalents to 4 Arabic words. The algorithm in question is expected to measure positional distance and thus make a rough alignment as in the following figure.

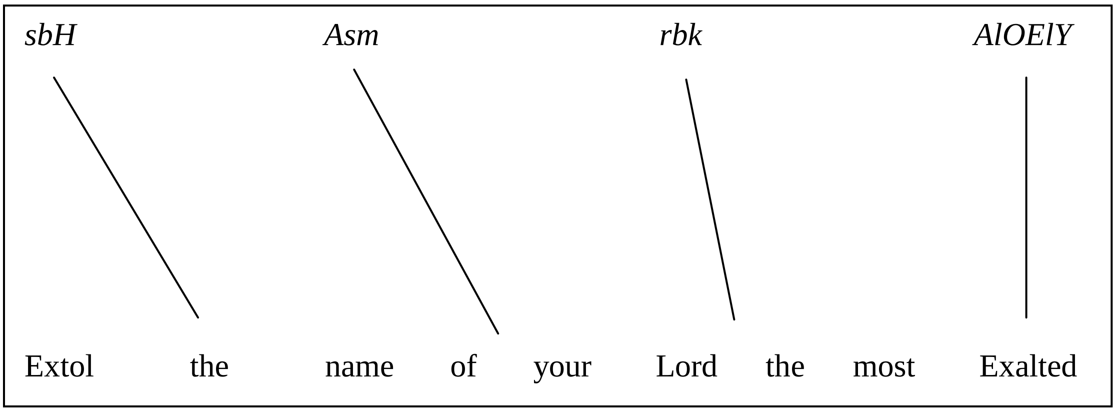


Figure 6.9: Positional distance based on weighted 1 algorithm

We can notice in the above figure that the final word in the Arabic sentence is aligned with the final word in the English sentence. Then the penultimate word occupies a position in the Arabic sentence relative to the position between the English words *Lord* and *the*. This is measured by dividing the number of the position of a word by the total number of words in a sentence. Thus, the Arabic word *rbk* comes in the third position. So, it is 3rd out of total 4. This position is relative to the position between the two English words *Lord* and *the* which fall in the 6th and 7th positions out of total 9. Hence, one of the two words, either *Lord* or *the* should be aligned with the Arabic word in question. The same principle applies to the other words. This is expected to lead to low accuracy of the extracted lexicon. However, when we use the filter, all function words in the above English sentence are excluded and we are left with four English words in correspondence with four Arabic words and thus the alignment is improved, leading to improvement of accuracy in the lexicon.

The third algorithm is **weighted 2** where the distance between the relative positions of SL and TL words is measured as in weighted 1 above then multiplied by itself. Thus, if a word is in the second position in a two-word sentence, it thus constitutes ½ out of the total according to weighted 1 algorithm, but ¼ according to weighted 2 algorithm. The difference between weighted 1 and weighted 2 scoring algorithms can be shown in the following figure:

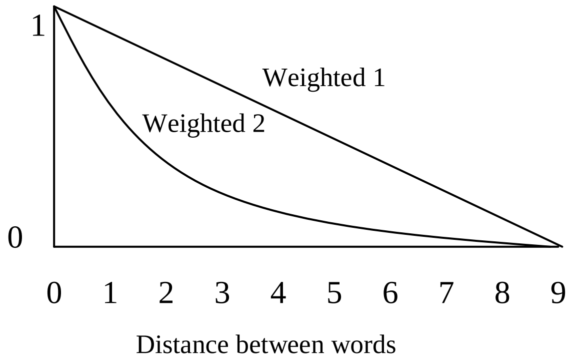


Figure 6.10: Difference between weighted 1 and weighted 2 algorithms

In figure (6.10) the curve shows that the distance between words decreases when we apply weighted 2 algorithm, whereas in weighted 1 algorithm the line is straight to show that the distance between words increases. The figure signifies that when an SL word and a TL word are far away we pay less attention to them in our matching. We only pay attention to those words that are nearer each other in their positions in a parallel sentence.

6.3.3.1.2 Target Word Selection in Context

Having extracted the bilingual lexicons from the parallel corpus using different scoring algorithms with or without stemming, we now proceed to select the TL word for a given SL word in their contextual verses. We achieve this task by carrying out two steps in order.

1- We go through the extracted lexicon in which every SL word has a number of corresponding TL words. These TL words are listed in a descending order according to their frequency of occurrence in the context of the SL word in question. It is worth mentioning that sometimes the extracted lexicon for a given SL word may be empty without any corresponding TL words. We then pick up the first TL word in the lexicon that has the highest frequency of occurrence as a possible candidate for the current SL word. This can be illustrated in the following figure.

```
{AlmflHwn: [(11, 'prosperers'), (2, 'written'), (2,
'weigh'), (2, 'scales'), (2, 'protected'), (2, 'party'),
(2, 'maleficence'), (2, 'heavy'), (2, 'beneficence'), (1,
'wicked'), (1, 'weight'), (1, 'wayfarer'), (1, 'though'),
(1, 'tawrah'), (1, 'striven'), (1, 'spirit'), (1,
'shackles'), (1, 'residence'), (1, 'making'), (1,
'location'), (1, 'kinsmen'), (1, 'kinsman'), (1, 'injil'),
(1, 'indigent'), (1, 'having'), (1, 'forbidding'), (1,
'forbid'), (1, 'ear'), (1, 'brothers'), (1, 'breasts'), (1,
'aided')]}
```

Figure 6.11: An example showing the order of TL words in the lexicon according to their frequency of occurrence

The previous figure shows that the word المفلحون *AlmflHwn* “the prosperers/successful” has a number of corresponding TL words in the extracted lexicon. These TL words are listed in descending order in a tuple comprising the number of occurrences and the TL word. This example is from a bilingual lexicon extracted using the baseline scoring algorithm with the filter on raw unstemmed texts. That is why no function word has shown up in the lexicon. It is observable that the first word among the TL words, namely “prosperers”, has the highest score of frequency and thus will be chosen to be the likely translational equivalent for the current SL word. In actual fact, the SL word المفلحون *AlmflHwn* includes the definite article which should be translated as “the prosperers” as found in the English corpus that we are using. However, as pointed out throughout the thesis, we are interested in the open-class words and not the closed-class words. Thus, we regard this translation as correct.

2- The previous step can be called ‘learning bilingual equivalents’. We have given an illustrative figure (1.2) that shows the mechanism for the learning process. This second step is what we call ‘the application phase’ and for which we have also provided an illustrative figure (1.3). In this phase we use the Arabic text of the parallel corpus and the extracted bilingual lexicon to select the translation for every Arabic word in its contextual verse. This following table shows the translation of open-class lexical items in the context of their verses using those lexicons that were extracted applying the baseline scoring algorithm with the filter on raw text. We give the results for both unstemmed and stemmed texts. The SL refers to Arabic and the TL refers to English.

SL Words	**Suggested Equivalents**			
	Unstemming SL and TL	Stemming SL and TL	Stemming SL only	Stemming TL only
الكتاب *AlktAb*	book	******	******	book
ريب *ryb*	suspicion	suspicion	suspicion	suspicion
هدى *hdY*	guidance	guidance	guidance	guidance
للمتقين *llmtqyn*	admonition	******	******	admonition

Table 6.21: Selection of equivalents using the baseline algorithm with the filter on raw texts

We can observe in the previous table that the results for the translation of the above words are different depending on whether the SL and TL texts are stemmed or not. Thus, some words have been translated into English and some others were left without any translation at all. The SL word, الكتاب *AlktAb* “the book”, for instance, is translated as “book” which we consider right after ignoring the definite article “the” when the SL and TL are kept as they are without stemming or when the TL only is stemmed. But when both the SL and TL are stemmed or when the SL only is stemmed the SL word has no translational equivalent. This lack of equivalence occurs under the current algorithm and the filter constraint but may be obtained under other algorithms. The first three words are translated correctly under certain types of text, whereas the final word is either translated wrongly or has no equivalent at all.

Following the presentation of the general proposer and the algorithms used we now move on to discuss the evaluation of the extracted lexicons and the selection of translational equivalents in their context. Since we deal with both raw texts and annotated texts in our general framework we subdivide the proposer into three types according to the kind of text we are handling. These types we will call **raw proposer** for raw texts, **tagged proposer** for POS-tagged texts and **parsed proposer** for DR-labelled texts, which we use to obtain ‘head-dependent’ translation pairs. Both raw and annotated texts can be stemmed or not. We will start with shedding light on the evaluation of the raw proposer then go on to discuss the other types later.

6.3.4 Bilingual Proposer for Raw Texts

As pointed out above, we will test our general proposer on raw unannotated texts as well as texts annotated with POS tags and DRs. In section 6.3.2.3 we have presented a stemmer for Arabic and English. We use the stemmer with the general proposer in our experiments to select lexical equivalents. Thus, we have tested the general proposer on both stemmed and unstemmed Arabic and English texts. This method of testing on both stemmed and unstemmed texts will be applied to all types of texts, i.e. raw, POS-tagged, and DR-labelled texts. Notably, we have referred earlier to using a filter to exclude function words from being selected as candidate translations for open-class words. Moreover, we have also described three different scoring algorithms that we use to extract the lexical equivalents. Combining all these constraints, i.e. stemming, filtering and scoring algorithms, leads to 24 different outputs for every type of proposer. This is because we have three scoring algorithms and a filter that can be used or not. So, 3 * 2 = 6. Then, both Arabic and English texts can be both stemmed or one of them. So, we have four different combinations. The final outcome is thus 6 * 4 = 24. As noted above, we have basically three types of proposer: raw proposer, tagged proposer and parsed proposer. So, every type of these three proposers has 24 different outputs. The algorithm used to extract the equivalents in raw texts is the same algorithm discussed above under the general proposer heading. We only change it in case of tagged and parsed proposers. This is because these two proposers deal with different types of texts, i.e. linguistically annotated texts whether annotated with POS tags or DRs. In the following lines we will discuss the evaluation of the raw proposer with regard to two main points: bilingual lexicon extraction and target word selection in context.

6.3.4.1 Evaluation

As pointed out above, the first point we will evaluate in this section is the accuracy of the bilingual lexicons that are extracted by applying the different constraints mentioned above. We have clarified that after applying these constraints we end up with 24 different outputs, which are the extracted lexicons. The second point concerns the evaluation of the translation module that selects the lexical equivalents in their contexts. First, we will describe the measures that we have used to evaluate both the extracted lexicons and the translation pairs in their contexts. Then, we will discuss the scores we have obtained for both lexicons and translational equivalents.

It goes without saying that manual evaluation of MT output is informative but is also time-consuming, expensive and not reusable. Automatic evaluation, on the other hand, has a number of advantages: it is quick, cheap, language-independent and used for large-scale evaluation. Moreover, it can be applied repeatedly to the MT output during system development to assess any changes made without incurring any extra cost. However, this automatic evaluation should also correlate highly with human judgements. Accordingly, developing and validating automatic MT evaluation metrics has proved challenging (Hearne, 2005). Different metrics for evaluating MT output have been proposed recently. Among the most known metrics are BLEU

(Papineni et al., 2002), NIST[17] (Doddington, 2002) and F-Measure (Melamed et al., 2003; Turian et al., 2003). Broadly speaking, all these metrics involve comparing candidate translations outputted by an MT system with their reference translations. But they are different with respect to two main criteria.

(i) How they measure the similarity between candidate and reference translations.

(ii) How they reward the similarities and penalize the differences between those translations.

NIST (2002) observes that automatic scoring is mostly reliable when reference translations are of high quality and the input sentences are from within the same genre. As our Arabic and English data comprise the religious text of the Qur’an translated by a professional translator, we believe that our experiments are particularly well suited to evaluation using automatic metrics. We thus use an automatic evaluation metric to judge the accuracy of the proposer on both raw and annotated texts. In general terms, both BLEU (Bilingual Evaluation Understudy) and NIST metrics evaluate an MT system quality by comparing output translations to their reference translations in terms of the number of co-occurring *n*-grams: “the closer a machine translation is to a professional human translation, the better it is” (Papineni et al., 2002.) The basic unit of evaluation in both metrics is the sentence, which is outside the scope of our current study. We have pointed out throughout the thesis that we are dealing only with the translation of lexical items not complete sentences. Therefore, we consider that both measures are not suitable for our task. The standard F-measure, on the other hand, is a suitable measure for our task. F-measure is a well-known measure for evaluation, which considers both the *precision* and the *recall* of the test to compute the score. The F-measure can be defined as follows:

[17] Its name comes from the US National Institute of Standards and Technology.

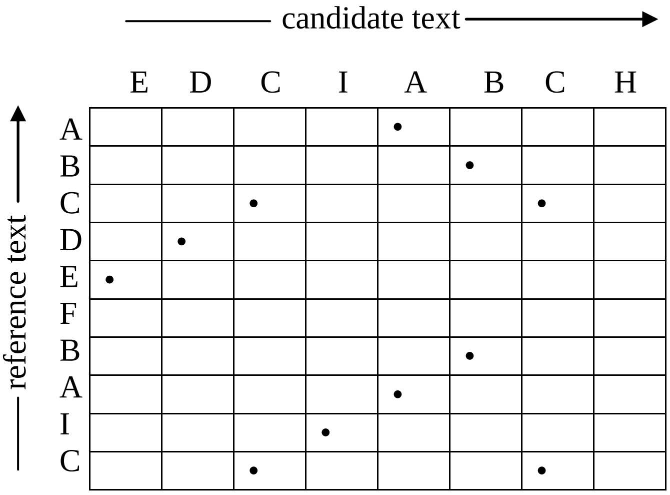
$$F = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{6.2}$$

Thus, the two parameters used to compute F-score are precision and recall. Following Melamed et al. (2003) and Turian et al. (2003), precision and recall scores for candidate item Y with respect to reference item X are calculated according to equations (6.3) and (6.4) respectively.

$$\text{precision}\ (Y|X) = \frac{|X \cap Y|}{|Y|} \tag{6.3}$$

$$\text{recall}\ (Y|X) = \frac{|X \cap Y|}{|X|} \tag{6.4}$$

In the field of MT evaluation, precision can be simply defined as the number of correct translations outputted by an MT system out of the total number of the output. Recall, on the other hand, is defined as the number of correct translations outputted by such a system out of the total number of the words that should have been translated. The intersection between a candidate translation and a reference translation is given a score in favour of the MT output. This intersection is best illustrated by Melamed et al. (2003) and Turian et al. (2003) through what they refer to as a bitext grid, which is shown in figure (6.12) below.



candidate text
reference text

	E	D	C	I	A	B	C	H
A					•			
B						•		
C			•				•	
D		•						
E	•							
F								
B						•		
A					•			
I				•				
C			•				•	

Figure 6.12: Bitext grid illustrating the relationship between an example candidate translation and its corresponding reference translation - the words of the reference translation are shown from top to bottom down the left-hand side of the grid and the words of the candidate translation are shown from left to right across the top of the grid. Each bullet, called a hit, indicates a word contained in both the candidate and reference strings. (This figure is adapted from Figure 1 of Melamed et al. (2003) and Turian et al. (2003).)

As a matter of fact, MT systems have been known to generate more words than appear in a reference text. This is because there is a multitude of ways to express any given concept in natural language (Turian et al., 2003). Furthermore, as shown throughout the thesis, Arabic words can have a number of clitics that are often translated into separate words in English. But since we do not care about the clitics as described above, or the inflections of verbs, we have to put a number of possible translations for a given SL string in our reference translations (the Gold Standard). Another point that needs to be mentioned relates to the translation of an SL lexical item with an MWE in English. That an SL word can be translated with a number of lexical items in a TL is not uncommon. This phenomenon is very common in the parallel corpus that we use, because there are a lot of Arabic lexical items in the Qur'an that cannot be translated word for word, but need to be conveyed into English through the use of MWEs. The proposer deals only with single words and cannot tackle MWEs. Thus, it selects one word of the whole MWE. Only the parsed proposer selects ‘head-dependent’ translation pairs, which we use to carry out the bootstrapping techniques. When it comes to scoring the proposer, we give a half score to an MWE that is reduced to only one word if the word selected is the meaning-bearing word. Otherwise, we regard the translation as wrong. Table (6.22) below throws light on the Gold Standard we use concerning the first three points (i.e. possible multiple translations, clitics and verb inflections). As for the fourth point that is related to MWEs, we will illustrate it through a separate table.

SL String	**Gold Standard**	**Comment**
ريب *ryb*	suspicion doubt	An SL string can have a number of TL equivalents that are mostly synonymous.
الصلاة *AlSlAp*	the prayer prayer	We ignore the clitics and focus on the meaning-bearing words. For instance, we do not care about the definite article, as this example shows.
يؤمنون *yWmnwn*	believe believes believing	We give different inflectional forms of a verb as candidate translations because we do not pay attention to such differences.

Table 6.22: A sample of the Gold Standard for some words

The following table shows how we deal with MWEs with regard to the evaluation of the proposer’s output for both lexicons and selection of equivalents in context.

SL String	**Gold Standard**	**Accepted with a Half Score**	**Totally Wrong**
ويقيمون *wyqymwn*	and keep up	keep	up

	keep up establish		
يخادعون *yxAdEwn*	try to deceive	deceive deceiving	try to
الصالحات *AlSAlHAt*	deeds of righteousness	righteousness	deeds of
يذكر *y*kr*	constantly remember	remember	constantly
نطفة *nTfp*	a sperm drop	sperm	drop

Table 6.23: A sample of the Gold Standard for some MWEs

In the previous table, the SL items are translated as MWEs. When the proposer selects the word that has the main meaning in an MWE, we give it a half score. However, when it selects any of the other words that do not contribute to the overall meaning of the MWE, we consider it totally wrong and so it is given a [0] score.

6.3.4.1.1 Bilingual Lexicons

An extracted bilingual lexicon could contain a number of TL translation candidates for a given SL word. These candidates come in order of frequency, and the correct equivalent may occupy any position in the list. The suggested TL words vary in number according to the used algorithm. However, in some cases no translation candidate is suggested. We use the first 100 words in an extracted lexicon as a test set to evaluate its accuracy. As pointed out above, we use the F-measure to evaluate the accuracy of the extracted bilingual lexicons. Nevertheless, the F-measure is used to evaluate a given lexicon based on the first suggested TL word, which may be the correct equivalent or not. Indeed, the correct equivalent of an SL word may come in the second, third, or any other position in the lexicon. We will apply the **F-measure** to test only the words that come in the first position in all the lexicons. But we will apply another evaluation measure to show how often a correct equivalent is suggested in the first 10 positions in the lexicon that gets the best F-measure. This measure is the **Confidence-Weighted Score** (CWS), which will be discussed below. Sometimes a lexicon could contain a big number of suggested TL words, especially when the filter is not used. In this case the correct equivalent may come after the 10th position. However, investigation of results has shown that any lexicon in which the correct equivalent comes after the 10th position is not useful anyway.

All the results we have obtained on various types of raw text using different algorithms are shown in the following table. We use some abbreviations in the tables throughout this chapter. The letter C stands for the canonical (or stemmed) form, AV stands for the Arabic verses, while EV stands for the English verses. The (+) sign refers to the use of the filter, while the (-) sign refers to the absence of it. The letters B and W are sometimes used to refer to Baseline and Weighted algorithms respectively.

Verses	**Scoring Algorithm**	**Filter**	**Precision**	**Recall**	**F-score**
AV & EV	Baseline	–	0.02	0.02	0.02
		+	0.2682	0.22	0.2417
	Weighted 1	–	0.04	0.04	0.04
		+	0.3536	0.29	0.3186
	Weighted 2	–	0.06	0.06	0.06
		+	0.3536	0.29	0.3186
CAV & CEV	Baseline	–	0.01	0.01	0.01
		+	0.3375	0.27	0.3
	Weighted 1	–	0.075	0.075	0.075
		+	0.41875	0.335	0.3722
	Weighted 2	–	0.075	0.075	0.075
		+	0.41875	0.335	0.3722
CAV & EV	Baseline	–	0.01	0.01	0.01
		+	0.3833	0.345	0.3631
	Weighted 1	–	0.07	0.07	0.07
		+	**0.5166**	**0.465**	**0.4894**
	Weighted 2	–	0.07	0.07	0.07
		+	**0.5166**	**0.465**	**0.4894**
AV & CEV	Baseline	–	0.02	0.02	0.02
		+	0.1780	0.13	0.1502
	Weighted 1	–	0.04	0.04	0.04

		+	0.2191	0.16	0.1849
	Weighted 2	–	0.07	0.07	0.07
		+	0.2191	0.16	0.1849

Table 6.24: F-scores for the extracted lexicons using raw texts

We can observe in the table that the scoring algorithm ‘weighted 2’ has achieved the same score as the scoring algorithm ‘weighted 1’ as far as lexicons are concerned. In addition, we can notice that using the filter has improved the accuracy in all types of text. It is noticeable that using a stemmed Arabic text against an unstemmed English text has achieved a better score than all other combinations. Thus, the lexicon that has achieved the highest F-score is the one extracted using Arabic stemmed text and English unstemmed text, applying the filter with either weighted 1 or weighted 2 algorithms. It has been indicated that we have evaluated the bilingual lexicons based on the idea that the correct equivalent is suggested as the first word in the lexicon. However, as pointed out above, sometimes the correct equivalent comes in the 2nd, 3rd or any other position in the lexicon. A second measure, as noted above, will be used in this regard to evaluate the lexicon. This measure is CWS (also known as Average Precision) which, according to Dagan et al. (2006), indicates that judgements of the test examples are sorted by their confidence (in decreasing order). They illustrate the CWS by the following equation:

$$CWS = \frac{1}{n}\sum_{i=1}^{r}\frac{\# correct - at - rank - i}{i} \tag{6.5}$$

As far as our task is concerned, *n* in this equation refers to the number of the SL words in the test set, *i* ranges over ranks of the sorted translation candidates, and *r* is the maximum rank considered. The CWS has been measured for the lexicon that has achieved the best F-score, namely CAVEVW1+ or CAVEVW2+ and the result is shown in the following table.

Ranks	**Correct Answers**
1	46.5
2	8
3	2
4	1
5	1
6	0
7	3
8	1
9	1
10	0
CWS	**0.5228**

Table 6.25: Confidence-Weighted Score for a bilingual lexicon regarding the first 10 positions using raw texts

We calculate CWS for the first 10 words in the lexicon as follows. The correct answers are divided by their rank then added together and then divided by the total number in the test set, which is 100 words. The calculation is done as follows:

(6.6)

$$\frac{\frac{46.5}{1}+\frac{8}{2}+\frac{2}{3}+\frac{1}{4}+\frac{1}{5}+\frac{0}{6}+\frac{3}{7}+\frac{1}{8}+\frac{1}{9}+\frac{0}{10}}{100}$$

A sample of this lexicon is given below.

SL Lexical Item	**Suggested TL Words**	**Reference Translation**	**Comments**
طلوع *TlwE*	(1) various (2) extremes	rising	No correct equivalent is suggested.
وشيبة *w$ybp*	**(1) hoariness**	and hoariness	We consider here that the equivalent in the lexicon is correct, despite the fact that the conjunction has

			not been translated.
مدبر *mdbr*	**(1) withdrawing** (2) steps (3) staff (4) deaf (5) turning (6) strait (7) safeguard (8) spacious (9) idols (10) availed	withdrawing	The correct translation is suggested as the first word among the TL words.
الأتقى *AlOtqY*	******	the most pious	No equivalent is suggested for the SL word.
ناصبة *nASbp*	**(1) toiling** (2) laboring	toiling	The correct translation is suggested along with another incorrect one.
نمكن *nmkn*	(1) generation **(2) establish** (3) snatched (4) plentifully (5) sanctuary (6) Haman	(we) establish	The correct equivalent occupies the second place in the lexicon.
تلفح *tlfH*	**(1) searing** (2) glumly (3) glowering	is searing	We regard the first word as the right translation, since we ignore auxiliary verbs in English.

Table 6.26: A sample of the bilingual lexicon using weighted 2 algorithm with filtering on stemmed Arabic and unstemmed English

We can notice that some SL words have a number of translation candidates in the lexicon, while some others have one or no candidates at all. Thus, the word طلوع *TlwE* “rising”, for instance, has two equivalents in the lexicon but no one of them is the right translation of the word and thus is given [0] score in the F-score. The word وشيبة *w$ybp* “and hoariness”, on the other hand, has been given [1] score because the correct TL word comes in the first position. Notably, it is also the only suggested word in the lexicon. We have indicated earlier that we ignore the clitics that are attached to open-class words such as the conjunction in this example. As for the word مدبر *mdbr* “withdrawing”, all the 10 words are suggested as equivalents but only the first one is the correct equivalent. So, it has scored [1] in F-score. It is noticeable that no equivalent has been suggested for the word الأتقى *AlOtqY* “the most pious”. This action of ‘no answer’ will definitely decrease the recall but increase the precision. As for the word ناصبة *nASbp* “toiling”, it has two TL candidates which include the correct translation. Therefore, it is given [1] score in the F-score. We can observe that the word نمكن *nmkn* “establish” has a number of TL candidates in the lexicon, of which the second one is the right equivalent. So, it scored [0] in the current F-score evaluation. Finally, the word تلفح *tlfH* is used in the present continuous in the reference translation as “is searing”. But since we focus on the content words we ignore the auxiliary verb and consider the word “searing” as a correct equivalent for the SL word.

6.3.4.1.2 Target Word Selection in Context

Having evaluated the extracted bilingual lexicons, we proceed to evaluate the selection of equivalents in their context. In other words, we evaluate the accuracy of the raw proposer with regard to choosing the correct TL words in their sentential context. We should make it clear that we are mainly concerned with the translation of lexical items not a whole phrase or sentence. In order to have a valid evaluation of the proposer we have tested three different samples from different parts of the corpus. The evaluation of these samples will be described below.

6.3.4.1.2.1 Tested Samples

We have tested three different samples of our corpus to validate our results. Each sample consists of 100 words. There are no specific criteria for our choice of these three samples. We have chosen the three samples from different parts of the corpus. In fact, our selection exhibits difference in the length of verses in these samples. We were keen on observing the performance of the proposer on both long and short verses because we do not use alignment techniques. The first two samples consist of long verses, whereas the third one contains short verses. The first sample contains the first part of سورة البقرة Surat *Al-Baqarah* “the Cow”. The second sample consists of the first words in سورة الكهف Surat *Al-Kahf* “the Cave”. As for the final sample, it is composed of the first words in سورة عبس Surat *Abasa* “he frowned”. The scores of the three samples are given in order below. Firstly, we will test the raw proposer on all words (i.e. both open-class and closed-class words) in each sample. Secondly, we will test it on the open-class words only. We should recall that the raw proposer is applied to raw, unannotated texts. Thus, it cannot distinguish between closed-class and open-class words. But it produces the output for all words in the parallel texts. Consequently, we had to manually remove the closed-class words from the proposer's output when we do the testing on only open-class words.

I- First Sample

Verses	**Scoring Algorithm**	**Filter**	**Precision**	**Recall**	**F-score**
AV & EV	Baseline	_	0.0	0.0	F not available
		+	0.5054	0.465	0.4843
	Weighted 1	_	0.01	0.01	0.01
		+	0.5380	0.495	0.5156
	Weighted 2	_	0.03	0.03	0.03
		+	**0.5597**	**0.515**	**0.5364**
CAV & CEV	Baseline	_	0.0	0.0	F not available
		+	0.4767	0.205	0.2867
	Weighted 1	_	0.0	0.0	F not available
		+	0.5465	0.235	0.3286
	Weighted 2	_	0.0	0.0	F not available
		+	0.5465	0.235	0.3286
CAV & EV	Baseline	_	0.0	0.0	F not available
		+	0.4767	0.205	0.2867
	Weighted 1	_	0.0	0.0	F not available
		+	0.5465	0.235	0.3286
	Weighted 2	_	0.0	0.0	F not available
		+	0.5465	0.235	0.3286
AV & CEV	Baseline	_	0.0	0.0	F not available
		+	0.4402	0.405	0.4218
	Weighted 1	_	0.01	0.01	0.01
		+	0.4728	0.435	0.4531

	Weighted 2	-	0.05	0.05	0.05
		+	0.4945	0.455	0.4739

Table 6.27: Raw proposer's scores on all words in the first sample

In the previous table we have mentioned the scores for all the algorithms with different types of text. From now on we will mention only the algorithms that have achieved the best scores and leave out the algorithms that obtain lower scores. Thus, all unfiltered algorithms will be removed from the rest of the evaluation tables. We will mention only the scores obtained when the filter is used. Also, we will focus on 'weighted 2' because it is the algorithm that achieved the best scores but will mention 'weighted 1' and 'baseline' if they are not similar to 'weighted 2'.

Now we will give the scores obtained when the raw proposer was tested on only the open-class words. This is shown in table (6.28) below.

Verses	Scoring Algorithm	Precision	Recall	F-score
AV & EV	Weighted 1	0.6770	0.5803	0.625
	Weighted 2	**0.6979**	**0.5982**	**0.6442**
CAV & CEV	Weighted 2	0.6785	0.1696	0.2714
CAV & EV	Weighted 2	0.6785	0.1696	0.2714
AV & CEV	Weighted 1	0.5520	0.4732	0.5096
	Weighted 2	0.5729	0.4910	0.5288

Table 6.28: Raw proposer's scores on only open-class words in the first sample

There are general observations that can be made about the tests conducted on all words or only open-class words in the first sample. These observations can be summarized as follows:

(i) The raw proposer has obtained higher scores when tested on only open-class words than when tested on all words. This can be partly due to the fact that a function word is used with different senses and thus has different corresponding translations. Moreover, there are a lot of function clitics in

Arabic that are translated into independent words in English, which explains the reason for the less accurate translation of function words.

(ii) Not using the filter has resulted in 0.0 score in most algorithms, whether tested on all words or open-class only. Thus, doing without the filter results in very low scores in this stage of the proposer.

(iii) Focusing on the filtered algorithms only in all the following observations, we can notice that the best F-score was obtained on the unstemmed Arabic and English texts in tests both on all words and on open-class only, with slight differences between the three different scoring algorithms (i.e. baseline, weighted 1, and weighted 2). It should be noted that stemming Arabic resulted in a high score of precision, but a low score of recall for testing on all words or only open-class words.

(iv) The scores obtained with regard to target word selection in context are different from those scores obtained on the extracted lexicons above. Thus, in case of the contextual translations we find that the best scores were achieved on unstemming Arabic and English. But in case of the extracted lexicons the best score obtained was on stemming Arabic only. This may be due to the fact that we score the lexicon’s accuracy for the first 100 words which have been found to be mostly uncommon. The stemmer has performed well with those uncommon words but not with the common words. This may be due to the fact that common words have a number of different inflected forms, and the stemmer may not be able to conflate all of them.

II- Second Sample

Here is another tested sample. We will start by giving the results we have obtained on all words, and then the results obtained on open-class only.

Verses	**Scoring Algorithm**	**Precision**	**Recall**	**F-score**
AV & EV	Baseline	0.3548	0.33	0.3419
	Weighted 2	**0.4193**	**0.39**	**0.4041**
CAV & CEV	Baseline	0.2209	0.095	0.1328
	Weighted 2	0.3604	0.155	0.2167

CAV & EV	Baseline	0.2209	0.095	0.1328
	Weighted 2	0.3604	0.155	0.2167
AV & CEV	Baseline	0.2903	0.27	0.2797
	Weighted 2	0.3548	0.33	0.3419

Table 6.29: Raw proposer's scores on all words in the second sample

Now we will give the scores that have been obtained when the raw proposer was tested on only the open-class words. This is shown in table (6.30) below.

Verses	Scoring Algorithm	Precision	Recall	F-score
AV & EV	**Weighted 2**	**0.4905**	**0.4406**	**0.4642**
CAV & CEV	Weighted 2	0.4642	0.1101	0.1780
CAV & EV	Weighted 2	0.4642	0.1101	0.1780
AV & CEV	Weighted 2	0.3773	0.3389	0.3571

Table 6.30: Raw proposer's scores on only open-class words in the second sample

The observations outlined above with regard to the first sample are the same for the second sample, with two more things to note here.

(i) The proposer's scores concerning the first sample are higher than the scores obtained in the second sample. We will discuss the possible reasons for the difference in scores between the three samples in the following section.

(ii) Both weighted 1 and weighted 2 algorithms obtain the same scores in all types of text when tested on all words or open-class only. Remarkably, baseline algorithm obtains the same score as weighted 1 and weighted 2 in the test on open-class words only, but gets lower scores in the test on all words. That is why we have removed both baseline and weighted 1 algorithms from the previous table.

III- Third Sample

Following are the scores achieved by the proposer on the third and final sample. The scores for testing on all words are given first in the following table. Then the scores for testing on only open-class words are given in another table.

Verses	Scoring Algorithm	Precision	Recall	F-score
AV & EV	Baseline	0.3734	0.31	0.3387
	Weighted 2	**0.4216**	**0.35**	**0.3825**
CAV & CEV	Baseline	0.4285	0.21	0.2818
	Weighted 2	0.5102	0.25	0.3355
CAV & EV	Baseline	0.44	0.22	0.2933
	Weighted 2	0.52	0.26	0.3466
AV & CEV	Baseline	0.3414	0.28	0.3076
	Weighted 2	0.3902	0.32	0.3516

Table 6.31: Raw proposer's scores on all words in the third sample

The scores for open-class words are given in table (6.32) below.

Verses	Scoring Algorithm	Precision	Recall	F-score
AV & EV	Baseline	0.4791	0.3593	0.4107
	Weighted 2	**0.5208**	**0.3906**	**0.4464**
CAV & CEV	Baseline	0.65	0.2031	0.3095
	Weighted 2	0.75	0.2343	0.3571
CAV & EV	Baseline	0.6666	0.2187	0.3294
	Weighted 2	0.7619	0.25	0.3764
AV & CEV	Baseline	0.4255	0.3125	0.3603
	Weighted 2	0.4680	0.3437	0.3963

Table 6.32: Raw proposer's scores on only open-class words in the third sample

First, there are general observations that we will make here about this third sample. Then we will discuss some observations regarding the difference between the three samples in general.

As for this sample, there are some observations that are summed up as follows:

(i) Not using the filter when testing on this sample has resulted in a higher score than the first and second samples. Thus, while the unfiltered algorithms did not exceed an F-score of 0.7 in the previous samples, they have scored up to 0.296 in this third sample. This indicates that doing without the filter does not harm the selection of equivalents too much because this sample includes short verses and so word correspondences are nearly similar in many instances of this sample. Nonetheless, using the filter has increased the scores in all algorithms.

(ii) The scores on open-class words only are higher than the scores on all words because many of the closed-class words are excluded by the filter. This also explains the big difference in the precision and recall scores in this sample. The precision score in this sample is the highest of all samples when stemming the Arabic text, but the recall is the lowest of all. The result is thus a lower F-score in this sample. This may be partly due to the fact that there are many Arabic words in this sample that are translated into English MWEs. The proposer could not get even the meaning-bearing component of such MWEs and thus left it empty without translation. Also this sample includes many closed-class words that have been excluded when the filter is used and thus this leads to a high precision but low recall.

The other general observations about all the three samples can be summarized as follows:-

- Regarding the F-score we find that the first sample scored better than the second and third samples with respect to the algorithm that achieved the best score, namely weighted 2 using the filter on unstemmed Arabic and English. This may be due to the distribution of individual words in the corpus. We will throw light on this issue below.
- By and large, we obtain a higher score of precision than recall in all samples for the raw proposer. In our view this is not bad given the task of translation, since in our task it is precision more than recall that counts. It has been suggested that “what you want a machine translation system to do is to tell

you enough about the document to confirm that you want a proper (human) translation” (oral quotation from Harold Somers).

- Weighted 2 algorithm generally scores very slightly higher than weighted 1 algorithm when not using the filter, especially in the first sample. However, when the filter is used they often have more or less the same score. They achieve the same score also with regard to the accuracy of the lexicons as described above. It seems that weighted 2 algorithm does some improvement compared to weighted 1 algorithm when the filter is not used. But when the filter is used, the filter compensates for this slight difference and both algorithms get the same results.

It is worth noting that the best score we obtained in all the three samples is that achieved by using the weighted 2 filtered algorithm on Arabic and corresponding English verses without stemming. Here are the best scores in the three samples shown in the following table.

Samples	**Precision**	**Recall**	**F-score**
First Sample	0.5597	0.515	0.5364
Second Sample	0.4193	0.39	0.4041
Third Sample	0.4216	0.35	0.3825
Average Score	0.4668	0.4183	0.441

Table 6.33: Raw proposer’s best score on all words in all samples

Samples	**Precision**	**Recall**	**F-score**
First Sample	0.6979	0.5982	0.6442
Second Sample	0.4905	0.4406	0.4642
Third Sample	0.5208	0.3906	0.4464
Average Score	0.5697	0.4764	0.5182

Table 6.34: Raw proposer’s best score on only open-class words in all samples

It is also noteworthy that stemming both Arabic and English or stemming Arabic only got the least results, which is in contrary with the situation for lexicons, where stemming Arabic and not English got the best result as shown above. There is one more point to note here. The difference of scores between an extracted lexicon and the selection of equivalents based on the same lexicon may be attributed to the fact that some words that are correctly suggested in the first position of the lexicon as translational equivalents occur frequently in the tested samples. That is why the scores are higher in the translation of words than the lexicons. It has been observed that there is difference in scores between the three samples discussed above. This may be due to the distribution of words and their frequency in the entire corpus, as will be shown below.

6.3.4.1.2.2 Reasons behind Difference in Scores

The difference in F-score between the three samples of the corpus that we have examined may be due to the distribution of words in the entire corpus. We will show below the frequency of words in the three samples. We will focus only on the best score we have obtained, namely AV & EV with weighted 2 algorithm using the filter. As mentioned earlier, we are mainly interested in open-class (or content) words. So, we will ignore closed-class (or function) words. We will list the open-class words only in every sample in the following tables and discuss the difference between them as far as the accuracy of the translation is concerned. These words are sorted by frequency of occurrence in the corpus to see the relationship between frequency and accuracy.

I- First Sample

Word-Form	Freq.	Proposer's Output	Reference Translation	Accuracy
الله *Allh*	2153	Allah	Allah	R
الأرض *AlOrD*	287	earth	the earth	R
آمنوا *|mnwA*	263	believed	believed	R
قالوا *qAlwA*	249	say	say	R
كفروا *kfrwA*	189	disbelieved	disbelieved	R
الناس *AlnAs*	183	mankind	mankind	R
الكتاب *AlktAb*	163	book	the book	R
عذاب *E*Ab*	150	torment	a torment	R
بالله *bAllh*	139	believe	in Allah	W
ربهم *rbhm*	111	providence	their lord	W
أنزل *Onzl*	95	book	has been sent down	W
يؤمنون *yWmnwn*	86	believe	believe	R
أنفسهم *Onfshm*	72	themselves	themselves	R
قلوبهم *qlwbhm*	65	hearts	their hearts	R

الصلاة *AlSlAp*	58	prayer	the prayer	R
أليم *Olym*	52	painful	painful	R
عظيم *EZym*	49	tremendous	tremendous	R
يقول *yqwl*	39	says	say	R
هدى *hdY*	38	guidance	guidance	R
آمنا *|mnA*	37	secure	we have believed	W
قيل *qyl*	34	she	it is said	W
الآخر *Al|xr*	29	last	last	R
سواء *swA'*	26	equal	equal	R
يشعرون *y$Erwn*	21	aware	aware	R
ينفقون *ynfqwn*	20	expend	expend	R
للمتقين *llmtqyn*	18	admonition	to the pious	W
ريب *ryb*	17	suspicion	suspicion	R
بالغيب *bAlgyb*	12	dog	in the unseen	W
المفلحون *AlmflHwn*	12	prosperers	the prosperers	R
مرض *mrD*	12	sickness	a sickness	R
يوقنون *ywqnwn*	11	certitude	have certitude in	R
رزقناهم *rzqnAhm*	10	secretly	(We) have provided them	W
أبصارهم *ObSArhm*	9	submissive	beholdings	W
بمؤمنين *bmWmnyn*	6	belong	believers	W
يكذبون *yk*bwn*	6	lying	lie	R
تفسدوا *tfsdwA*	4	depreciate	corrupt	W
سمعهم *smEhm*	3	envelopment	their hearing	W
ويقيمون *wyqymwn*	2	******	and keep up	N
أأنذرتهم *OOn*rthm*	2	******	whether you have warned them	N
تنذرهم *tn*rhm*	2	******	warned them	N
غشاوة *g$Awp*	2	envelopment	an envelopment	R
يخادعون *yxAdEwn*	2	deceive	(they) try to deceive	P
فزادهم *fzAdhm*	2	******	has increased them	N
وبالآخرة *wbAl|xrp*	1	******	the hereafter	N
ختم *xtm*	1	envelopment	has set a seal	W
وباليوم *wbAlywm*	1	******	and in the day	N
يخدعون *yxdEwn*	1	******	deceive	N
مرضا *mrDA*	1	******	sickness	N

Table 6.35: Accuracy along with the frequency of open-class words in the first sample

II- Second Sample

Word-Form	**Freq.**	**Proposer's Output**	**Reference Translation**	**Accuracy**
الله *Allh*	2153	Allah	Allah	R
الأرض *AlOrD*	287	earth	the earth	R
قالوا *qAlwA*	249	say	have said	R
الكتاب *AlktAb*	163	book	the Book	R
لله *llh*	116	praise	to Allah	W

ربنا rbnA	106	make	our lord	W
أنزل Onzl	95	book	has sent down	W
المؤمنين AlmWmnyn	78	believers	the believers	R
أصحاب OSHAb	62	companions	companions	R
الصالحات AlSAlHAt	61	deeds	deeds of righteousness	W
علم Elm	60	knowledge	knowledge	R
يعملون yEmlwn	56	doing	do	R
يقولون yqwlwn	51	fabricated	they say	W
رحمة rHmp	36	taste	mercy	W
آياتنا |yAtnA	34	ayat	our ayat/signs	R
أحسن OHsn	32	fairest	fairest	R
جعلنا jElnA	29	nation	(We) have made	W
أبدا ObdA	28	forever	forever	R
الحمد AlHmd	26	praise	praise	R
حسنا HsnA	23	provision	fair	W
أجرا OjrA	22	magnificent	reward	W
كلمة klmp	18	word	word	R
فقالوا fqAlwA	18	peace	so they said	W
يجعل yjEl	15	wills	make	W
اتخذ Atx*	15	child	has taken	W
كذبا k*bA	15	lie	a lie	R
ولدا wldA	13	child	a child	R
يؤمنوا yWmnwA	12	believing	believe	R
شديدا $dydA	11	very	(very) strict	P
تخرج txrj	9	white	coming out	W
نفسك nfsk	9	yourself	yourself	R
علما EmlA	8	try	deed	W
عوجا EwjA	7	crooked	crookedness	R
عبده Ebdh	6	suffice	His bondman	W
آثارهم |vArhm	6	tracks	their tracks	R
الحديث AlHdyv	6	skins	discourse	W
زينة zynp	6	hurled	an adornment	W
لينذر lyn*r	4	arabic	to warn	W
أفواههم OfwAhhm	4	displayed	their mouths	W
صعيدا SEydA	4	soil	soil	R
الكهف Alkhf	4	cave	the cave	R
عجبا EjbA	4	wonder	wonder	R
آتنا |tnA	4	page	bring us	W
أسفا OsfA	3	sorrowful	sorrow	R
قيما qymA	2	******	most upright	N
بأسا bOsA	2	torture	violence	W
ويبشر wyb$r	2	******	and to give good tidings to	N
لآبائهم l|bA}hm	2	mistakes	their fathers	W
باخع bAxE	2	consume	consume	R
ماكثين mAkvyn	1	******	staying	N
وينذر wyn*r	1	******	and to warn	N
كبرت kbrt	1	******	an odious	N

لنبلوهم *lnblwhm*	1	******	that (We) may try	N
لجاعلون *ljAElwn*	1	arid	will indeed make	W
جرزا *jrzA*	1	arid	arid	R
حسبت *Hsbt*	1	éarraqîm	you reckon	W
والرقيم *wAlrqym*	1	éarraqîm	and éarraqîm	R
أوى *OwY*	1	dispose	took abode	W
الفتية *Alftyp*	1	dispose	young men	W

Table 6.36: Accuracy along with the frequency of open-class words in the second sample

III- Third Sample

Word-Form	**Freq.**	**Proposer's Output**	**Reference Translation**	**Accuracy**
الأرض *AlOrD*	287	earth	the earth	R
شيء *$y'*	190	everything	thing	P
الإنسان *AlInsAn*	58	man	man	R
شاء *$A'*	56	decides	decides	R
السبيل *Alsbyl*	28	indigent	the way	W
الماء *AlmA'*	17	therewith	water	W
قتل *qtl*	12	killed	slain	R
جاءك *jA'k*	11	prejudices	has come to you	W
نطفة *nTfp*	11	sperm	a sperm drop	P
أمره *Omrh*	10	spirit	has commanded him	W
يذكر *y*kr*	9	mentioned	constantly remember	W
وتولى *wtwlY*	7	cries	and turned away	W
جاءه *jA'h*	7	riba (usury)	came to him	W
الأعمى *AlOEmY*	7	houses	the blind man	W
تذكرة *t*krp*	7	reminder	a reminder	R
حبا *HbA*	7	grain	grain	R
متاعا *mtAEA*	7	means	an enjoyment	W
الذكرى *Al*krY*	6	profits	the reminding	W
يسعى *ysEY*	6	along	endeavouring	W
يخشى *yx$Y*	6	colors	is apprehensive	W
خلقه *xlqh*	6	sperm	created him	W
مطهرة *mThrp*	5	purified	purified	R
فلينظر *flynZr*	4	money	so, let (man) look	W
فأنبتنا *fOnbtnA*	4	grain	so, We caused to grow	W
يدريك *ydryk*	3	******	makes you realize	N
صحف *SHf*	3	scrolls	scrolls	R
مرفوعة *mrfwEp*	3	upraised	upraised	R
وفاكهة *wfAkhp*	3	fruit	and fruits	R
عبس *Ebs*	2	frowned	(he) frowned	R
يزكى *yzkY*	2	******	cleanse himself	N
استغنى *AstgnY*	2	thinks	thinks himself self-sufficient	W

ذكره *krh*	2	******	will remember it	N
فقدره *fqdrh*	2	******	so (He) determined him	N
ولأنعامكم *wlOnEAmkm*	2	******	and for your cattle	N
فتنفعه *ftnfEh*	1	******	would profit him	N
تصدى *tSdY*	1	attend	attend	R
تلهى *tlhY*	1	******	being unmindful	N
مكرمة *mkrmp*	1	high	high-honored	P
بأيدي *bOydy*	1	scribes	by the hands of	W
سفرة *sfrp*	1	scribes	scribes	R
كرام *krAm*	1	******	honorable	N
بررة *brrp*	1	******	benign	N
أكفره *Okfrh*	1	******	How disbelieving he is!	N
يسره *ysrh*	1	eased	eased for him	R
أماته *OmAth*	1	entombs	makes him to die	W
فأقبره *fOqbrh*	1	entombs	so He entombs him	R
أنشره *On$rh*	1	******	makes him rise again	N
يقض *yqD*	1	******	performs	N
طعامه *TEAmh*	1	******	his food	N
صببنا *SbbnA*	1	abundance	poured	W
صبا *SbA*	1	abundance	in abundance	R
شققنا *$qqnA*	1	clove	(We) clove	R
شقا *$qA*	1	fissures	in fissures	R
وعنبا *wEnbA*	1	vines	and vines	R
وقضبا *wqDbA*	1	clover	and clover	R
وزيتونا *wzytwnA*	1	******	and olives	N
ونخلا *wnxlA*	1	******	and palm trees	N
وحدائق *wHdA}q*	1	dense	and enclosed orchards	W
غلبا *glbA*	1	dense	with dense trees	P
وأبا *wObA*	1	grass	and grass	R

Table 6.37: Accuracy along with the frequency of open-class words in the third sample

The letter (R) is used to indicate that the translation is ‘right’, whereas the letter (W) is used to indicate that it is ‘wrong’. When the letter (P) is used, it means that the translation is ‘partially correct’ and thus is given a half score, as mentioned earlier. The stars (******) used in the table mean that the proposer could not suggest a TL word for the SL word in question and the letter (N) is thus used to mean ‘no answer is given’. The case of partial accuracy occurs when an Arabic word is translated in the English corpus as an MWE, which includes often two or more words. The proposer picks up only one word from the whole MWE. As we mentioned earlier, if the proposed word is the meaning-bearing word, we give it a half score. Otherwise, it is considered totally wrong. It is worth mentioning that the words that occur more than once in the above samples are mentioned only once in the table. Thus, the word الكهف *Alkhf*, for instance, is repeated twice in the second sample and is translated with the same word “cave”. So, it is mentioned only one time in the table.

We have indicated earlier that we cannot account for tense differences in the translation of verbs in our corpus, because in some cases Arabic past tense verbs are normally translated into English present or future tense verbs. This phenomenon is recurrent in the Qur’anic corpus, where talk about future events is expressed in past tense verbs to signify that these events will inevitably take place. So, in our reference translation the verb شاء *$A’*, for instance, is translated in the verses of the third sample as “decides” in the present tense, though the Arabic verb is used in the past tense.

The difference in the proposer’s performance with regard to the three samples can be attributed to the frequency of words in the entire corpus. For instance, the first sample includes a number of words that have high frequency in the corpus. There are 25 words in this sample that have scored over 20 hits and most of them are translated correctly, whereas there are only 5 words in the third sample that have scored more than 20 hits in the entire corpus. Similarly, the second sample includes many words that have high frequency in the entire corpus.

We can generally conclude that the more frequent a word is the more likely to be translated correctly by the proposer. This can be thought of as having a double benefit. The first benefit is that of getting them right and the second benefit is that getting the most common words right is advantageous for the task as a whole, since they have high frequency in the corpus and will thus improve the overall system of selection. However, in some cases there are some words that have high frequency and are translated wrongly. This can be attributed to the following reasons.

(A) Most of the high frequency words that are translated wrongly by the proposer are basically cliticized lexical items. These items consist of the main content word, whether verb or noun, and cliticized functional items that may be attached to it at the beginning like prepositions or at the end like pronouns. These clitics are translated as separate words in English. For example, the lexical item بالله *bAllh* “in Allah” contains a cliticized preposition besides the main noun in the word. Similarly, the lexical item ربهم *rbhm* “their lord” includes a cliticized pronoun besides the main noun.

(B) Another reason for getting some high frequency words wrong is concerned with the translation of passive verbs. These passive verbs are formed in Arabic by changing the vocalic pattern of words, while it is made in English by a combination of an auxiliary verb and the past participle of the verb in question. In addition, the Arabic passive verb may include a hidden impersonal pronoun that has to be translated also in English. For example, the passive verb قيل *qyl* is translated into English as “it is said”.

(C) A third reason may be due to the fact that many Arabic words are translated as MWEs in English. The proposer deals only with single words. So, it leaves out other components of an MWE. These MWEs may not necessarily be idiomatic expressions, but may be an Adj + Noun compound or Noun+of+Noun compound in the TL. For example, the Arabic word السحت *AlsHt* is translated as “illicit gains”. The proposer suggests only “illicit” as translation and leaves out the second word. Likewise, الصالحات *AlSAlHAt* is translated as “deeds of righteousness”. The proposer picks up “deeds” as a translational equivalent and leaves out the remaining components. Moreover, many Arabic verbs are translated as phrasal verbs in English. These verbs may be used also in the passive voice, which results in more words in the TL text. For example, the passive verb أنزل *Onzl* is translated into English as “has been sent down”. The proposer cannot suggest the four TL words as candidates for the SL word.

As a matter of fact, the above-mentioned cases, i.e. cliticized words, passive verbs and MWEs, are not handled by the proposer. But these points can be tackled in the future. We have stated earlier that the proposer selects a single TL word. It cannot pick up a combination of words as a likely candidate for an SL lexical item. This is because of the constraints under which we are doing the current research. These constraints are lack of a lexicon, lack of punctuation in the text under analysis and lack of fine-grained morphological analysis.

6.3.4.2 Summary

We have described our proposer for raw unannotated texts. These texts may be stemmed or unstemmed. We have evaluated the proposer with regard to two main points that comprise the structure of the proposer, namely bilingual lexicon building and target word selection in context. We have observed that the best score we have obtained with regard to the extraction of bilingual lexicons is achieved on stemmed Arabic text and unstemmed English text. But the situation is different with regard to the selection of words in context, where the best scores are obtained on unstemmed Arabic and English texts. The scoring algorithm that achieved the best score in both modules is weighted 2, though with slight difference, if any, from weighted 1 algorithm.

6.3.5 Bilingual Proposer for Tagged Texts

Now we will discuss the application of the bilingual proposer to POS-tagged texts and the different results that we have obtained in this regard. The same methods of evaluation, i.e. F-score and CWS, are used to test the accuracy of the tagged proposer. We evaluate the tagged proposer, as the case with the raw proposer, with both automatic extraction of bilingual lexicons and selection of TL translational candidates in their context. We will start with presenting the algorithm that we have used for the tagged proposer and then discuss its evaluation.

6.3.5.1 Algorithm

The same general method that we have applied to raw texts is also applied to POS-tagged texts but with an added constraint that will be discussed below. Both the Arabic corpus and its English translation corpus now consist of tuples comprising a given word in addition to its POS tag and the actual number of its position in the corpus. We have given a portion of the Arabic and English POS-tagged corpus in isolation when we discussed the Arabic and English shallow parsers in the previous chapter. For the purpose of illustration we will give below these two portions combined together in parallel to illustrate the way we have modified the main algorithm to work for POS-tagged texts.

(::,newverse,13099)(*lk,DEMO,13100)(AlktAb,NN,13101)(lA,PART,13102)(ryb,NN,13103)(fyh,PREP+PRO,13104)(hdY,NN,13105)(llmtqyn,PREP+NN,13106)(::,newverse,13107)(Al*yn,RELPRO,13108)(yWmnwn,VV,13109)(bAlgyb,PREP+NN,13110)(wyqymwn,CONJ+VV,13111)(AlSlAp,NN,13112)(wmmA,CONJ+PREP+RELPRO,13113)(rzqnAhm,VV+PRO,13114)(ynfqwn,VV,13115)(::,newverse,13116)(wAl*yn,CONJ+RELPRO,13117)(yWmnwn,VV,13118)(bmA,PREP+RELPRO,13119)(Onzl,VV,13120)(Ilyk,PREP+PRO,13121)(wmA,CONJ+PART,13122)(Onzl,VV,13123)(mn,PREP,13124)(qblk,PREP+PRO,13125)(wbAl|xrp,CONJ+NN,13126)(hm,PRO,13127)(ywqnwn,VV,13128)(::,newverse,13129)(Owl}k,DEMO,13130)(ElY,PREP,13131)(hdY,NN,13132)(mn,PREP,13133)(rbhm,NN+PRO,13134)(wOwl}k,CONJ+DEMO,13135)(hm,PRO,13136)(AlmflHwn,NN,13137)	(::,PU,26496)(that,CJ,26497)(is,VB,26498)(the,AT,26499)(book,NN,26500)(there,EX,26501)(is,VB,26502)(no,AT,26503)(suspicion,NN,26504)(about,PR,26505)(it,PN,26506)(a,AT,26507)(guidance,NN,26508)(to,PR,26509)(the,AT,26510)(pious,NN,26511)(::,PU,26512)(who,PN,26513)(believe,VV,26514)(in,PR,26515)(the,AT,26516)(unseen,AJ,26517)(and,CJ,26518)(keep,VV,26519)(up,AV,26520)(the,AT,26521)(prayer,NN,26522)(and,CJ,26523)(expend,NN,26524)(of,PR,26525)(what,DT,26526)(we,PN,26527)(have,VH,26528)(provided,VV,26529)(them,PN,26530)(::,PU,26531)(and,CJ,26532)(who,PN,26533)(believe,VV,26534)(in,PR,26535)(what,DT,26536)(has,VH,26537)(been,VB,26538)(sent,NN,26539)(down,AV,26540)(to,TO,26541)(you,PN,26542)(and,CJ,26543) (what,DT,26544)(has,VH,26545)(been,VB,26546)(sent,NN,26547)(down,AV,26548)(before,PR,26549)(you,PN,26550)(and,CJ,26551)(they,PN,26552)(constantly,AV,26553)(have,VH,26554)(certitude,NN,26555) (in,PR,26556)(the,AT,26557)(hereafter,NN,26558)(::,PU,26559)(those,DT,26560)(are,VB,26561)(upon,PR,26562)(guidance,NN,26563)(from,PR,26564)(their,DP,26565)(lord,NN,26566)(and,CJ,26567)(those,DT,26568)(are,VB,26569)(they,PN,26570)(who,PN,26571)(are,VB,26572)(the,AT,26573)(prosperers,NN,26574)

Figure 6.13: A sample of the POS-tagged parallel corpus

As we can see in the previous sample of the corpus, the numbers for the positions of Arabic words in the corpus are smaller than those for the positions of their corresponding English words. The English positions are nearly the double of their Arabic counterparts. This is due to the fact that the Arabic original text is almost half the English translation in size. The reason for this may be attributed to the fact that an Arabic word could have a number of clitics which are translated into separate words in English. Additionally, the Qur'anic text is characterized by being terse in style and fraught with meanings that need to be expressed in more words in the TL.

As far as the tagged proposer's algorithm is concerned, we have used the same algorithm that we used for the raw proposer but with the addition of the following constraint.

- A chosen TL candidate for a given SL word must have the same POS tag as that of the SL word.

This notion is referred to by Melamed (1995) as follows.

> ".....word pairs that are good translations of each other are likely to be the same parts of speech in their respective languages. For example, a noun in one language is very unlikely to be translated as a verb in another language. Therefore, candidate translation pairs involving different parts of speech should be filtered out."

We thus match TL words with SL words based on the similarity of POS tags in addition to applying the same basic proposed method that we discussed earlier for the raw proposer. However, for this approach to be feasible the tagset for Arabic and English should be similar. Since we are mainly interested in open-class words, we have made the tagset for nouns and verbs similar in the two languages. Thus, we use a more general common tagset, which ignores many of the language-specific details. This has its implication for the current task of finding translational equivalents, as having a fine-grained tagset that pays attention to superficial differences like tense and capitalization would filter out correct translation pairs (Melamed, ibid.). Here lies the reason for having coarse-grained tagsets for both the Arabic and English POS taggers. In this way we have applied our matcher to open-class words only, ignoring all closed-class words (or function words). Accordingly, we match verbs in Arabic with verbs in English and nouns in Arabic with nouns in English. Still, there are some POS tags for open-class words in English that have no counterparts in Arabic. These are **AJ** for 'adjective' and **NP** for 'proper noun'. These POS categories have basically the corresponding POS tag **NN** 'noun' in the Arabic text. We do not have separate tags for adjectives and proper nouns. Therefore, we match **NN** in Arabic with **NN**, **AJ** or **NP** in English.

6.3.5.2 Evaluation

We apply the same measures that we have applied for the raw proposer above. We start with measuring the bilingual lexicons. Then we will discuss the evaluation of TL word selection in the context of verses.

6.3.5.2.1 Bilingual Lexicons

We use the same three scoring algorithms, i.e. baseline, weighted 1 and weighted 2, for bilingual lexicons building. We use the same evaluation measures that we used for evaluating the lexicons extracted by the raw proposer. We start with the first measure which is the F-measure. Then we discuss the other measure, namely CWS. All the F-scores that we have obtained on various types of POS-tagged text using these different algorithms are shown in the following table.

Verses	**Scoring Algorithm**	**Filter**	**Precision**	**Recall**	**F-score**
AV & EV	Baseline	–	0.2676	0.265	0.2663
		+	0.25	0.15	0.1875
	Weighted 1	–	0.4242	0.42	0.4221
		+	0.35	0.21	0.2625
	Weighted 2	–	0.4595	0.455	0.4572
		+	0.35	0.21	0.2625
CAV & CEV	Baseline	–	0.4432	0.43	0.4365
		+	0.3478	0.24	0.2840
	Weighted 1	–	0.5721	0.555	0.5634
		+	0.3623	0.25	0.2958
	Weighted 2	–	0.6134	0.595	0.6040
		+	0.3623	0.25	0.2958
CAV & EV	Baseline	–	0.4329	0.42	0.4263
		+	0.3552	0.27	0.3068
	Weighted 1	–	0.5927	0.575	0.5837
		+	0.4210	0.32	0.3636
	Weighted 2	**–**	**0.6237**	**0.605**	**0.6142**
		+	0.4210	0.32	0.3636
AV & CEV	Baseline	–	0.26	0.26	0.26
		+	0.1923	0.10	0.1315
	Weighted 1	–	0.3989	0.395	0.3969
		+	0.2307	0.12	0.1578

	Weighted 2	–	0.4343	0.43	0.4321
		+	0.2307	0.12	0.1578

Table 6.38: Accuracy of the extracted lexicons using POS-tagged texts

There are a number of observations that can be made about the evaluation of the extracted lexicons using the POS-tagged texts.

(i) When we discussed the raw proposer we illustrated that using the filter achieved far higher scores with all algorithms on all types of text with regard to both lexicon extraction and selection of equivalents in context. Surprisingly, the situation is totally different when we apply the proposer to POS-tagged texts. It is the opposite of what happened before, since not using the filter with POS-tagged texts achieves higher accuracy than using it. This is observed for both bilingual lexicon building and selection of translation pairs in their contextual sentences. This is surprising, since it is normally expected that adding a new constraint will improve things. Thus, using POS tags as another filter besides the main filter of occurrence should have increased the accuracy. However, it turned out that using only one of them has got better results. Thus, using the main filter of occurrence in case of raw texts increases the score dramatically, whereas using the POS tags filter without the main filter has nearly the same effect in case of POS-tagged texts. But combining the two in case of matching words that have similar POS tags decreases the accuracy. This may be due to the following reason:

- As explained earlier, the tagged proposer’s algorithm selects translational equivalents based on matching an SL word with a TL word if they have the same POS tag. So, the extracted lexicon in this way includes all the TL words that have the same POS tags as a given SL word and the other candidates are excluded from the lexicon. However, using the occurrence filter removes some (or in some cases all) of the candidates that are extracted based on POS tags matching. For example, the Arabic word-form وجد *wjd* “found” has the corresponding TL “found” in the lexicon when the filter is not used, since both SL and TL words have the same **VV** tag. Nonetheless, when the main filter of occurrence is used the extracted lexicon for this SL word does not include the TL word “found”, because the filter removes it from the lexicon. This may be due to the fact that the TL word's occurrence is >= 3 times of the SL word's occurrence and thus is removed as we instructed the main filter to do. This incident may be attributed to the following fact: As we noted earlier, the Arabic language is morphologically rich where words are composed of stems and clitics. Thus, different Arabic words share the same stem. This stem in all similar word-forms is translated into the same English word, while the other clitics have corresponding prepositions or pronouns in English. So, the word-forms وجد *wjd* "found", وجده *wjdh* "found him", وجدها *wjdhA* "found her", وجدهم *wjdhm* "found them", وجدك *wjdk* "found you (sing.)", وجدكم *wjdkm* "found you (pl.)", وجدني *wjdny* "found me" and وجدنا *wjdnA* "found us" share the same stem with the same English equivalent "found". In this way the word-form "found" occurs more often than every Arabic word-form of the same class in isolation. This mismatch of occurrence between an SL lexical item and its corresponding TL item should be solved by stemming both texts. Though we did stem both texts, the stemmer we developed still has some mistakes. Two of the extracted lexicons for the Arabic word-form وجد *wjd* can be illustrated in the following figures.

```
{wjd:[(3.416160889934204,'found
'),(2.859511960919260,'said'),(
0.968994140625,'entered'),(0.80
99999999999998,'made'),(0.71127
88503411236,'watering'),(0.6871
537396121884,'grow'),(0.6359557
763178978,'caused'),(0.61734693
87755102,'reached'),(0.56049643
97037374,'keeping'),(0.54934256
05536331,'accepted'),(0.4424757
00709436,'recompense'),(0.43066
40625,'decides'),(0.42401297998
91834,'cannot'),(0.414957281041
81556,'give'),(0.41326530612244
89,'reckoning'),(0.397694992289
5869,'drink'),
(0.366251692149258, 'drive')]}
```

(A) Unfiltered lexicon

```
{wjd:[(0.7112788503411236,'wa
tering'),(0.560494397037374,'
keeping'),(0.5493425605536331
,'accepted'),(0.4240129799891
834,'cannot')]}
```

(B) Filtered Lexicon

Figure 6.14: An Example of two different extracted lexicons for POS-tagged texts using weighted 2 algorithm on unstemmed texts

It should be made clear that the extracted TL words in both lexicons are POS-tagged as verbs like the SL in question and thus are listed in the lexicon.

(ii) The recall score is lower than the precision score when the filter is used. But when the filter is not used, the difference between the precision and recall is very small.

(iii) It has been observed that the best score that has been obtained with respect to bilingual lexicon extraction is that achieved by using ‘weighted 2’ algorithm on stemmed Arabic and unstemmed English. This has been noticed with both the raw proposer and the tagged proposer. However, the situation is totally different with regard to selection of equivalents in context, where the best score is obtained on unstemmed texts in the two languages when testing either the raw or tagged proposer. This will be made clearer when the scores for selection of equivalents are discussed in the following section.

(iv) Using the occurrence filter only in case of the raw proposer has improved the best F-score for the extracted lexicons from 0.07 to 0.489. On the other hand, using the POS tags filter only in case of the tagged proposer has improved the best F-score to 0.614. However, combining both filters as discussed above results in lower scores.

(v) The wrong TL candidates in the bilingual lexicons may be attributed to one of the two following reasons:

1- Both Arabic and English tagged texts have some wrong tags introduced by the Arabic and English POS taggers. There may be the case that an Arabic word is tagged correctly while the corresponding English word is not. The opposite situation may occur where an English word may be tagged correctly and the corresponding Arabic word is not. This, consequently, results in mismatch of tags between some Arabic words and their supposed English equivalents and so are not selected as translation pairs.

2- It may be the case that some SL verbs are translated as nouns in the TL or that some SL nouns are translated as verbs in the TL. In this case the POS tags of both SL and TL words are incompatible and thus no matching is made.

As for the CWS evaluation measure, it has been done for the lexicon that has achieved the best F-score, namely CAVEVW2- and the result is shown in the following table.

Ranks	**Correct Answers**
1	60.5
2	18
3	6
4	1
5	5
6	3.5
7	2
8	2
9	1
10	2.5
CWS	**0.742**

Table 6.39: Confidence-Weighted Score for a bilingual lexicon regarding the first 10 positions using POS-tagged texts

The CWS for the tagged text has scored higher than that for the raw text, as it increased from 0.522 to 0.742.

A sample of this lexicon is given below.

SL Lexical Item	**Suggested TL Words**	**Reference Translation**	**Comments**
جديد *jdyd*	(1) creation **(2) new** (3) indeed (4) come (5) lord (6) case (7) remains (8) earth (9) bones (10) disbelievers	new	The correct translation is suggested as the second word among the TL candidates.
وجه *wjh*	**(1) face** (2) willing (3) turn (4) surrendered (5) said (6) mosque (7) say (8) seeking (9) indeed (10) blackened	face	The correct translation is suggested as the first word among the TL candidates.
ناصبة	******	toiling	No translation is

nASbp			suggested at all.
وجد *wjd*	**(1) found** (2) said (3) say (4) promised (5) reached (6) find (7) take (8) come **(9) finds** (10) entered	found	The correct translation is suggested as the first word among the TL words and another form of the verb is suggested in the 9th position.

Table 6.40: A sample of the bilingual lexicon using weighted 2 algorithm without filtering on stemmed Arabic and unstemmed English

The previous table shows that some words have correct equivalents in the lexicon. These equivalents come in the first, second or any other position in the lexicon. Only the correct equivalents that come in the first positions are given [1] score in F-score evaluation.

6.3.5.2.2 Target Word Selection in Context

We now move on to evaluate the accuracy of the tagged proposer with regard to choosing the correct TL word in its sentential context. We apply the same F-measure to evaluate the proposer on POS-tagged texts. The same observations that we have made concerning extraction of bilingual lexicons are also noticeable in case of the selection of equivalents in their context. This will be made clear when we discuss the scores we have obtained for a number of tested samples below.

6.3.5.2.2.1 Tested Samples

We test the tagged proposer on the same three samples that we used for testing the raw proposer. We will give below the scores for every sample then give the average score for the best algorithm in all the three samples. When we discussed these samples with regard to the raw proposer we gave the scores obtained for both all words including the closed-class words and also for open-class words only. However, for the tagged proposer we can only evaluate the open-class words, which are our concern in the present study. This is because the tagged proposer matches only open-class words based on POS tags similarity and excludes all closed-class words from the matching process. In this way suggested equivalents are given only for open-class words, whereas closed-class words have no corresponding equivalents.

I- First Sample

Verses	**Scoring Algorithm**	**Filter**	**Precision**	**Recall**	**F-score**
AV & EV	Baseline	–	0.625	0.625	0.625
		+	0.6195	0.5089	0.5588
	Weighted 1	–	0.6428	0.6428	0.6428
		+	0.6413	0.5267	0.5784
	Weighted 2	**–**	**0.6785**	**0.6785**	**0.6785**
		+	0.6413	0.5267	0.5784
CAV & CEV	Baseline	–	0.4583	0.2946	0.3586
		+	0.3166	0.1696	0.2209
	Weighted 1	–	0.5416	0.3482	0.4239
		+	0.3166	0.1696	0.2209
	Weighted 2	–	0.5972	0.3839	0.4673
		+	0.3166	0.1696	0.2209
CAV & EV	Baseline	–	0.5138	0.3303	0.4021
		+	0.2968	0.1696	0.2159
	Weighted 1	–	0.5694	0.3660	0.4456
		+	0.2968	0.1696	0.2159
	Weighted 2	–	0.625	0.4017	0.4891
		+	0.3281	0.1875	0.2386
AV & CEV	Baseline	–	0.625	0.625	0.625
		+	0.5108	0.4196	0.4607
	Weighted 1	–	0.6428	0.6428	0.6428
		+	0.5108	0.4196	0.4607
	Weighted 2	-	0.6785	0.6785	0.6785
		+	0.5108	0.4196	0.4607

Table 6.41: Tagged proposer's scores in the first sample

II- Second Sample

Following below are the results we have obtained in the second sample.

Verses	Scoring Algorithm	Filter	Precision	Recall	F-score
AV & EV	Baseline	–	0.60169	0.60169	0.60169
		+	0.4387	0.3644	0.3981
	Weighted 1	–	0.6610	0.6610	0.6610
		+	0.4387	0.3644	0.3981
	Weighted 2	**–**	**0.6779**	**0.6779**	**0.6779**
		+	0.4387	0.3644	0.3981
CAV & CEV	Baseline	–	0.4428	0.2627	0.3297
		+	0.24	0.1016	0.1428
	Weighted 1	–	0.5142	0.3050	0.3829
		+	0.24	0.1016	0.1428
	Weighted 2	–	0.5142	0.3050	0.3829
		+	0.24	0.1016	0.1428
CAV & EV	Baseline	–	0.5428	0.3220	0.4042
		+	0.2857	0.1355	0.1839
	Weighted 1	–	0.6571	0.3898	0.4893
		+	0.2857	0.1355	0.1839
	Weighted 2	–	0.6571	0.3898	0.4893
		+	0.2857	0.1355	0.1839
AV & CEV	Baseline	–	0.5084	0.5084	0.5084
		+	0.3854	0.3135	0.3457
	Weighted 1	–	0.5338	0.5338	0.5338
		+	0.3854	0.3135	0.3457
	Weighted 2	-	0.5508	0.5508	0.5508
		+	0.3854	0.3135	0.3457

Table 6.42: Tagged proposer’s scores in the second sample

III- Third Sample

Here are the scores obtained regarding the third and final sample.

Verses	Scoring Algorithm	Filter	Precision	Recall	F-score
AV & EV	Baseline	–	0.4435	0.4296	0.4365
		+	0.4468	0.3281	0.3783
	Weighted 1	–	0.6693	0.6484	0.6587
		+	0.4893	0.3593	0.4144
	Weighted 2	**–**	**0.6854**	**0.6640**	**0.6746**
		+	0.4893	0.3593	0.4144
CAV & CEV	Baseline	–	0.3611	0.2031	0.26
		+	0.5	0.1796	0.2643
	Weighted 1	–	0.5555	0.3125	0.4
		+	0.5434	0.1953	0.2873
	Weighted 2	–	0.5555	0.3125	0.4
		+	0.5434	0.1953	0.2873
CAV & EV	Baseline	–	0.3888	0.2187	0.28
		+	0.4423	0.1796	0.2555
	Weighted 1	–	0.5833	0.3281	0.42
		+	0.5192	0.2109	0.3
	Weighted 2	–	0.5833	0.3281	0.42
		+	0.5192	0.2109	0.3
AV & CEV	Baseline	–	0.4032	0.3906	0.3968
		+	0.3913	0.2812	0.3272
	Weighted 1	–	0.5645	0.5468	0.5555
		+	0.4130	0.2968	0.3454
	Weighted 2	-	0.5806	0.5625	0.5714
		+	0.4130	0.2968	0.3454

Table 6.43: Tagged proposer's scores in the third sample

It is noticeable in the above tables that the best F-score is obtained using Weighted 2 algorithm on unstemmed Arabic and English but without using the filter. It is thus clear that combining the POS tags constraint with the occurrence constraint results in lower scores than using only one of them, as discussed above. The following table shows the average score for the best algorithm in the three samples.

Samples	Precision	Recall	F-score
First Sample	0.6785	0.6785	0.6785
Second Sample	0.6779	0.6779	0.6779
Third Sample	0.6854	0.6640	0.6746
Average Score	**0.680**	**0.673**	**0.677**

Table 6.44: Tagged proposer's best score in all samples

It is remarkable that the best score for the tagged proposer has been obtained on ustemmed Arabic and English texts regarding the evaluation of TL word selection in context. However, the scores for bilingual lexicons are different where stemming Arabic or both Arabic and English gave better results than using unstemmed data for any of them. The possible reason for this has been discussed before.

A sample of the tagged proposer's selection of equivalents is given in the following table.

Word-Form	Proposer's Output	Reference Translation	Accuracy
الكتاب *AlktAb*	book	the book	R
ريب *ryb*	suspicion	suspicion	R
هدى *hdY*	guidance	guidance	R
للمتقين *llmtqyn*	pious	to the pious	R
يؤمنون *yWmnwn*	believe	believe	R
بالغيب *bAlgyb*	lord	in the unseen	W
ويقيمون *wyqymwn*	keep	and keep up	P
الصلاة *AlSlAp*	prayer	the prayer	R
رزقناهم *rzqnAhm*	provided	(We) have provided them	R
ينفقون *ynfqwn*	expend	expend	R

Table 6.45: A sample of the extracted equivalents by the tagged proposer

6.3.5.3 Summary

In this section we have described the performance of the proposer with regard to the POS-tagged parallel texts, which may be stemmed or unstemmed, as shown above. We have evaluated the proposer in this phase with respect to two main points that comprise the structure of the proposer, namely bilingual lexicon extraction and target word selection in context. We have observed that the best score we have obtained regarding the extraction of bilingual lexicons is achieved on stemmed Arabic text and unstemmed English text. But the situation is different with regard to the selection of equivalents in context, where the best score is achieved on unstemmed Arabic and English texts. The scoring algorithm that obtained the best score in both lexicon extraction and TL word selection in context is weighted 2. This tendency of the tagged proposer regarding the algorithm that achieves the best score echoes that of the raw proposer, though the tagged proposer has achieved a higher score with regard to both lexicon extraction and translation of words in their contextual verses. The only difference is that using the occurrence filter largely increases the scores with respect to the raw proposer, but decreases the scores when used with the tagged proposer.

6.3.6 Bootstrapping Techniques

It is time now to discuss the way we make use of dependency relations (DRs) to aid in improving the final proposer that we have. As shown above, the best final proposer we have now is the ‘tagged proposer’ when applied to unstemmed Arabic and English texts using weighted 2 algorithm without the filter, or AVEVW2- for short. This proposer has achieved an average F-score of 0.677.

To improve the accuracy of this proposer we have applied a number of bootstrapping techniques, making use of the DRs for some basic constructions in both Arabic and English, as described in chapter 5. Having obtained some DRs for some constructions in the two languages, the POS-tagged parallel corpus now includes some DRs for some constructions. As pointed out in chapter 5, in our implementation of the Arabic and English dependency parsing we focus on two elements in a given construction. For instance, we focus on a verb and the immediately following noun. In Arabic this following noun may be the subject or the object and in this case the subject is pro-drop. It is difficult to decide the grammatical function of this noun in Arabic without using a subcategorization lexicon, which we lack in the current project. So, we only obtain the DR without labelling the grammatical function, as we described in the previous chapter. As for English, this following noun is normally functioning as an object since the subject precedes the verb in English. In this way we could match a 'head-dependent' in Arabic with the corresponding 'head-dependent' in English. The 'head-dependent' in the current example is 'verb-noun' in Arabic and 'verb-object' in English. This point is made clearer through giving some examples of the corpus in the following figure.

($qqnA-AlOrD,ATTACHVVNN,109)	**(clove-earth,ATTACHVVNN,216)**
(tWvrwn-AlHyAp,ATTACHVVNN,300)	**(prefer-life,ATTACHVVNN,590)**
(*AqA-Al$jrp,ATTACHVVNN,569)	**(tasted-tree,ATTACHVVNN,1143)**
(fElwA-fAH$p,ATTACHVVNN,683)	**(perform-obscenity,ATTACHVVNN,1388)**
(nfSl-Al|yAt,ATTACHVVNN,778)	**(expound-signs,ATTACHVVNN,1561)**

Figure 6.15: Examples for matching 'head-dependent' equivalents in the parallel corpus

As the previous figure shows, the Arabic head verb and the following dependent noun are attached in a DR. Thus, for instance, شققنا الأرض *$qqnA AlOrD* "clove the earth" is attached together in a DR where the noun depends on the preceding verb. This is clear in the first part of the whole example. Then in the second part the word **ATTACH** is written beside the POS tags of the verb and noun in question. Thus, we see the string **ATTACHVVNN** as one word. As for the third part, it gives the number **109** which is the actual position of the first word, i.e. the verb, in the corpus. The number for the second word is omitted. Similarly, the corresponding English example shows also the same DR between the verb *clove* and the noun *earth*. We focus on the content words here. Thus, the definite article *the* is not included in the English DRs since it is a separate word not a cliticized item like the definite article in Arabic and we do not do tokenization, as indicated in different parts of the thesis. The number of positions for the English examples is nearly the double of the corresponding Arabic number. This indicates that the English text is wordier than the Arabic text and thus poses a challenge for the current task, as pointed out throughout the thesis. Looking at all the examples in the previous figure, we can notice that all of them show the same DR of 'verb-object'. Definitely, 'subject-verb' DR is also exploited along with other relations. But, as described in the previous chapter, the Arabic text is not labelled with such grammatical functions.

We apply this notion of DR between two elements in a given syntactic construction to a number of other patterns that we have illustrated when we discussed the dependency parsers in chapter 5. We end up with a POS-tagged parallel corpus with attachments for some constructions. By doing so, we seek to automatically extract a number of ‘head-dependent’ trusted equivalents. Then we filter these equivalents to obtain a number of one-word translation **seeds** that we could then use to start our bootstrapping techniques. Specifically, these seeds could be used to resegment the parallel corpus to help improve the matching of equivalents in the parallel corpus. It has been indicated that the current corpus is composed of unpunctuated verses where there are no sentence boundaries. Thus, resegmenting the corpus, we hypothesize, could improve the current proposer. Broadly speaking, the bootstrapping techniques can be divided into two basic steps. The first step is the automatic extraction of seeds and the second step is resegmenting the corpus, relying on these seeds. We will discuss each step in detail in the following sections.

6.3.6.1 Extraction of Seeds

As pointed out above, the POS-tagged parallel corpus contains now some DRs. We now start to extract a number of dependency pairs, i.e. ‘head-dependent’ translation pairs. Then we filter these pairs to obtain a number of one-word translation pairs which we call ‘seeds’. These seeds will be used as anchor points to resegment the SL verses and the corresponding TL verses in the parallel corpus and consequently introduce a new alignment of whole SL verses with corresponding TL verses.

To extract dependency pairs, we firstly apply the same algorithm for extracting bilingual lexicons from the parallel corpus, which we have described for the tagged proposer. In this regard, we pointed out that we extract bilingual equivalents based on matching POS tags in both the SL and TL. This time, however, we extract those bilingual equivalents based on matching the compound tags that include the word ATTACH along with the respective POS tags of the ‘head-dependent’ pair. For example, when the string **`ATTACHVVNN`** is seen in an SL verse, the matcher searches the corresponding TL verse for the same compound string to find the translation pairs. This results in a big number of ‘head-dependent’ translation pairs.

This matching between Arabic and English pairs is basically between two dependency trees to find corresponding heads and dependents. This can be made clearer through the following figure.

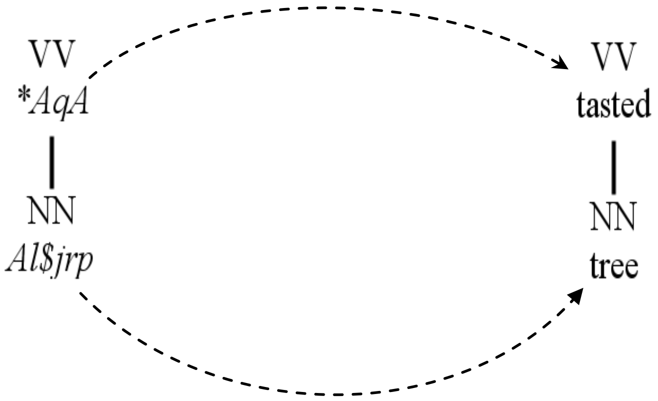


Figure 6.16: Mapping between two dependency trees in Arabic and English

We use the three scoring algorithms we discussed above, namely ‘baseline’, ‘weighted 1’ and ‘weighted 2’ to extract the ‘head-dependent’ bilingual lexicons. We then start to extract the dependency pairs from a given translation lexicon. Firstly, we extract those pairs based on the number of occurrence in the translation lexicon. In fact, every TL ‘head-dependent’ pair has a corresponding number which indicates the number of times this pair occurs with the corresponding SL ‘head-dependent’ pair. This is illustrated in the following figure.

```
{ 'wltjry-Alflk' (1, 'run-ships'), "yrsl-AlsmA'": (2,
'showering-heaven'), 'tjzy-nfs': (2, 'recompense-self'),
'qAl-rbk': (6, 'said-lord'), 'tklm-nfs' (1, 'speak-self'),
'yrsl-AlryAH': (2, 'bearing-winds'), 'qAl-AlmlO': (4, 'said-
people'), 'wgrthm-AlHyAp': (3, 'deluded-life'), 'qdmt-
Oydyhm': (5, 'forwarded-hands'), 'A$trwA-AlDlAlp' (1,
'gained-commerce'), 'qDy-AlOmr': (3, 'decreed-command')
```

Figure 6.17: Bilingual lexicon for extracted ‘head-dependent’ pairs

It should be made clear that the above lexicon of dependency pairs is extracted using the Arabic ‘verb-noun’ DR and the corresponding ‘subject-verb’ relation in English. The ‘noun’ in the Arabic relation may be functioning as the subject, object or the passive subject of verbs in the passive voice. Since the noun (i.e. subject) in the English relation comes before the verb, which is contrary to the order of the noun in the current Arabic relation, we have inverted the order of the English noun to follow the verb so as to be identical to the Arabic order and thus matching is achieved. Most of the ‘head-dependent’ translation pairs in the above lexicon are correct equivalents. However, some pairs are totally wrong such as اشتروا الضلالة *A$trwA-AlDlAlp* “gained-commerce”, which should have the correct pair “traded-errancy”, and some other pairs have one of the elements right and the other wrong, such as يرسل الرياح *yrsl-AlryAH* “bearing-winds”, which should have the correct pair “sends-winds” after ignoring the definite article in the word الرياح *AlryAH* “the winds”. The reason for having such wrong pairs is that the noun in the Arabic DR functions as the object and the subject is pro-drop, whereas the noun in the English DR is definitely the subject. This may result in a wrong translation pair as a whole as is the case with the first example, or that one element of the pair, namely the head, is right and the other is wrong, as is the case with the second example. In other words, with respect to the second example the matching is made between both relations with regard to the head only, i.e. the verb, and not the dependent, i.e. the noun. This mistake will be fixed when we match this Arabic relation with the English ‘verb-object’ relation. Other mistakes will arise, but we will filter the pairs and end up with a number of trusted one-word translation seeds as will be described below. Moreover, as will be explained below, we exclude those ‘head-dependent’ translation pairs that occur less than 2 times, as is the case with the first example. It is worth noting that other bilingual pairs are obtained using the different DRs that were described in chapter 5.

Accordingly, we start the first stage of extracting dependency pairs based on the number of occurrence for a given translation pair. We tried different numbers, setting the threshold at 2 or more occurrences, since we did not trust those translation pairs that occurred only once. Using weighted 1 or weighted 2 algorithms for extracting pairs reduces the number of the correct translation pairs. Thus, we stick to the baseline algorithm with or without the filter on unstemmed text only, as this is the type of text that has achieved the best score in the previous experiments of the raw and tagged proposers.

This first stage produces a large number of candidates, many of which are wrong. We, therefore, carry out a filtering process to obtain a number of trusted one-word translation seeds. To do this filtering we collect all the other TL words that have been suggested as translational candidates for a given element of the dependency pair in question, whether the head or dependent, besides the TL candidate that is given in the current pair. For example, the SL item الحياة *AlHyAp* has the TL candidate "life" in the extracted pair وغرتهم الحياة *wgrthm-AlHyAp* "deluded-life"[18]. In addition, the same TL dependent noun, i.e. "life", has occurred with other heads (i.e. verbs) in other 'head-dependent' extracted pairs, such as تؤثرون الحياة *tWvrwn-AlHyAp* "prefer-life". Then we impose the condition that the occurrence of a TL suggested head or dependent for a given SL head or dependent should be >= the half of the total number of occurrence of all other suggested equivalents with the exception, of course, of the times of occurrence for the current equivalent in the pair before filtering. This can be made clearer when we consider the dictionary of the suggested words for the current dependent الحياة *AlHyAp* "life", which is `{'earth': 1, 'life': 1, 'parent': 1}`. Thus, counting the occurrence of the two other TL candidates, i.e. "earth" and "parent", gives us a total of 2 times. Then, the TL word "life" here occurs half of the total of the other TL words. In fact, the two other words have the same number of occurrence like the word "life" in the dictionary, but the word "life" is selected because it comes with the whole seed in question, i.e. وغرتهم الحياة *wgrthm-AlHyAp* "deluded-life". The effectiveness of this procedure can be shown when we look at other examples like the following dictionary for the word أيديهم *Oydyhm*. It has the following suggested equivalents: `{'legs': 1, 'people': 1, 'angels': 1, 'hands': 3}`. Here the word "hands", which is the correct equivalent when clitics are ignored, occurs 3 times, that is more than half of the total of the three other words. As a matter of fact, we have tried different numbers, setting the threshold at 0.5, 0.3 and 0.25 of the total, but we obtained better scores when we set it to 0.5. Thus, we end up with a number of one-word translation seeds that we automatically collect to be used as anchor points for resegmenting the corpus.

The two above stages for extracting seeds can be generally described as 'finding possible dependency translation pairs' as done in the first stage and then 'obtaining the trusted one-word translation seeds' as done in the second stage. As mentioned above, we tried the three different scoring algorithms with different types of text, i.e. stemmed or unstemmed, and chose the one that resulted in a good number of accurate pairs in the first stage. The best score for trusted pairs was obtained when using the baseline algorithm on unstemmed Arabic text, whether the English text is

[18] The correct equivalent should be "and the life deluded them", but, as illustrated throughout the thesis, we focus on content words and ignore the clitics. This expression is wholly mentioned in the Qur'anic corpus as وغرتهم الحياة الدنيا *wgrthm AlHyAp AldnyA* "and the present life deluded them", but we match only the first noun in the current dependency relation.

stemmed or not. The following table shows the different scores for different algorithms regarding both finding the dependency pairs and then filtering them to obtain the one-word translation seeds. These pairs are extracted using the matching between the Arabic ‘verb-noun’ DR and the English ‘subject-verb’ DR.

Verses	Algorithm	Filter	Freq. of Dependency Pairs = 3 or more	Trusted Head Accuracy	Trusted Dependent Accuracy
AV-EV	Baseline	-	18	8/8	7/9
		+	16	5/5	4/5
	Weighted 1	-	7	3/3	2/4
		+	7	2/2	2/3
	Weighted 2	-	6	2/2	2/4
		+	6	1/1	2/3
CAV-CEV	Baseline	-	20	5/5	6/6
		+	17	4/4	5/5
	Weighted 1	-	11	2/2	2/2
		+	10	2/2	2/2
	Weighted 2	-	10	1/1	2/2
		+	9	1/1	2/2
CAV-EV	Baseline	-	20	5/5	6/6
		+	17	4/4	5/5
	Weighted 1	-	11	2/2	2/2
		+	10	2/2	2/2
	Weighted 2	-	10	1/1	2/2
		+	9	1/1	2/2
AV-CEV	Baseline	-	18	8/8	7/9
		+	16	5/5	4/5
	Weighted 1	-	7	3/3	2/4
		+	7	2/2	2/3
	Weighted 2	-	6	2/2	2/4
		+	6	1/1	2/3

Table 6.46: Accuracy of extracted seeds using Arabic ‘verb-noun’ relation against English ‘subject-verb’ relation

The accuracy in the above table can be read as follows. The first row in the head accuracy column means that 8 head words have correct equivalents out of total 8 suggested words. So, 100% accuracy is achieved for heads using this algorithm. As for dependent accuracy, it means that 7 dependents have correct equivalents out of total 9 suggested words. Thus, the accuracy score here is 77.77% for extracted dependents. In the above table the extracted pairs before filtering are those ones that have occurred in parallel with their English equivalents 3 or more times in the corpus. We also extracted those pairs that occur 2 times with their parallel English equivalents as will be shown below. As for those pairs that occur only once in the corpus, we have just tested their accuracy but have not included them in the final extracted lexicon of seeds, because they give lower scores.

The Arabic 'verb-noun' relation is also compared with the English 'verb-object' relation and the scores obtained are given in the following table.

Verses	Algorithm	Filter	Freq. of Dependency Pairs = 3 or more	Trusted Head Accuracy	Trusted Dependent Accuracy
AV-EV	Baseline	-	113	20/25	25/29
		+	75	18/20	17/19
	Weighted 1	-	58	16/24	15/20
		+	39	11/13	8/10
	Weighted 2	-	49	15/20	14/19
		+	35	11/13	8/10
CAV-CEV	Baseline	-	131	25/30	25/27
		+	89	23/24	19/20
	Weighted 1	-	69	22/29	19/21
		+	46	15/16	12/12
	Weighted 2	-	69	21/25	19/21
		+	42	16/17	12/12
CAV-EV	Baseline	-	131	25/30	27/29
		+	92	23/24	22/23
	Weighted 1	-	68	22/29	20/22
		+	47	15/16	14/14
	Weighted 2	-	59	21/25	19/21
		+	43	16/17	14/14
AV-CEV	Baseline	-	114	19/24	24/28
		+	73	18/20	14/15
	Weighted 1	-	59	16/24	15/19
		+	37	11/13	7/8
	Weighted 2	-	50	15/20	14/19
		+	33	11/13	7/8

Table 6.47: Accuracy of extracted seeds using Arabic 'verb-noun' relation against English 'verb-object' relation

We have explained earlier that we match Arabic 'verb-noun' relation with both 'subject-verb' and 'verb-object' relations in English. This is because the noun that follows the verb in Arabic may be the subject or the object and the subject in this case may be pro-drop. This case of matching is carried out if there is only one noun following the verb in Arabic. But there may be a number of nouns following the verb. We focus only on the first two nouns following an Arabic verb. As noted in the previous chapter, there are a number of possible grammatical functions for the first and second nouns that come after a verb. As repeatedly noted throughout the thesis, we could not distinguish between the grammatical functions of nouns because of the constraints under which we undertake the current project, namely the lack of a lexicon of words and fine-grained morphology. Therefore, if there is only one noun following the Arabic verb, we match this 'verb-noun' relation with English 'subject-verb' and 'verb-object' relations. If there are two nouns following a verb in Arabic, we match each one with English 'subjects' and 'objects'. Furthermore, we match Arabic prepositional phrases (PP), i.e. a preposition followed by a noun, with their English counterparts. The PP may be preceded by a verb or not. If it is preceded by a verb, we match the verb and the noun only in the PP, and leave out the preposition with the corresponding pattern in English. If there is no verb, we match the whole PP with its corresponding PP in English.

The previous tables show that stemming both Arabic and English gives the same number of trusted seeds like stemming Arabic only. Likewise, using unstemmed bi-texts gives the same result as stemming English only. So, what counts here is stemming Arabic or not. We have chosen the baseline algorithm to extract the seeds, since it gives a bigger number of trusted seeds than the other two scoring algorithms. We also used the unstemmed text in Arabic and English, since it is the type of text that achieved better scores in the previous experiment of selecting translational equivalents using POS-tagged texts. The following table shows different scores for the trusted seeds that are extracted using Arabic 'verb-noun' relation against English 'subject-verb' and 'verb-object' relations.

Parallel Relations	**Algorithm**	**Freq. Threshold**	**Pairs**	**Trusted Heads**	**Trusted Deps.**
Arabic 'verb- noun' & English 'subject-verb'	AVEV Baseline-	3	18	8/8	7/9
	AVEV Baseline+		16	5/5	4/5
	AVEV Baseline-	2	82	9/11	18/22
	AVEV Baseline+		58	6/7	13/15

	AVEV Baseline-	1	1504	18/43	44/69
	AVEV Baseline+		1143	14/31	36/53
Arabic ‘verb- noun’ & English ‘verb-object’	AVEV Baseline-	3	113	20/25	25/29
	AVEV Baseline+		75	18/20	17/19
	AVEV Baseline-	2	416	34/51	45/57
	AVEV Baseline+		271	25/30	35/39
	AVEV Baseline-	1	3953	74/127	99/152
	AVEV Baseline+		3196	72/99	93/124

Table 6.48: Accuracy of extracted seeds for Arabic ‘verb-noun’ relation against English ‘subject-verb’ and ‘verb-object’ relations with different frequency thresholds

This previous table shows the accuracy of trusted seeds when we match an Arabic verb that is followed by only one noun with either English ‘subject-verb’ or ‘verb-object’ relations. As pointed out before, sometimes an Arabic verb is followed by two nouns. We match the first noun against the English ‘subject-verb’ or ‘verb-object’ relations. The same step is also done for the second of the two nouns. The accuracy scores for other extracted seeds using a number of other DRs are given in Appendix B.

Generally speaking, we can notice a number of observations about matching dependency pairs using various DRs:

i. The pairs that occur only one time in a given parallel relation are bigger in number than the other pairs that occur two or more times in the corpus, but are lower in their accuracy. We thus trust only those pairs that occur more than 2 times in the parallel corpus.

ii. Matching the second noun following an Arabic verb against English ‘subject-verb’ relation obtains very low scores of accuracy for both heads and dependents, as will be shown in Appendix B. This signifies that the Arabic second noun is not the subject of the verb in this Arabic construction and thus no matching is made between both relations in Arabic and English. The same tendency is also observed when this second noun is matched against the English ‘verb-object’ relation, but with a bit higher score in this case. This may indicate that most cases of the second noun in this Arabic construction in the current corpus are either the second element of a construct phrase or a cognate object, as pointed out in chapter 5.

iii. In most cases the noun following a verb in Arabic is most likely to be the object and the subject is often pro-drop, according to the figures we have obtained. However, this cannot be taken as representative of the Arabic language in general. But it is a general tendency of the structures in the corpus we are using, which is the Qur'anic corpus.

We end up with a number of trusted one-word translation seeds which are then automatically collected in one dictionary. We use these seeds as anchor points for the second stage of bootstrapping techniques, i.e. resegmenting the parallel corpus. The final dictionary of seeds that we use for the coming phase of bootstrapping is given in the following figure.

```
{'|yAt':'signs','$k':'doubt','yHb':'love','|mnwA':'believed','yjA
dlwn':'dispute','rbnA':'lord','|yp':'sign','Ox*':'took','qAl':'sa
id','Oydyhm':'hands','w|twA':'bring','mWmnyn':'sign','wqAl':'said
','yWmnwn':'believe','tsmE':'make','kfrwA':'disbelieved','jnAt':'
gardens',"y$A'":'decides','sryE':'swift',"|bA'nA":'fathers','Alry
AH':'winds','AlmlO':'people','AlOrD':'earth','sbyl':'way','nEdhm'
:'promise','njzy':'recompense','rbhm':'lord','AlOmr':'command','b
|yAtnA':'signs','tdEwn':'invoke','llkAfryn':'disbelievers','kntm'
:'used','DlAl':'error','Onzl':'say','ydEwn':'invoke','Alqwm':'peo
ple','ywEdwn':'promised','flytwkl':'let','fOx*thm':'took','OTyEwA
':'obey','yqwlwn':'say','EbAdh':'bondmen','OmwAlhm':'riches','Al$
ms':'sun','tEbdwn':'worship','bAl|xrp':'hereafter','AlHmd':'prais
e','Al|yAt':'signs','b|yAt':'signs',"OhwA'hm":'prejudices','Alrjf
p':'commotion','wqAlt':'said','trk':'left','rbh':'lord','AlHyAp':
'life','|mn':'believed','jEl':'made','AlsmAwAt':'heavens','Alxlq'
:'creation','wjdnA':'found','tEmlwn':'make','SrAT':'straight','yr
wA':'see','kAnwA':'used','rbk':'lord','AlElm':'knowledge','yryd':
'willing','OwlwA':'endowed','wAlOrD':'earth','Atx*':'taken','yhdy
':'guide','qlnA':'said','ql':'say', 'xlq':'created'}
```

Figure 6.18: The final extracted lexicon of seeds after filtering

In the above lexicon we can notice that some SL words have wrong equivalents, especially when the TL equivalent is an MWE, such as **{'kntm':'used'}** which should be translated as "used to". But generally the precision of this lexicon reaches 0.892, which is a good score to start the second step of our bootstrapping techniques, as explained in the coming section.

6.3.6.2 Resegmenting the Corpus

We now use the high-precision lexicon of seeds, which are obtained from the dependency-parsed corpus, to help in resegmenting the parallel corpus which has no sentence boundaries and thus includes many long verses. The idea behind resegmenting the corpus is that having shorter parallel verses will lead the proposer to perform better than before. As indicated before, we use the seeds as anchor points for resegmenting the parallel corpus. We carry out three different experiments of resegmentation and test the tagged proposer after each one of these experiments. These three experiments can be illustrated as follows:

1- Remove seeds from the parallel corpus and start the tagged proposer on the new bi-texts with the absence of seeds.

2- Resegment the bi-verses in the corpus at the places where one of the seeds is found and keep the seeds.

3- Combine the previous two steps of resegmenting the verses and removing the seeds.

Different scores have been obtained for each of the three experiments for the final tagged proposer that we have. We will present the best score obtained after carrying out each of the three experiments with respect to the selection of open-class equivalents in their contextual verses. We will compare it with the best average score that we obtained before bootstrapping. The scores are given in the following table.

Type of Experiment	**Precision**	**Recall**	**F-score**
Before bootstrapping	0.680	0.673	0.677
After removing seeds only	0.695	0.684	0.690
After resegmentation only	0.701	0.690	0.696
After resegmentation & removing seeds	**0.707**	**0.695**	**0.701**

Table 6.49: Comparison of the F-score for the tagged proposer before and after bootstrapping techniques

It is clear that the best F-score has increased from 0.677 before bootstrapping to 0.701 after the first round of bootstrapping. The F-score for the three types of proposer can be illustrated through the following figure.

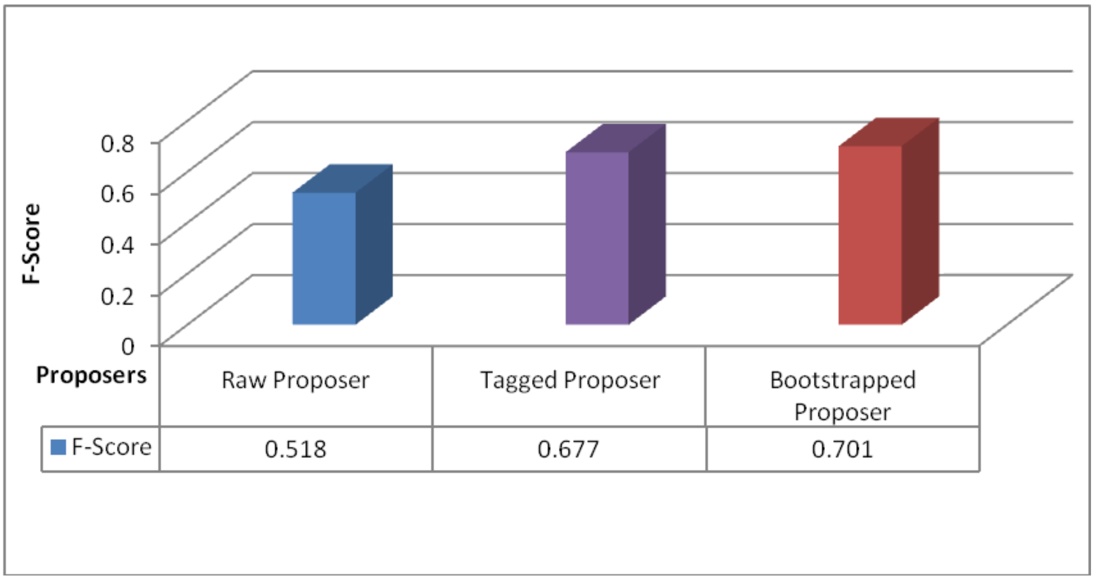


Figure 6.19: Comparison of results for the three different proposers

Having obtained a new parallel corpus after resegmenting the corpus and removing the seeds, we started to carry out another round of bootstrapping by repeating the two main steps of bootstrapping, i.e. extracting new seeds through matching the DRs in the entire corpus and resegmenting the corpus again. We thus increased the number of extracted seeds, i.e. about three times more than before, with an average precision score of 0.832 for the new lexicon of seeds. The extracted seeds in this round are obtained from those dependency pairs that occur 1 or 2 times only, where the ones that occur 3 or more times have been obtained in the previous round and thus have disappeared in this round. We resegmented the parallel corpus and removed all the seeds that we have now in the lexicon, i.e. the old and the new ones together, and tested the tagged proposer again. We hoped that carrying out another round of bootstrapping would improve the situation. However, we did not obtain any extra improvement, and thus did not carry out any further experiments.

6.3.6.3 Summary

In this section we have described the bootstrapping techniques that we have carried out to improve the final tagged proposer that we have. Two main steps were executed to start those bootstrapping techniques. First, a number of 'head-dependent' translation pairs were automatically extracted from the dependency-parsed parallel corpus, where these pairs were then filtered to obtain one-word translation seeds. Second, those seeds were used as anchor points to resegment the corpus to shorten the longer verses in the bi-texts. Then, we tested the tagged proposer on this newly resegmented corpus and obtained an average F-score of 0.701 for the selection of open-class words. This technique helped us cope with the presence of very long sentences in the corpus. As noted at the end of chapter 2, MSA texts also often include very long sentences due to the inconsistent use of punctuation marks (Mubarak et al., 2009b). We did another round of bootstrapping but we stopped at this round, as no further improvement was obtained.

6.3.7 Automatic Detection of Ambiguous Words

It has been pointed out before that ambiguity is an inherent feature of any natural language, which occurs at different levels of representation: lexical, syntactic, semantic, and anaphoric. Humans can easily resolve this inherent ambiguity, depending on the context in which words are used. However, it is a very difficult task for a machine to resolve this ambiguity (Diab, 2003). Since the current study deals with the lexical level, the other types of ambiguity are not of concern to us in this thesis.

Lexical ambiguity is pervasive in a natural language. It is usually the case that a string of words may have a number of interpretations due to the fact that a single word has multiple meanings. It is widely held that lexical ambiguity raises considerable problems for natural language processing and machine translation (MT) (Swart, 1998). It has been claimed that one of the remaining problems in MT persists to be the disambiguation of words, and consequently the problem of selecting the correct translations in the TL (Pedersen, 2000). It is indisputable that lexical ambiguity penetrates both Arabic and English. In other words, words in Arabic or English can be interpreted in different ways. We have pointed out at the beginning of this chapter that one word can have a number of different meanings that may be related (in this case they are **polysemous**) or unrelated and (so they are **homonymous**). We have also discussed a third type of lexical ambiguity which is **homographs**, i.e. two words with the same spelling shape but different meanings and often different pronunciations. We have indicated that this type of ambiguity is widespread in MSA where words are unvocalized. A fourth type of ambiguity is that caused by difference in POS category, which is normally called categorical ambiguity. This type is also characteristic of individual lexical items and is often classified as a source of lexical ambiguity. However, this type of ambiguity, i.e. difference in POS category (e.g. *book* as a “noun” or “verb”) will not concern us in this section, since it should have been resolved by our tagged proposer which selects translational equivalents based on similarity of POS tags in both Arabic and English. The first three types of lexical ambiguity, i.e. polysemes, homonyms and homographs which have the same POS category are what concern us in this part of the thesis.

In fact, how many senses a word has depends on both the genre (i.e. text type) and the task under consideration. For instance, a word can be ambiguous, i.e. have different meanings, in a political text, but is used with only one meaning in a religious type of text. Also, such a word may be considered ambiguous in an MT task, but not so in an information retrieval task, for instance. As far as our current task is concerned, a word is ambiguous if it has been translated with different words in the TL.

The translation lexicons which the tagged proposer outputs include some SL words that are lexically ambiguous. Thus, a given SL word can have a number of corresponding TL words which include the right candidates with their different interpretations besides other wrong candidates. We will give an example for two words from an automatically extracted translation lexicon with the first 10 corresponding TL words to make this point clearer. It should be noted that the corresponding TL list for a given SL word in an extracted lexicon can be of arbitrary length. The following words have more than 10 TL candidates but we focus on the first 10 words only.

```
mlk[(19.374426815131468,'kingdom'),(7.493482055406951,'angel'),(6
.011371395215319,'belongs'),(2.144183123658826,'determiner'),(1.8
532662916233091,'king'),(1.7794710125384323,'ash'),(1.60443405214
79026,'unseen'),(1.5760364700705423,'warner'),(1.2786528041048326
,'intercession'),(1.2050278121819233,'presence')]
EZym[(30.139317036622565,'tremendous'),(10.72725365959638,'magnif
icent'),(7.0,'reward'),(4.482964411954926,'hereafter'),(3.7713859
429718313,'fear'),(3.761472796071772,'present'),(2.70350086917791
2,'odious'),(2.4639149934075464,'owner'),(2.2466737806344645,'wom
en'),(2.1353183957247026, 'disgrace')]
```

Figure 6.20: Examples for ambiguous words in an extracted translation lexicon

The first word in this example is ملك *mlk* which is an Arabic homograph that could be pronounced as *mulk* and in this case mean “kingdom”, as *malak* meaning “angel” or as *malik* meaning “king”. The second word, namely عظيم *EZym*, can be used adjectively for a number of meanings that include “tremendous”, “magnificent” or “monstrous”. We have noted at the beginning of this chapter that these three meanings for this Arabic word differ according to the following word it collocates with. This word is thus a polyseme, where the different meanings of it are related. The first two meanings, i.e. “tremendous” and “magnificent”, occupy the first two positions in this extracted lexicon. As for the third meaning, namely “monstrous”, it did not show up in this lexicon, but has been found in other lexicons.

As pointed out before regarding the structure of the selection process, the proposer selects the first suggested TL word in the extracted lexicon for an SL word in its contextual sentence. In some cases, the proposed word is the right equivalent but in some others the selected word is the wrong equivalent. Thus, these lexically ambiguous words cause a problem for the selection of contextually correct equivalents, which consequently reduces the selection accuracy score. We thought of automatically handling these ambiguous words without any manual intervention. But before handling them we thought of writing an algorithm to automatically detect them in a given translation dictionary, then handle them in the second phase. However, due to time constraints we managed to do the first step, i.e. detecting ambiguous words automatically, but had no time to do the second step. The way of handling lexically ambiguous words will be dealt with in future work. Therefore, we will discuss the algorithm for detecting ambiguous words and the scores we obtain for this task in the following lines.

The algorithm for detecting ambiguous words in a translation lexicon is based on the following notion:

- If a given SL word is ambiguous, it will have different translations in different contexts.

Thus, we examine the SL words in a given extracted lexicon and the first suggested translation among the TL candidates in the entire corpus, applying three parameters to detect an ambiguous word, which are based on the following criteria:

1- The frequency of the SL word in the Arabic corpus, which must be greater than a given threshold *Thr*$_1$.

2- The number of sentences (or rather verses) in which the SL word occurs in the Arabic corpus, providing that the first proposed TL word in the extracted lexicon occurs in the corresponding English verses, which must be greater than a given threshold *Thr_2*.

3- The number of other sentences where the previous requirement *Thr_2* is not met. This also must be greater than a given threshold *Thr_3*.

We have carried out a number of experiments to find out which values are the best for the three above-mentioned parameters. Firstly, {*Thr_1*, *Thr_2*, *Thr_3*} were set to {30, 0.5, 0.2}. We have applied these thresholds on a number of extracted lexicons that are outputted using stemmed or unstemmed texts. We stick to Weighted 2 algorithm, since it is the one that obtains the best score in all the previous experiments in all types of proposer. Since this task is carried out purely automatically without any manual intervention, we do not know the actual number of ambiguous words in a given lexicon. Thus, we can only measure the precision and not the recall in this regard. The precision scores for detecting ambiguous words in such different lexicons are listed in the following table.

Types of Bilingual Lexicons	**Detection Precision**
AV & EV	0.333
CAV & CEV	0.193
CAV & EV	0.135
AV & CEV	0.666

Table 6.50: Precision scores for detecting ambiguous words in different types of extracted bilingual lexicons

We can see in this table that the best precision score is achieved using unstemmed Arabic verses and the canonical (or stemmed) English verses. In fact, using the detection algorithm with the other types of lexicon outputted more words but with significantly low precision as shown in the table. The reason why using stemmed English resulted in better scores than other types of text could be attributed to the following:

- Some TL candidates in a number of extracted lexicons are different morphological variants of the same lexeme, e.g. *guide*, *guides*, *guided*. The detection algorithm regards these variants as different words and wrongly detects them as ambiguous. However, when the TL words are stemmed, the SL word and its corresponding TL candidates do not show up in the algorithm’s output, and so the precision increases.

We thus stick to the extracted lexicon that has achieved the best score, namely AV & CEV and do another round of experiments in which we change the three thresholds mentioned above.

In this round of tests, we kept the same values for {Thr_1, Thr_2} and changed Thr_3 to {0.21} and {0.22} but it gave the same result as {0.2}. But changing this threshold to {0.23} increased the precision score to 0.857, though with the same number of correctly detected ambiguous words. This score remains the same when we increase this threshold till {0.29}. But when we set the threshold to {0.3} we get a precision score of 1.0, since all wrongly detected words are removed and only the rightly detected words remain. In this round the number of rightly detected words is the same but the wrongly detected ones decrease when we increase Thr_3.

Since the best precision score is obtained when {Thr_1, Thr_2, Thr_3} are set to {30, 0.5, 0.3}, we do another round of tests in which we keep both Thr_2, Thr_3 and decrease the frequency threshold Thr_1 to {20}. This time we obtain a score of 0.89 but with more rightly detected words as ambiguous. Decreasing Thr_1 to {10} results in significantly low precision. We consider that the best score is that obtained when {Thr_1, Thr_2, Thr_3} are set to {20, 0.5, 0.3}, which achieves a score of 0.89. This is because it outputs more rightly detected words than the one that achieves 1.0 score. A sample of the output of the best algorithm (this with 0.89 score) with the first three TL words in the lexicon is shown in the following table.

SL Words	**First 3 TL candidates**	**Ambiguity Detection Accuracy**
ملك *mlk*	(1) kingdom (2) angel (3) belong	✓
عظيم *EZym*	(1) tremendous (2) magnificent (3) reward	✓
حق *Hq*	(1) true (2) promise (3) hour	✗

Table 6.51: A sample of the output of the ambiguity detection algorithm

The first two words are correctly detected as ambiguous, whereas the third word is not ambiguous and so it is wrongly detected.

6.4 Summary

In this chapter we have described the main system for corpus-based lexical selection, which we call 'the proposer'. This proposer can have different types of bi-texts as input. These texts may be raw or linguistically annotated. We annotated the bi-texts with POS tags and DRs. We applied the proposer on raw texts, and obtained an average F-score of 0.518 on open-class words, using the co-occurrence filter. We then applied the proposer on POS-tagged bi-texts and obtained an average F-score of 0.677, which was achieved without using the co-occurrence filter. We have noted that using the filter along with POS tags similarity reduced the accuracy score. Then, we used the bi-texts that are annotated with DRs to extract a number of 'head-dependent' translation pairs which were then filtered to obtain a number of one-word translation seeds so as to be used as anchor points for resegmenting the corpus and bootstrapping the selection process once more. The final F-score we obtained after applying the bootstrapping technique to the tagged proposer is 0.701. Thus, the score has increased from 0.677 to 0.701.

It is well-known that lexical ambiguity is pervasive in natural languages. This ambiguity, which prevails in Arabic, affects the accuracy score of lexical selection. Thus, we proceeded to automatically find ambiguous words in a translation dictionary, with a view to handle them to obtain the contextually correct translation for a given word. We described an algorithm for ambiguity detection, which achieves a precision score of 0.89. As for handling the ambiguous words, they will be considered in future work due to time constraints.

Chapter 7

Conclusions and Future Work

In this chapter, the results are summarized in section 7.1, and the main contributions of the research are discussed in section 7.2. Finally, the further research is discussed in section 7.3.

7.1 Thesis Summary

As indicated in the introduction to the present study, we set out to automatically extract lexical equivalents of open-class words from a partially-aligned parallel corpus with a view to machine translation. The methodology we adopted can be applied to any parallel corpus for any language pair, but we have carried out our experiments on an Arabic-English parallel corpus. As pointed out early in the thesis, the corpus we use is challenging due to the nature of the Arabic language used in this particular type of text. This Arabic text has a number of features which make it exceptionally challenging for the current task of lexical selection. The main feature that poses a challenge for the current task is the lack of punctuation in that Arabic text, where there are no sentence boundaries but only verse boundaries. A verse may contain one or more sentences. We have chosen this text mainly because it has an available English translation and also its Arabic orthographic form is diacritized. In fact, we have removed diacritics from the text to be similar to MSA texts which are undiacritized and so highly ambiguous. But we needed the diacritized version at the start of the current project to get our Arabic lexicon-free POS tagger off the ground. The challenging nature of the current corpus emphasizes the robustness of the approach, since it indicates that if the current methods had been applied to an MSA text, which does not contain the challenging feature of lack of punctuation, they would have resulted in better accuracy scores.

In our endeavour to extract translational equivalents from the corpus, we have applied a lexicon-free approach, using as little, if any, hand-coded information as possible. Thus, the point of the work reported here was to see how well one can do without such manual intervention. This allowed us to investigate the effectiveness of different techniques without being distracted by the properties of the lexicon.

To achieve our main goal, we have carried out a number of preprocessing steps prior to starting the selection process. Thus, we have built a lexicon-free POS tagger for Arabic. This POS tagger has been built using a combination of rule-based, transformation-based learning and probabilistic techniques. This tagger requires minimal manual intervention. The first rule-based stage of the tagger, i.e. T_{RB}, made use of the morphological information that is available in diacritized Arabic. So, we used information about possible roots and affixes to detect the POS tag of a given open-class word. As for closed-class words, they are listed in a small dictionary with their appropriate tags. T_{RB} contains a set of patterns which can be matched with the start or end of a word. These patterns, written as REs, are composed of sub-patterns for roots, affixes and clitics. Some of these affixes and clitics are used only with nouns, while some others are used only with verbs. We have exploited this information to initially POS tag words in the text. This stage achieved an overall accuracy score of 75%. Then, in the second stage of the tagger we used TBL technique to correct the errors in the output of T_{RB}, leading to a combined tagger T_{RB+TBL}. TBL is an 'error-driven' machine learning technique for learning an ordered set of transformation rules. It extracts such rules automatically from a pre-tagged training corpus. In TBL every word is first assigned an initial tagging. Then a sequence of rules is applied that change the tags of words based upon the contexts in which they appear. To do this, we have manually corrected the output of T_{RB} on a set of only 1100 words (i.e. a Gold Standard). We used this small-sized Gold Standard to derive a number of corrective rules which were then applied to the entire T_{RB}-tagged corpus. Trying TBL with the Gold Standard (which was all we have correct tags for), we obtained a score of 90.8% correct unambiguous tags. We then removed diacritics from the T_{RB+TBL} tagged corpus and started the third stage of the tagger. This stage was thus applied to undiacritized Arabic. In this stage we used the Bayesian model, simply collecting the conditional probabilities linking the first and last three letters in a word with its tag (Bayesian tagger, T_B). The T_B tagger was then supplemented by considering the parts of speech assigned to the preceding and following words (maximum likelihood tagger, T_{ML}). The final stage involved reusing TBL on the output of T_B+T_{ML} to improve its accuracy. The final best score we obtained was 93.1%. Notably, we first obtained 95.8% accuracy (Ramsay and Sabtan, 2009), but the score decreased after we extended the tagset, as we introduced new separate tags for particular word classes. We used this developed Arabic POS tagger to tag the Arabic text in the parallel corpus. Similarly, we used a lexicon-free English POS tagger developed by Prof. Allan Ramsay at the School of Computer Science, University of Manchester, to POS tag the English text. The English tagger was developed using also machine learning and stochastic techniques.

Having tagged the bi-texts in the parallel corpus with POS categories, we started the second preprocessing step which was obtaining DRs in the parallel corpus. Thus, we have written an Arabic shallow dependency parser for some basic constructions to get the DRs between verbs and their arguments. We could not do full or deep parsing because we did not use a hand-coded lexicon that could give information about the valency (or subcategorization frames) for words. Second, Arabic is a relatively free word order language, where subjects can precede or follow verbs. Third, the text that we dealt with was difficult to parse fully, since there are no punctuation marks to denote sentence boundaries, as noted throughout the thesis. Consequently, we obtained DRs without labelling them with grammatical functions such as *subject*, *object*.etc. The average precision score for these unlabelled DRs in Arabic was 0.956 for five used dependency rules. Similarly, we used a lexicon-free shallow parser for English to obtain DRs between verbs and their arguments. The English parser is also partial not full, but we label DRs with the grammatical functions involved, because English does not have flexibility of word order like Arabic. We mapped between DRs in the Arabic corpus and the English translation in order to extract a number of ‘head-dependent’ pairs which are then filtered to obtain one-word translation seeds to be used as anchor points for resegmenting the long verses in the parallel corpus and restarting the selection process, as will be shown below.

The third preprocessing step was developing a knowledge-free stemmer for Arabic and English. The same approach to stemming was applied to both Arabic and English, with only one exception. For Arabic we removed inflectional prefixes and suffixes, but in case of English we removed only inflectional suffixes. The key idea behind both stemmers lies in firstly clustering similar word variants based on shared number of letters (i.e. supposed roots), and then removing a number of affixes from the clustered words. This tool, like the previous preprocessing steps, is purely corpus-based without any manual intervention. In actual fact, there is a well-known stemmer for English, namely the Porter Stemmer, which could have been used for stemming the English text. Nonetheless, we opted to use the same lexicon-free, corpus-based technique for English as we did with Arabic, so that the work as a whole has the same characteristic of being lexicon-free. The Arabic stemmer scored 0.96 precision with respect to clustering similar words which concerns us in the current task. As for producing the legitimate stem for a word, we did not measure its accuracy, since it is not of concern to us in this study. As for the scores for the English preprocessing tools, we did not measure their accuracy, since they are not part of the contributions of this thesis.

All the previous preprocessing steps pave the way for the main engine which extracts the translational equivalents form the parallel corpus. This engine, which we call 'the proposer', takes as input parallel texts and outputs word correspondences based on word co-occurrence information. These bi-texts are either raw or annotated with POS tags or DRs. We applied the proposer on raw as well as POS-tagged bi-texts and compared the results we have obtained. The basic principle that underlies our approach to proposing lexical equivalents is summarized as follows:

- For each sentence-pair, each word of the TL sentence is a candidate translation for each word of the parallel SL sentence.

This principle means that (*S*, *T*) is a candidate if *T* appears in the translation of a sentence containing *S*. Following the above principle we compute the frequency (the number of occurrences) of each word in the SL and TL sentences. Applying this general principle, the selection process was carried out on two stages:

i. Bilingual Lexicon Building
ii. Target Word Selection in Context

However, this procedure of building the lexicon considered all candidates for inclusion in the lexicon, and thus resulted in significantly low precision and recall. This has occurred because the function words are very common in the corpus, and consequently they were suggested as possible translations for many content words. Therefore, we had to use a 'filter' to exclude such function words from being considered as likely translational candidates. The use of the filter has resulted in higher scores for both lexicon building and translation of words in their context when the proposer was applied on raw texts. However, when we applied the proposer on POS-tagged texts, we found out that combining both constraints, i.e. the filter and POS tags similarity, resulted in lower scores than using POS tags only without the filter. Having automatically extracted a bilingual lexicon from the parallel corpus, we moved on to select the equivalents in their sentential context. The selection of contextual translation is based on the notion of ‘picking up the first TL word in the lexicon that has the highest frequency of occurrence’ as a possible candidate for a given SL word.

The selection of bilingual equivalents using the general principle outlined above is called the ‘baseline’ algorithm. Under the baseline algorithm all words in a TL sentence are considered as possible candidates for each word in the corresponding SL sentence. In order to improve the score of selection process we used two different algorithms that were applied in case of both lexicon extraction and rendering of contextually correct translation for a given word. These algorithms we call ‘weighted 1’ and ‘weighted 2’. Both algorithms give weight to the distance between the relative positions of SL words and corresponding TL words in the same parallel sentence (or verse in our corpus). The only difference between them is that weighted 2 measures the relative distance between words then multiplies the score. Thus, 0.5 becomes {0.5 * 0.5 = 0.25}. This measure gives precedence to those words that are nearer in their positions in a parallel sentence, and disregards those words that are far away.

A number of tests have been carried out to evaluate the proposer on both raw and POS-tagged texts. We call these proposers ‘the raw proposer’ and ‘the tagged proposer’ respectively. We have evaluated both proposers for both lexicon extraction and selection of equivalents in contextual verses. Different measures have been used to evaluate both proposers’ accuracy. As far as bilingual lexicon extraction is concerned, we used two different measures. The first measure is F-measure, which computes both precision and recall. This measure evaluates whether the first suggested TL word in the extracted lexicon is the right equivalent for a given SL word. It scored 0.489 in case of using raw texts and 0.614 when POS-tagged texts are used. The second measure is Confidence-Weighted Score (CWS) to measure the accuracy of the lexicon for the first suggested 10 words. The score reached 0.522 on raw texts and 0.742 on POS-tagged texts. The best F-score for lexicon building was obtained on using ‘weighted 2’ algorithm on stemmed Arabic and unstemmed English. As for evaluating the selection of correct TL equivalent in the context of sentences, we have used the F-score for both raw and tagged proposers. We have tested three different samples that are different with regard to the length of their verses. Focusing on only content words, the scores obtained in the three samples were different for raw texts, with an average F-score of 0.518, but nearly similar in case of POS-tagged texts, with an average score of 0.677. The situation for selection of contextual translation is different from that of lexicon extraction. Notably, the algorithm that achieved the best score for both modules was ‘weighted 2’. Nonetheless, unlike lexicons, using unstemmed Arabic and English achieved the best score in case of selection of TL words in context. This difference between lexicon building and selection of translational equivalents in context may be attributed to the distribution of words, where correct TL candidates occur more frequently in the tested samples.

In order to improve the final tagged proposer we have applied a number of bootstrapping techniques. The basic idea behind these techniques is that having shorter parallel verses will lead to improvement in the selection process. This is because the parallel corpus that we experimented with is composed of unpunctuated verses, where most verses are long, containing a number of sentences that have no boundaries between them. To start the bootstrapping techniques, we used the DR-labelled parallel corpus to extract a number of ‘head-dependent’ pairs that were used as anchor points to resegment the parallel corpus and restart the selection process. The precision score for the extracted seeds reaches 0.892. Having got these seeds, we carried out three methods of bootstrapping as shown below:

1. Remove the seeds from the parallel corpus and start the tagged proposer on the new bi-texts without the seeds.
2. Resegment the bi-verses in the corpus at the places where one of the seeds is found and keep the seeds.
3. Combine the previous two steps of resegmenting the verses and removing the seeds.

The third method resulted in the best F-score, which reached 0.701. As shown in table (3.1), this is comparable with other recent attempts at solving the same problem, especially for Arabic-English pair (Saleh & Habash, 2009; Bangalore et al. 2007). It should be noted, however, that Saleh & Habash (2009) use substantially more linguistic resources (in particular, they use a pre-coded Arabic lexicon in order to detect and hence detach clitic items). At first sight, Bangalore et al. (2007) achieve similar results to us with similar resources. On closer inspection it turns out that these results cover open and closed-class items. It is considerably easier to find equivalents for closed-class items. When we restrict our attention to open-class items, their score falls to 0.662.

We did another round of bootstrapping on the newly resegmented corpus and extracted more seeds. This led to the increase of the number of seeds, with an average precision score of 0.832. Then, we resegmented the corpus and removed all the seeds that we have now in the lexicon, i.e. the old and the new ones together, and tested the tagged proposer again. We hoped that carrying out another round would improve the proposer. But since no higher scores were obtained, we stopped and did not carry out any further experiments.

Ambiguity is an inherent feature of any natural language. This ambiguity problem occurs at different levels of representation: lexical, syntactic, semantic, and anaphoric. Depending on the context in which words are used, people can easily resolve this inherent ambiguity. Nevertheless, it is very difficult for machines to resolve it. Lexical ambiguity, which pertains to lexical items, is prevalent in Arabic as well as English. This lexical ambiguity is manifested in those words which are ‘polysemous’, ‘homonymous’ or ‘homographic’. The third type, namely ‘homographs’, is very common in undiacritized Arabic, since two forms could have the same orthographic form, but differ in meaning and pronunciation. These lexically ambiguous words cause a problem for the selection of contextually correct equivalents, which consequently reduces the selection accuracy score. We thought of handling these ambiguous words automatically, without any manual intervention. To do this we firstly wrote an algorithm to automatically detect ambiguous words in a given translation lexicon. We have conducted a number of tests for this algorithm to obtain the best score we could. We could measure the score for precision but not recall, because we have no idea about how many words are ambiguous in the lexicon in question. The best precision scores were 1.0 for words that occurred in the entire corpus more than 30 times, and 0.89 for words that occurred 20 times in the corpus. We consider that the second result is better than the first because in this case more words were outputted than in the previous one, in spite of being lower in precision. Moreover, it detects the ambiguity for less frequent words, i.e. 20 times or more, while the previous result is achieved on words that occur 30 or more times. Due to time constraints, the other main step of handling such ambiguous words in their context will be done in future work.

7.2 Main Contributions

Drawing upon the previous summary, the thesis has made the following main contributions:

- We have built a lexicon-free POS tagger for undiacritized Arabic, which requires very little manual training and thus overcomes the ‘knowledge acquisition’ bottleneck. Generally speaking, this POS tagger can be useful in other NLP applications on the Arabic language.
- We have developed a lexicon-free stemmer for Arabic. This stemmer reduces similar word-forms in the corpus to a shared form after removing inflectional affixes.
- We have written a shallow dependency parser for Arabic, which produces unlabelled DRs. Initially, we have written a large number of dependency rules between verbs and their different arguments. However, owing to the fact that we do not have a lexicon that includes subcategorization frames, several rules resulted in wrong DRs. So, we used only 5 rules that we trusted.
- The previous contributions were precursors for the main engine in the current study, namely the proposer which extracts translational equivalents from the parallel corpus. This proposer is based on unsupervised statistical techniques without any manual intervention. We used a number of DRs in Arabic and English to extract a number of ‘head-dependent’ pairs that were used as anchor points to resegment the corpus to bootstrap the selection process. We obtained a final F-score of 0.701, which is a reasonable score, given the fact that we deal with partially aligned, unpunctuated bi-texts.
- We have written an algorithm to detect ambiguous words in a given extracted translation dictionary. We could only measure its precision, which is estimated at 0.89.

7.3 Future Work

The author has the following plans for future work:

- Handling Ambiguous Words

The current study has addressed the problem of lexical selection of open-class words for the purpose of MT. Some of these words are lexically ambiguous, having two or more interpretations in different contexts. As far as the bilingual lexical selection is concerned, the words that have different senses but also different POS categories should have been disambiguated by the POS tagger. However, there are some words that have the same POS category and express different meanings. These are mostly polysemes, homonyms and homographs which are prevalent in the Arabic language. We have presented an algorithm to automatically detect such ambiguous words that have the same POS category in a given translation lexicon. However, due to time constraints we could not handle them in their contextual sentences. Thus, more work is needed to disambiguate these words in their context.

- Handling Muti-Word Expressions (MWEs)

Undoubtedly, MWEs are pervasive in any natural language. These MWEs put great hurdles in the way of syntactic parsing, machine translation and information retrieval. These MWEs cover those expressions that are traditionally classified as idioms, phrasal verbs, compound nouns and collocations. It has been pointed out in the current study that some SL words have their corresponding TL words as MWEs in the parallel corpus. The current system of lexical selection proposes one TL word only, thus leaving other words in an MWE in the TL. These MWEs pose a challenge for the current lexical selection task, and consequently reduces the overall accuracy score. Notably, the final F-score that we obtained, i.e. 0.701, could have increased if both ambiguous words and MWEs had been handled in the current system.

- Cascading rules in the current dependency parser for Arabic

Initially we wrote nearly 50 dependency rules in the parser but ultimately used only 5 dependency rules which we trusted in order to obtain the ‘head-dependent’ pairs in the parallel corpus. Currently the 5 dependency rules are executed simultaneously. More work is needed to cascade the used rules so that they can be applied in order. Thus, the most specific dependency rule in the corpus should be given priority of application. Then, it should be followed by other rules according to their specificity.

- Testing on an MSA Corpus

The current framework has been applied on a CA corpus but after removing diacritics from words to mimic the way MSA is written. This has been done with a view to be applied on a parallel corpus of MSA and its English translation. It would be of interest to test the current framework with all its stages on an MSA parallel corpus and see the results that could be obtained. This could be carried out for all the preprocessing steps, i.e. the POS tagger, the shallow dependency parser and the stemmer as well as the main engine, namely the proposer. All these tools have been executed on undiacritized text, with the exception of the early stages of the POS tagger. Thus, for the POS tagger in particular, we should have a diacritized MSA corpus to begin with, and at the start of this work we did not have such a corpus.

Appendix A

The Arabic POS Tagger's Gold Standard

*lk/DEMO AlktAb/NN lA/PART ryb/NN fyh/PREP+PRO hdY/NN llmtqyn/PREP+NN Al*yn/RELPRO yWmnwn/VV bAlgyb/PREP+NN wyqymwn/CONJ+VV AlSlAp/NN wmmA/CONJ+PREP+RELPRO rzqnAhm/VV+PRO ynfqwn/VV wAl*yn/CONJ+RELPRO yWmnwn/VV bmA/PREP+RELPRO Onzl/VV Ilyk/PREP+PRO wmA/CONJ+RELPRO Onzl/VV mn/PREP qblk/PREP+PRO wbAl|xrp/CONJ+PREP+NN hm/PRO ywqnwn/VV Owl}k/DEMO ElY/PREP hdY/NN mn/PREP rbhm/NN+PRO wOwl}k/CONJ+DEMO hm/PRO AlmflHwn/NN In/PART Al*yn/RELPRO kfrwA/VV swA'/NN Elyhm/PREP+PRO OOn*rthm/QPART+VV+PRO Om/CONJ lm/PART tn*rhm/VV+PRO lA/PART yWmnwn/VV xtm/VV Allh/NN ElY/PREP qlwbhm/NN+PRO wElY/CONJ+PREP smEhm/NN+PRO wElY/CONJ+PREP ObSArhm/NN+PRO g$Awp/NN wlhm/CONJ+PREP+PRO E*Ab/NN EZym/NN wmn/CONJ+PREP AlnAs/NN mn/RELPRO yqwl/VV |mnA/VV bAllh/PREP+NN wbAlywm/CONJ+PREP+NN Al|xr/NN wmA/CONJ+PART hm/PRO bmWmnyn/PREP+NN yxAdEwn/VV Allh/NN wAl*yn/CONJ+RELPRO |mnwA/VV wmA/CONJ+PART yxdEwn/VV IlA/PART Onfshm/NN+PRO wmA/CONJ+PART y$Erwn/VV fy/PREP qlwbhm/NN+PRO mrD/NN fzAdhm/CONJ+VV+PRO Allh/NN mrDA/NN wlhm/CONJ+PREP+PRO E*Ab/NN Olym/NN bmA/PREP+RELPRO kAnwA/AUX yk*bwn/VV wI*A/CONJ+PART qyl/VV lhm/PREP+PRO lA/PART tfsdwA/VV fy/PREP AlOrD/NN qAlwA/VV InmA/PART nHn/PRO mSlHwn/NN OlA/PART Inhm/PART+PRO hm/PRO Almfsdwn/NN wlkn/CONJ+CONJ lA/PART y$Erwn/VV wI*A/CONJ+PART qyl/VV lhm/PREP+PRO |mnwA/VV kmA/PART |mn/VV AlnAs/NN qAlwA/VV OnWmn/QPART+VV kmA/PART |mn/VV AlsfhA'/NN OlA/PART Inhm/PART+PRO hm/PRO AlsfhA'/NN wlkn/CONJ+CONJ lA/PART yElmwn/VV wI*A/CONJ+PART lqwA/VV Al*yn/RELPRO |mnwA/VV qAlwA/VV |mnA/VV wI*A/CONJ+PART xlwA/VV IlY/PREP $yATynhm/NN+PRO qAlwA/VV InA/PART+PRO mEkm/PREP+PRO InmA/PART nHn/PRO msthzWwn/NN Allh/NN ysthzY'/VV bhm/PREP+PRO wymdhm/CONJ+VV+PRO fy/PREP TgyAnhm/NN+PRO yEmhwn/VV Owl}k/DEMO Al*yn/RELPRO A$trwA/VV AlDlAlp/NN bAlhdY/PREP+NN fmA/CONJ+PART rbHt/VV tjArthm/NN+PRO wmA/CONJ+PART kAnwA/VV mhtdyn/NN mvlhm/NN+PRO kmvl/PREP+NN Al*y/RELPRO Astwqd/VV nArA/NN flmA/CONJ+PART ODA't/VV mA/RELPRO Hwlh/PREP+PRO *hb/VV Allh/NN bnwrhm/PREP+NN+PRO wtrkhm/CONJ+VV+PRO fy/PREP ZlmAt/NN lA/PART ybSrwn/VV Sm/NN bkm/NN Emy/NN fhm/CONJ+PRO lA/PART yrjEwn/VV Ow/CONJ kSyb/PREP+NN mn/PREP AlsmA'/NN fyh/PREP+PRO ZlmAt/NN wrEd/CONJ+NN wbrq/CONJ+NN yjElwn/VV OSAbEhm/NN+PRO fy/PREP |*Anhm/NN+PRO mn/PREP AlSwAEq/NN H*r/NN Almwt/NN wAllh/CONJ+NN mHyT/NN bAlkAfryn/PREP+NN ykAd/AUX Albrq/NN yxTf/VV ObSArhm/NN+PRO klmA/PART ODA'/VV lhm/PREP+PRO m$wA/VV fyh/PREP+PRO wI*A/CONJ+PART OZlm/VV Elyhm/PREP+PRO qAmwA/VV wlw/CONJ+PART $A'/VV Allh/NN l*hb/COMP+VV bsmEhm/PREP+NN+PRO wObSArhm/CONJ+NN+PRO In/PART Allh/NN ElY/PREP kl/DET $y'/NN qdyr/NN yA/PART OyhA/DET+PRO AlnAs/NN AEbdwA/VV rbkm/NN+PRO Al*y/RELPRO xlqkm/VV+PRO wAl*yn/CONJ+RELPRO mn/PREP qblkm/PREP+PRO lElkm/PART+PRO ttqwn/VV Al*y/RELPRO jEl/VV lkm/PREP+PRO AlOrD/NN frA$A/NN wAlsmA'/CONJ+NN bnA'/NN wOnzl/CONJ+VV mn/PREP AlsmA'/NN mA'/NN fOxrj/CONJ+VV bh/PREP+PRO mn/PREP AlvmrAt/NN rzqA/NN lkm/PREP+PRO flA/CONJ+PART tjElwA/VV llh/PREP+NN OndAdA/NN wOntm/CONJ+PRO tElmwn/VV wIn/CONJ+PART kntm/VV fy/PREP ryb/NN mmA/PREP+RELPRO nzlnA/VV ElY/PREP EbdnA/NN+PRO fOtwA/CONJ+VV bswrp/PREP+NN mn/PREP mvlh/NN+PRO wAdEwA/CONJ+VV $hdA'km/NN+PRO mn/PREP dwn/PREP Allh/NN In/PART kntm/VV SAdqyn/NN fIn/CONJ+PART lm/PART tfElwA/VV wln/CONJ+PART tfElwA/VV fAtqwA/CONJ+VV AlnAr/NN Alty/RELPRO wqwdhA/NN+PRO AlnAs/NN wAlHjArp/CONJ+NN OEdt/VV llkAfryn/PREP+NN wb$r/CONJ+VV Al*yn/RELPRO |mnwA/VV wEmlwA/CONJ+VV AlSAlHAt/NN On/PART lhm/PREP+PRO jnAt/NN tjry/VV mn/PREP tHthA/PREP+PRO AlOnhAr/NN klmA/PART rzqwA/VV mnhA/PREP+PRO mn/PREP vmrp/NN rzqA/NN qAlwA/VV h*A/DEMO Al*y/RELPRO rzqnA/VV mn/PREP qbl/PREP wOtwA/CONJ+VV bh/PREP+PRO mt$AbhA/NN wlhm/CONJ+PREP+PRO fyhA/PREP+PRO OzwAj/NN mThrp/NN whm/CONJ+PRO fyhA/PREP+PRO xAldwn/NN In/PART Allh/NN lA/PART ystHyy/VV On/PART yDrb/VV mvlA/NN mA/PART bEwDp/NN fmA/CONJ+RELPRO fwqhA/PREP+PRO fOmA/CONJ+PART Al*yn/RELPRO |mnwA/VV fyElmwn/CONJ+VV Onh/PART+PRO AlHq/NN mn/PREP rbhm/NN+PRO wOmA/CONJ+PART Al*yn/RELPRO kfrwA/VV fyqwlwn/CONJ+VV mA*A/RELPRO OrAd/VV Allh/NN bh*A/PREP+DEMO mvlA/NN yDl/VV bh/PREP+PRO kvyrA/NN wyhdy/CONJ+VV bh/PREP+PRO kvyrA/NN wmA/CONJ+PART yDl/VV bh/PREP+PRO IlA/PART AlfAsqyn/NN Al*yn/RELPRO ynqDwn/VV Ehd/NN Allh/NN mn/PREP bEd/PREP myvAqh/NN+PRO wyqTEwn/CONJ+VV mA/RELPRO Omr/VV Allh/NN bh/PREP+PRO On/PART ywSl/VV wyfsdwn/CONJ+VV fy/PREP AlOrD/NN Owl}k/DEMO hm/PRO AlxAsrwn/NN kyf/PART tkfrwn/VV bAllh/PREP+NN wkntm/CONJ+VV OmwAtA/NN fOHyAkm/CONJ+VV+PRO vm/CONJ ymytkm/VV+PRO vm/CONJ yHyykm/VV+PRO vm/CONJ Ilyh/PREP+PRO trjEwn/VV hw/PRO Al*y/RELPRO xlq/VV lkm/PREP+PRO mA/RELPRO fy/PREP AlOrD/NN jmyEA/NN vm/CONJ AstwY/VV IlY/PREP AlsmA'/NN fswAhn/CONJ+VV+PRO sbE/NUM smAwAt/NN whw/CONJ+PRO bkl/PREP+DET $y'/NN Elym/NN wI*/CONJ+PART qAl/VV rbk/NN+PRO llmlA}kp/PREP+NN Iny/PART+PRO jAEl/NN fy/PREP AlOrD/NN xlyfp/NN qAlwA/VV OtjEl/QPART+VV fyhA/PREP+PRO mn/RELPRO yfsd/VV fyhA/PREP+PRO wysfk/CONJ+VV AldmA'/NN wnHn/CONJ+PRO nsbH/VV bHmdk/PREP+NN+PRO wnqds/CONJ+VV lk/PREP+PRO qAl/VV Iny/PART+PRO OElm/VV mA/RELPRO lA/PART tElmwn/VV wElm/CONJ+VV |dm/NN AlOsmA'/NN klhA/DET+PRO vm/CONJ ErDhm/VV+PRO ElY/PREP AlmlA}kp/NN fqAl/CONJ+VV Onb}wny/VV+PRO bOsmA'/PREP+NN hWlA'/DEMO In/PART kntm/VV SAdqyn/NN qAlwA/VV sbHAnk/NN+PRO lA/PART Elm/NN lnA/PREP+PRO IlA/PART mA/RELPRO ElmtnA/VV+PRO Ink/PART+PRO Ont/PRO AlElym/NN AlHkym/NN qAl/VV yA/PART |dm/NN Onb}hm/VV+PRO bOsm|}hm/PREP+NN+PRO flmA/CONJ+PART OnbOhm/VV+PRO bOsm|}hm/PREP+NN+PRO qAl/VV Olm/QPART+PART Oql/VV lkm/PREP+PRO Iny/PART+PRO OElm/VV gyb/NN AlsmAwAt/NN wAlOrD/CONJ+NN wOElm/CONJ+VV mA/RELPRO tbdwn/VV wmA/CONJ+RELPRO kntm/AUX tktmwn/VV wI*/CONJ+PART qlnA/VV llmlA}kp/PREP+NN AsjdwA/VV l|dm/PREP+NN fsjdwA/CONJ+VV IlA/PART Iblys/NN ObY/VV wAstkbr/CONJ+VV wkAn/CONJ+VV mn/PREP AlkAfryn/NN wqlnA/CONJ+VV yA/PART |dm/NN Askn/VV Ont/PRO wzwjk/CONJ+NN Aljnp/NN wklA/CONJ+VV mnhA/PREP+PRO rgdA/NN Hyv/PART $}tmA/VV wlA/CONJ+PART tqrbA/VV h*h/DEMO Al$jrp/NN ftkwnA/CONJ+VV mn/PREP AlZAlmyn/NN fOzlhmA/CONJ+VV+PRO Al$yTAn/NN EnhA/PREP+PRO fOxrjhmA/CONJ+VV+PRO mmA/PREP+RELPRO kAnA/VV fyh/PREP+PRO wqlnA/CONJ+VV AhbTwA/VV bEDkm/DET+PRO lbED/PREP+DET Edw/NN wlkm/CONJ+PREP+PRO fy/PREP AlOrD/NN mstqr/NN wmtAE/CONJ+NN IlY/PREP Hyn/NN ftlqY/CONJ+VV |dm/NN mn/PREP rbh/NN+PRO klmAt/NN ftAb/CONJ+VV Elyh/PREP+PRO Inh/PART+PRO hw/PRO AltwAb/NN AlrHym/NN qlnA/VV AhbTwA/VV mnhA/PREP+PRO jmyEA/NN fImA/CONJ+PART yOtynkm/VV+PRO mny/PREP+PRO hdY/NN fmn/CONJ+RELPRO tbE/VV hdAy/NN+PRO flA/CONJ+PART xwf/NN Elyhm/PREP+PRO wlA/CONJ+PART hm/PRO yHznwn/VV wAl*yn/CONJ+RELPRO kfrwA/VV wk*bwA/CONJ+VV b|yAtnA/PREP+NN+PRO Owl}k/DEMO OSHAb/NN AlnAr/NN hm/PRO fyhA/PREP+PRO xAldwn/NN yA/PART bny/NN IsrA}yl/NN A*krwA/VV nEmty/NN+PRO Alty/RELPRO OnEmt/VV Elykm/PREP+PRO wOwfwA/CONJ+VV bEhdy/PREP+NN+PRO Owf/VV bEhdkm/PREP+NN+PRO wIyAy/CONJ+PRO fArhbwn/CONJ+VV w|mnwA/CONJ+VV bmA/PREP+RELPRO Onzlt/VV mSdqA/NN lmA/PREP+RELPRO mEkm/PREP+PRO wlA/CONJ+PART tkwnwA/VV Owl/NUM kAfr/NN bh/PREP+PRO wlA/CONJ+PART t$trwA/VV b|yAty/PREP+NN+PRO vmnA/NN qlylA/NN wIyAy/CONJ+PRO fAtqwn/CONJ+VV wlA/CONJ+PART tlbswA/VV AlHq/NN bAlbATl/PREP+NN wtktmwA/CONJ+VV AlHq/NN wOntm/CONJ+PRO tElmwn/VV wOqymwA/CONJ+VV AlSlAp/NN w|twA/CONJ+VV AlzkAp/NN wArkEwA/CONJ+VV mE/PREP AlrAkEyn/NN OtOmrwn/QPART+VV AlnAs/NN bAlbr/PREP+NN wtnswn/CONJ+VV Onfskm/NN+PRO wOntm/CONJ+PRO ttlwn/VV AlktAb/NN OflA/QPART+CONJ+PART tEqlwn/VV wAstEynwA/CONJ+VV bAlSbr/PREP+NN wAlSlAp/CONJ+NN wInhA/CONJ+PART+PRO lkbyrp/PREP+NN IlA/PART ElY/PREP AlxA$Eyn/NN Al*yn/RELPRO yZnwn/VV Onhm/PART+PRO mlAqw/NN rbhm/NN+PRO wOnhm/CONJ+PART+PRO Ilyh/PREP+PRO rAjEwn/NN yA/PART bny/NN IsrA}yl/NN A*krwA/VV nEmty/NN+PRO Alty/RELPRO OnEmt/VV Elykm/PREP+PRO wOny/CONJ+PART+PRO fDltkm/VV+PRO ElY/PREP AlEAlmyn/NN wAtqwA/CONJ+VV ywmA/NN lA/PART tjzy/VV nfs/NN En/PREP nfs/NN $y}A/NN wlA/CONJ+PART yqbl/VV mnhA/PREP+PRO $fAEp/NN wlA/CONJ+PART yWx*/VV mnhA/PREP+PRO Edl/NN wlA/CONJ+PART hm/PRO ynSrwn/VV wI*/CONJ+PART njynAkm/VV+PRO mn/PREP |l/NN frEwn/NN yswmwnkm/VV+PRO sw'/NN AlE*Ab/NN y*bHwn/VV ObnA'km/NN+PRO wystHywn/CONJ+VV nsA'km/NN+PRO wfy/CONJ+PREP *lkm/DEMO blA'/NN mn/PREP rbkm/NN+PRO EZym/NN wI*/CONJ+PART frqnA/VV bkm/PREP+PRO AlbHr/NN fOnjynAkm/CONJ+VV+PRO wOgrqnA/CONJ+VV |l/NN frEwn/NN wOntm/CONJ+PRO tnZrwn/VV wI*/CONJ+PART wAEdnA/VV mwsY/NN OrbEyn/NUM lylp/NN vm/CONJ Atx*tm/VV AlEjl/NN mn/PREP bEdh/PREP+PRO wOntm/CONJ+PRO ZAlmwn/NN vm/CONJ EfwnA/VV Enkm/PREP+PRO mn/PREP bEd/PREP *lk/DEMO lElkm/PART+PRO t$krwn/VV wI*/CONJ+PART |tynA/VV mwsY/NN AlktAb/NN wAlfrqAn/CONJ+NN lElkm/PART+PRO thtdwn/VV wI*/CONJ+PART qAl/VV mwsY/NN lqwmh/PREP+NN+PRO yA/PART qwm/NN+PRO Inkm/PART+PRO Zlmtm/VV Onfskm/NN+PRO bAtxA*km/PREP+NN+PRO AlEjl/NN ftwbwA/CONJ+VV IlY/PREP bAr}km/NN+PRO fAqtlwA/CONJ+VV Onfskm/NN+PRO *lkm/DEMO xyr/NN lkm/PREP+PRO End/PREP bAr}km/NN+PRO ftAb/CONJ+VV Elykm/PREP+PRO Inh/PART+PRO hw/PRO AltwAb/NN AlrHym/NN wI*/CONJ+PART qltm/VV yA/PART mwsY/NN ln/PART nWmn/VV lk/PREP+PRO HtY/PREP nrY/VV Allh/NN jhrp/NN fOx*tkm/CONJ+VV+PRO AlSAEqp/NN wOntm/CONJ+PRO tnZrwn/VV vm/CONJ bEvnAkm/VV+PRO mn/PREP bEd/PREP mwtkm/NN+PRO lElkm/PART+PRO t$krwn/VV wZllnA/CONJ+VV Elykm/PREP+PRO AlgmAm/NN wOnzlnA/CONJ+VV Elykm/PREP+PRO Almn/NN wAlslwY/CONJ+NN klwA/VV mn/PREP TybAt/NN mA/RELPRO rzqnAkm/VV+PRO wmA/CONJ+PART ZlmwnA/VV+PRO wlkn/CONJ+CONJ kAnwA/AUX Onfshm/NN+PRO yZlmwn/VV wI*/CONJ+PART qlnA/VV AdxlwA/VV h*h/DEMO Alqryp/NN fklwA/CONJ+VV mnhA/PREP+PRO Hyv/PART $}tm/VV rgdA/NN wAdxlwA/CONJ+VV AlbAb/NN sjdA/NN wqwlwA/CONJ+VV HTp/NN ngfr/VV lkm/PREP+PRO xTAyAkm/NN+PRO wsnzyd/CONJ+VV AlmHsnyn/NN fbdl/CONJ+VV Al*yn/RELPRO ZlmwA/VV qwlA/NN gyr/PART Al*y/RELPRO qyl/VV lhm/PREP+PRO fOnzlnA/CONJ+VV ElY/PREP Al*yn/RELPRO ZlmwA/VV rjzA/NN mn/PREP AlsmA'/NN bmA/PREP+RELPRO kAnwA/AUX yfsqwn/VV wI*/CONJ+PART AstsqY/VV mwsY/NN lqwmh/PREP+NN+PRO fqlnA/CONJ+VV ADrb/VV bESAk/PREP+NN+PRO AlHjr/NN fAnfjrt/CONJ+VV mnh/PREP+PRO AvntA/NUM E$rp/NUM EynA/NN qd/PART Elm/VV kl/DET OnAs/NN m$rbhm/NN+PRO klwA/VV wA$rbwA/CONJ+VV mn/PREP rzq/NN

Allh/NN wlA/CONJ+PART tEvwA/VV fy/PREP AlOrD/NN mfsdyn/NN wI*/CONJ+PART qltm/VV yA/PART mwsY/NN ln/PART nSbr/VV ElY/PREP TEAm/NN wAHd/NN fAdE/CONJ+VV lnA/PREP+PRO rbk/NN+PRO yxrj/VV lnA/PREP+PRO mmA/PREP+RELPRO tnbt/VV AlOrD/NN mn/PREP bqlhA/NN+PRO wqv|}hA/CONJ+NN+PRO wfwmhA/CONJ+NN+PRO wEdshA/CONJ+NN+PRO wbSlhA/CONJ+NN+PRO qAl/VV Otstbdlwn/QPART+VV Al*y/RELPRO hw/PRO OdnY/NN bAl*y/PREP+RELPRO hw/PRO xyr/NN AhbTwA/VV mSrA/NN fIn/CONJ+PART lkm/PREP+PRO mA/RELPRO sOltm/VV wDrbt/CONJ+VV Elyhm/PREP+PRO Al*lp/NN wAlmsknp/CONJ+NN wb|WwA/CONJ+VV bgDb/PREP+NN mn/PREP Allh/NN *lk/DEMO bOnhm/PREP+PART+PRO kAnwA/AUX ykfrwn/VV b|yAt/PREP+NN Allh/NN wyqtlwn/CONJ+VV Alnbyyn/NN bgyr/PREP+PART AlHq/NN *lk/DEMO bmA/PREP+RELPRO ESwA/VV wkAnwA/CONJ+AUX yEtdwn/VV In/PART Al*yn/RELPRO |mnwA/VV wAl*yn/CONJ+RELPRO hAdwA/VV wAlnSArY/CONJ+NN wAlSAb}yn/CONJ+NN mn/RELPRO |mn/VV bAllh/PREP+NN wAlywm/CONJ+NN Al|xr/NN wEml/CONJ+VV SAlHA/NN flhm/CONJ+PREP+PRO Ojrhm/NN+PRO End/PREP rbhm/NN+PRO wlA/CONJ+PART xwf/NN Elyhm/PREP+PRO wlA/CONJ+PART hm/PRO yHznwn/VV wI*/CONJ+PART Ox*nA/VV myvAqkm/NN+PRO wrfEnA/CONJ+VV fwqkm/PREP+PRO AlTwr/NN x*wA/VV mA/RELPRO |tynAkm/VV+PRO bqwp/PREP+NN wA*krwA/CONJ+VV mA/RELPRO fyh/PREP+PRO lElkm/PART+PRO ttqwn/VV vm/CONJ twlytm/VV mn/PREP bEd/PREP *lk/DEMO flwlA/CONJ+PART fDl/NN Allh/NN Elykm/PREP+PRO wrHmth/CONJ+NN+PRO lkntm/COMP+VV mn/PREP AlxAsryn/NN wlqd/CONJ+PART Elmtm/VV Al*yn/RELPRO AEtdwA/VV mnkm/PREP+PRO fy/PREP Alsbt/NN fqlnA/CONJ+VV lhm/PREP+PRO kwnwA/VV qrdp/NN xAs}yn/NN fjElnAhA/CONJ+VV+PRO nkAlA/NN lmA/PREP+RELPRO byn/PREP ydyhA/NN+PRO wmA/CONJ+RELPRO xlfhA/PREP+PRO wmwEZp/CONJ+NN llmtqyn/PREP+NN wI*/CONJ+PART qAl/VV mwsY/NN lqwmh/PREP+NN+PRO In/PART Allh/NN yOmrkm/VV+PRO On/PART t*bHwA/VV bqrp/NN qAlwA/VV Ottx*nA/QPART+VV+PRO hzwA/NN qAl/VV OEw*/VV bAllh/PREP+NN On/PART Okwn/VV mn/PREP AljAhlyn/NN qAlwA/VV AdE/VV lnA/PREP+PRO rbk/NN+PRO ybyn/VV lnA/PREP+PRO mA/RELPRO hy/PRO qAl/VV Inh/PART+PRO yqwl/VV InhA/PART+PRO bqrp/NN lA/PART fArD/NN wlA/CONJ+PART bkr/NN EwAn/NN byn/PREP *lk/DEMO fAfElwA/CONJ+VV mA/RELPRO tWmrwn/VV qAlwA/VV AdE/VV lnA/PREP+PRO rbk/NN+PRO ybyn/VV lnA/PREP+PRO mA/RELPRO lwnhA/NN+PRO qAl/VV Inh/PART+PRO yqwl/VV InhA/PART+PRO bqrp/NN SfrA'/NN fAqE/NN lwnhA/NN+PRO tsr/VV AlnAZryn/NN qAlwA/VV AdE/VV lnA/PREP+PRO rbk/NN+PRO ybyn/VV lnA/PREP+PRO mA/RELPRO hy/PRO In/PART Albqr/NN t$Abh/VV ElynA/PREP+PRO wIn|/CONJ+PART+PRO In/PART $A'/VV Allh/NN lmhtdwn/PREP+NN qAl/VV Inh/PART+PRO yqwl/VV InhA/PART+PRO bqrp/NN lA/PART *lwl/NN tvyr/VV AlOrD/NN wlA/CONJ+PART tsqy/VV AlHrv/NN mslmp/NN lA/PART $yp/NN fyhA/PREP+PRO qAlwA/VV Al|n/PART j}t/VV bAlHq/PREP+NN f*bHwhA/CONJ+VV+PRO wmA/CONJ+PART kAdwA/AUX yfElwn/VV

Appendix B

The Arabic Tagset

The following table illustrates both the basic tagset which we use to tag Arabic texts, as described in 4.4.1.2, and the tagger's generated tagset that includes both simple and complex tags.

The Basic Tagset

No.	Tags
1	AUX
2	COMP
3	CONJ
4	DEMO
5	DET
6	DHUW
7	EXVV
8	NN
9	NUM
10	PART
11	PREP
12	PRO
13	QPART
14	RELPRO
15	UN
16	VV

The Tagger's Generated Tagset

No.	Tags	No.	Tags
1	AUX	49	DEMO
2	AUX+PRO	50	DET
3	COMP+VV	51	DET+PRO
4	COMP+VV+PRO	52	DHUW
5	CONJ	53	EXVV
6	CONJ+AUX	54	NN
7	CONJ+COMP+VV	55	NN+PRO
8	CONJ+COMP+VV+PRO	56	NUM
9	CONJ+CONJ	57	NUMT+PRO
10	CONJ+CONJ+PART	58	PART
11	CONJ+CONJ+PREP	59	PART+AUX
12	CONJ+CONJ+PRO	60	PART+DEMO
13	CONJ+DEMO	61	PART+NN
14	CONJ+DET	62	PART+PREP
15	CONJ+DET+PRO	63	PART+PRO
16	CONJ+DHUW	64	PREP
17	CONJ+EXVV	65	PREP+AUX

18	CONJ+NN	66	PREP+DEMO
19	CONJ+NN+PRO	67	PREP+DET
20	CONJ+NUMT	68	PREP+DHUW
21	CONJ+NUMT+PRO	69	PREP+EXVV
22	CONJ+PART	70	PREP+NN
23	CONJ+PART+AUX	71	PREP+NN+PRO
24	CONJ+PART+PREP	72	PREP+NUMT
25	CONJ+PART+PRO	73	PREP+PART
26	CONJ+PREP	74	PREP+PART+PRO
27	CONJ+PREP+AUX	75	PREP+PRO
28	CONJ+PREP+DEMO	76	PREP+QPART+NN
29	CONJ+PREP+DET	77	PREP+QPART+NN+PRO
30	CONJ+PREP+DHUW	78	PREP+QPART+VV
31	CONJ+PREP+EXVV	79	PREP+QPART+VV+PRO
32	CONJ+PREP+NN	80	PREP+RELPRO
33	CONJ+PREP+NN+PRO	81	PREP+VV
34	CONJ+PREP+PART	82	PREP+VV+PRO
35	CONJ+PREP+PRO	83	PRO
36	CONJ+PREP+QPART+NN	84	QPART+CONJ
37	CONJ+PREP+QPART+VV	85	QPART+CONJ+PART
38	CONJ+PREP+RELPRO	86	QPART+NN
39	CONJ+PREP+VV	87	QPART+NN+PRO
40	CONJ+PREP+VV+PRO	88	QPART+PART
41	CONJ+PRO	89	QPART+VV
42	CONJ+QPART+NN	90	QPART+VV+PRO
43	CONJ+QPART+NN+PRO	91	QPART+COMP+VV+PRO
44	CONJ+QPART+VV	92	QPART+PART+PRO
45	CONJ+QPART+VV+PRO	93	RELPRO
46	CONJ+RELPRO	94	UN
47	CONJ+VV	95	VV
48	CONJ+VV+PRO	96	VV+PRO

Appendix C

Accuracy Scores for Translation Seeds

The following tables show the accuracy scores for the extracted seeds, using a number of parallel dependency relations in Arabic and English.

Parallel Relations	Algorithm	Freq. Threshold	Pairs	Trusted Head Seeds	Trusted Dep. Seeds
Arabic 'verb-first noun' & English 'subject-verb'	AVEV Baseline-	3	7	1/1	2/2
	AVEV Baseline+		5	1/1	1/1
	AVEV Baseline-	2	24	1/1	5/5
	AVEV Baseline+		18	1/1	2/2
	AVEV Baseline-	1	543	7/11	15/24
	AVEV Baseline+		409	6/9	8/13
Arabic 'verb-first noun' & English 'verb-object'	AVEV Baseline-	3	48	6/8	8/9
	AVEV Baseline+		31	6/6	7/7
	AVEV Baseline-	2	151	11/15	18/19
	AVEV Baseline+		95	10/11	16/16
	AVEV Baseline-	1	1462	20/28	41/56
	AVEV Baseline+		1189	21/25	39/47
Arabic 'verb-second noun' & English 'subject-verb'	AVEV Baseline-	3	5	0/0	1/2
	AVEV Baseline+		4	0/0	0/1
	AVEV Baseline-	2	22	0/0	1/3
	AVEV Baseline+		15	0/0	0/2

	AVEV Baseline-	1	561	7/16	7/16
	AVEV Baseline+		414	5/8	4/8
Arabic ‘verb-second noun’ & English ‘verb-object’	AVEV Baseline-	3	38	5/7	1/8
	AVEV Baseline+		19	3/3	0/4
	AVEV Baseline-	2	151	12/19	1/15
	AVEV Baseline+		84	11/12	0/7
	AVEV Baseline-	1	1248	28/39	6/24
	AVEV Baseline+		1574	31/46	6/40
Arabic ‘prep.-noun’ & English ‘prep.-noun’	AVEV Baseline-	3	94	8/10	7/18
	AVEV Baseline+		34	3/3	5/5
	AVEV Baseline-	2	87	4/4	7/8
	AVEV Baseline+		214	2/11	9/28
	AVEV Baseline-	1	580	4/6	13/16
	AVEV Baseline+		933	2/12	14/40

Bibliography

Abbès, R., Dichy, J., and Hassoun, M. (2004). The Architecture of a Standard Arabic lexical database: some figures, ratios and categories from the DIINAR.1 source program, In *The Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004*. Geneva, Switzerland, pp. 15-22.

Abdel Haleem, M. (1992). Grammatical Shift for Rhetorical Purposes: *Iltifāt* and Related Features in the Qur'ān. In *Bulletin of the School of Oriental and African Studies*, 55 (3), pp. 407-432.

Abdul-Raof, H. (2001). *Qur'an Translation: Discourse, Texture and Exegesis*. London and New York: Routledge.

Abney, S. P. (1989). A Computational Model of Human Parsing. In *The Journal of Psycholinguistic Research*, 18 (3), pp. 129-144.

Ahmed, H. M. (2005). *Natural Language Processing Engine for Modern Standard Arabic Text-to-Speech*. Ph.D. Thesis, University of Manchester, UK.

Aijmer, K. (2008). Parallel and Comparable Corpora. In A. Lüdeling and M. Kytö (eds.), *Corpus Linguistics: An International Handbook*, vol. 1, Walter de Gruyter GmbH & Co. KG, Berlin, Germany, pp. 257-291.

Al-Alusi, S. (nd). *Rouħ al-Ma^{c}aani fy Tafsiir Al-Qur'an Al^{c}aziim wa al-Sab^c al-Mathani*. Daar Iħyaa' al-Turath Al^{c}araby, Beirut, Lebanon.

Al-Baydawi, A. (1912). *Anwaar al-Tanziil wa Asraar al-Ta'wiil*. Daar al-Kutub al-^{c}arabiyah al-Kubra (Halabi), Cairo, Egypt.

AlGahtani, S., Black, W., and McNaugh, J. (2009). Arabic Part-Of-Speech Tagging Using Transformation-Based Learning. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 22-23 April 2009, Cairo, Egypt, pp. 66-70.

Ali, N. (2003). The Second Wave of Arabic Natural Language Processing. *A paper presented to the Expert Group Meeting on the Promotion of Digital Arabic Content*, Beirut, 3-5 June 2003.

Allen, J. (1995). *Natural Language Understanding*. The Benjamin/Cummings Publishing Company. Inc. Redwood City, CA.

Allerton, D. J. (1982). *Valency and the English Verb*. London: Academic Press.

Al-Maskari, A., and Sanderson, M. (2006). The Affect of Machine Translation on the Performance of Arabic-English QA System. In *Proceedings of EACL-2006: 11th Conference of the European Chapter of the Association for Computational Linguistics, Workshop on Multilingual Question Answering (MLQA06),* Trento, Italy, April 4, 2006, pp. 9-14.

Almisned, O. A. (2001). *Metaphor in the Qur'an: An Assessment of Three English Translations of Suurat Al-Hajj*. Ph.D. Thesis, University of Durham, UK.

Alqrainy, S. (2008). *A Morphological-Syntactical Analysis Approach For Arabic Textual Tagging*. Ph.D. Thesis, De Montfort University, Leicester, UK.

Alqrainy, S., AlSerhan, H. M. and Ayesh, A. (2008). Pattern-based Algorithm for Part-of-Speech Tagging Arabic Text. In *Proceedings of the International Conference on Computer Engineering and Systems (ICCES 08),* Cairo, Egypt, pp. 119-124.

Al Shamsi, F. and Guessoum A. (2006). A Hidden Markov Model–Based POS Tagger for Arabic, In *JADT'06: 8es Journées internationales d'Analyse statistique des Données Textuelles,* France, pp. 31-42.

Altenberg, B. and Granger, S. (2002). Recent Trends in Cross-Linguistic Lexical Studies. In B. Altenberg and S. Granger (eds.), *Lexis in Contrast: Corpus-based Approaches* , Amsterdam/Philadelphia: J. Benjamins, pp. 3-48.

Androutsopoulou, A. (2001). D-raising and Asymmetric Agreement in French In W. Griffin (ed.), *The Role of Agreement in Natural Language, Proceedings of the 2001 Texas Linguistic Society Conference,* Austin, Texas, pp. 35-46.

Arnold, D., Balkan, L., Meijer, S., Humphreys, R. L., and Sadler, L. (1994). *Machine Translation: An Introductory Guide*. Manchester: NCC Blackwell.

Attia, M. A. (2006). An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks. In *The Challenge of Arabic for NLP/MT Conference*, October 2006. The British Computer Society, London, UK, pp. 48-67

Attia, M.A. (2008). *Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation*. Ph.D. Thesis, University of Manchester, UK.

Awad, A. (2005). *Translating Arabic into English with Special Reference to Qur'anic Discourse*. Ph.D. Thesis, University of Manchester, UK.

Badawi, E. M. (1973). *Mustawayat El-Arabiyya El-Mu^{c}asirah fy Misr*, Daar El-Ma^{c}aarif lil-nashr, Egypt.

Badawi, E., Carter, M.G. and Gully, A. (2004) *Modern Written Arabic: A Comprehensive Grammar*. Routledge.

Baker, M. (1992). *In Other Words: a course book on translation*. London and New York: Routledge.

Bangalore, S., Haffner, P., and Kanthak, S. (2007). Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 152–159.

Baptista, M. (1995). On the Nature of Pro-drop in Capeverdean Creole. *Harvard Working Papers in Linguistics*, 5, pp. 3-17.

Barnbrook, G. (1996). *Language and Computers: A Practical Introduction to the Computer Analysis of Language*. Edinburgh University Press.

Beesley, K. R. (1998a). Arabic Morphological Analysis on the Internet. In *The 6th International Conference and Exhibition on Multilingual Computing*, Cambridge, UK.

Beesley, K. R. (1998b). Consonant Spreading in Arabic Stems, In *Proceedings of the 17th International Conference on Computational linguistics COLING'98*, Montreal, Quebec, Canada, pp. 117-123.

Beesley, K. R. (1998c). Arabic Morphology Using Only Finite-State Operations. In *The Workshop on Computational Approaches to Semitic languages*, Montreal, Quebec, pp. 50-57.

Beesley, K. R. (2001). Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001. In *The ACL 2001 Workshop on Arabic Language Processing: Status and Prospects*, Toulouse, France, pp. 1-8.

Belnap, R.K., and Shabaneh, O. (1992). Variable Agreement and Nonhuman Plurals in Classical and Modern Standard Arabic. In E. Broselow, M. Eid and J. McCarthy (eds.), *Perspectives on Arabic Linguistics IV: Papers from the Fourth Annual Symposium on Arabic Linguistics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp. 245-262.

Bennett, P. (2003). The relevance of linguistics to machine translation. In H. L.

Somers (ed.), *Computers and Translation*, Amsterdam: Benjamins, pp. 143-160.

Biber, D., Conrad, S., and Reppen, R. (1998). *Corpus Linguistics: Investigating Language Structure and Use*. Cambridge: Cambridge University Press.

Biber, D., and Finegan, E. (1991). On the Exploitation of Computerized Corpora in Variation Studies. In K. Aijmer and B. Altenberg (eds.), *English Corpus Linguistics*, Longman: London and New York, pp. 204-220.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly Media, Inc.

Bloomfield, L. (1933). *Language*. The University of Chicago Press.

Boguraev, B., and Pustejovsky, J. (1996). Issues in Text-based Lexicon Acquisition. In B. Boguraev and J. Pustejovsky (eds.), *Corpus Processing for Lexical Acquisition*, Cambridge, Mass.; London: MIT Press, pp. 3-17.

Brill, E. (1992). A Simple Rule-Based Part of Speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, Trento, Italy, pp. 152-155.

Brill, E. (1994). Some Advances in Transformation-Based Part of Speech Tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'94)*, Seattle, WA.

Brill, E. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. In *Computational Linguistics*, 21 (4), pp. 543-565.

Brill, E., and Wu, J. (1998). Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of the 17th international conference on Computational linguistics*, vol.1, Montreal, Quebec, Canada, pp. 191-195.

Bröker, N. (1997). *Word Order without Phrase Structure: A Modal Logic for Dependency Grammar*, IMS, Universität Stuttgart.

Brown, P., Cocke, J., Pietra, S. D., Pietra, V. D., Jelinek, F., Mercer, R., and Roossin, P. (1988). A Statistical Approach to Language Translation. In *Proceedings of the 12th Conference on Computational Linguistics (COLING'88)*, Budapest, Hungary, pp. 71-76.

Brown, P., Cocke, J., Pietra, S. D., Pietra, V. D., Jelinek, F., Mercer, R., and Roossin, P. (1990). A Statistical Approach to Machine Translation. In *Computational*

Linguistics, 16 (2), pp. 79-85.

Brown, P., Pietra, S. D., Pietra, V. D., and Mercer, R. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. In *Computational Linguistics*, 19 (2), pp. 263-311.

Buchholz, S., and Marsi, E. (2006). CoNLL-X shared task on Multilingual Dependency Parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York City, USA, pp. 149-164.

Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0. In *Linguistic Data Consortium. Catalog number LDC2002L49, and ISBN 1-58563-257-0.*

Burnard, L. (ed.) (2007). *Reference Guide for the British National Corpus (XML Edition)*. The Research Technologies Service at Oxford University Computing Services. URL: http://www.natcorp.ox.ac.uk/docs/URG/.

Carstairs-McCarthy, A. (2002). *An Introduction to English Morphology: Words and Their Structure*, Edinburgh University Press.

Cavalli-Sforza, V., Soudi, A., and Mitamura, T. (2000*).* Arabic Morphology Generation Using a Concatenative Strategy. In *Proceedings of the First Conference of the North-American Chapter of the Association for Computational Linguistics (NAACL-2000)*, Washington, USA, pp. 86-93.

Chalabi, A. (2000). MT-Based Transparent Arabization of the Internet TARJIM.COM. In *Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future, Lecture Notes In Computer Science*; Vol. 1934. Springer-Verlag, pp. 189-191.

Chalabi, A. (2004a). Sakhr Arabic Lexicon. In *NEMLAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, pp. 21-24.

Chalabi, A. (2004b). Elliptic Personal Pronoun and MT in Arabic. In *JEP-2004-TALN 2004 Special Session on Arabic Language Processing-Text and Speech*.

Charniak, E. (1993). *Statistical Language Learning*. The MIT Press, Cambridge, Massachusetts.

Charniak, E., Knight, K., and Yamada, K. (2003). Syntax-based Language Models for Statistical Machine Translation. In *Proceedings of MT Summit IX*, New Orleans, USA, pp. 40–46.

Chomsky, N. (1957). *Syntactic Structures*. Mouton: The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Chomsky, N. (1981). *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, N. (1995) *The Minimalist Program*. Cambridge, Mass.; London: MIT Press.

Church, K. W., and Mercer, R. L. (1993). Introduction to the Special Issue on Computational Linguistics Using Large Corpora. In *Computational Linguistics*, 19 (1), pp. 1–24.

Cohen, W. (1996). Learning Trees and Rules with Set-valued Features. In *Proceedings of the 13th National Conference on Artificial Intelligence,* Portland, OR, pp. 709-716.

Corbett, G. G. (2001). Agreement: Terms and boundaries. In W. Griffin (ed.), *The Role of Agreement in Natural Language, Proceedings of the 2001 Texas Linguistic Society Conference,* Austin, Texas, pp. 109-122.

Covington, M. A. (1990). A Dependency Parser for Variable-Word-Order Languages. In *Computational Linguistics*, 16, pp. 234-236.

Covington, M. A. (2001). A Fundamental Algorithm for Dependency Parsing. In *Proceedings of the 39th Annual ACM Southeast Conference,* pp. 95-102.

Cruse, D. A. (1986). *Lexical Semantics*. Cambridge University Press.

Cruse, D. A. (2000). *Meaning in Language: An Introduction to Semantics and Pragmatics*. Oxford University Press.

Crystal, D. (2008). *A Dictionary of Linguistics and Phonetics*. 6th ed., Oxford: Blackwell.

Cutting, D., Kupiec, J., Pedersen, J. and Sibun, P. (1992). A Practical Part-of-Speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing,* Trento, Italy, pp. 133-140.

Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. In J. Quinonero-Candela, I. Dagan, B. Magnini, and F. d'Alché-Buc (eds.), *Machine Learning Challenges*. Lecture Notes in Computer Science, Vol. 3944, pp. 177-190, Springer.

Dagan, I., Itai, A., and Schwall, U. (1991). Two Languages Are More Informative

Than One. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berkeley, California, pp. 130-137.

Daimi, K. (2001). Identifying Syntactic Ambiguities in Single-Parse Arabic Sentence. In *Computers and the Humanities*, 35 (3), Kluwer Academic Publishers, pp. 333-349.

Daniels, M. W. (2005). *Generalized ID/LP Grammar: A Formalism for Parsing Linearization-based HPSG Grammars*. Ph.D. Thesis, The Ohio State University.

Debusmann, R., Duchier, D., and Kruijff, G.-J. M. (2004). Extensible Dependency Grammar: A New Methodology. In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*, pp. 78–85.

Diab, M. (2003). *Word Sense Disambiguation Within a Multilingual Framework*. Ph.D. Thesis, University of Maryland.

Diab, M. (2007). Improved Arabic Base Phrase Chunking with a new enriched POS tag set. In *ACL 2007 Proceedings of the Workshop on Computational Approaches to Semitic Languages Common Issues and Resources*, Prague, Czech Republic, pp. 89-96.

Diab, M. (2009). Second Generation AMIRA Tools for Arabic Processing: Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools,* Cairo, Egypt, pp. 285-288.

Diab, M., Hacioglu K., and Jurafsky, D. (2004) Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks, In *Proceedings of NAACL-HLT 2004*. Boston, pp. 149-152.

Dichy, J. (2001). On lemmatization in Arabic. A formal definition of the Arabic entries of multilingual lexical databases. In *ACL 39th Annual Meeting. Workshop on Arabic Language Processing; Status and Prospect*. Toulouse, France, pp 23-30.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research (HLT'02)*, San Francisco, CA, USA, pp. 138–145.

Dukes, K., and Buckwalter, T. (2010) A Dependency Treebank of the Quran using

Traditional Arabic Grammar. In *Proceedings of the 7th International Conference on Informatics and Systems (INFOS 2010)*, Cairo, Egypt, pp. 1-7.

Dukes, K., Atwell, E., and Sharaf, A. (2010). Syntactic Annotation Guidelines for the Quranic Arabic Dependency Treebank. In *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, pp. 1822-1827.

Earl, L. L. (1973). Use of Word Government in Resolving Syntactic and Semantic Ambiguities. In *Information Storage and Retrieval*, 9 (12), pp. 639-664.

Eid, M. (1991). Verbless Sentences in Arabic and Hebrew. In B. Comrie and M. Eid (eds.), *Perspectives on Arabic Linguistics III: Papers from the Third Annual Symposium on Arabic Linguistics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp. 31-61.

El Hadj, Y., Al-Sughayeir, I., and Al-Ansari, A. (2009). Arabic Part-Of-Speech Tagging Using the Sentence Structure, In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 22-23 April 2009, Cairo, Egypt, pp. 241-245.

Elewa, A. (2004). *Collocation and Synonymy in Classical Arabic: A Corpus-Based Study*. Ph.D. Thesis, University of Manchester, UK.

Elimam, A. (2009). *Clause-Level Foregrounding in the Translation of the Qurān into English: Patterns and Motivations*. Ph.D. Thesis, University of Manchester, UK.

Eynde, F. V. (ed.) (1993). *Linguistic Issues in Machine Translation*. London: Pinter Publishers.

Farghaly, A., and Shaalan, K. (2009) Arabic Natural Language Processing: Challenges and Solutions. In *ACM Transactions on Asian Language Information Processing (TALIP)*, 8 (4), pp. 1-22.

Farghaly, A. (2003). Language Engineering and the Knowledge Economy. In A. Farghaly (ed.), *Handbook for Language Engineers*, CSLI Publications, pp. 419-432.

Ferguson, C (1959). Diglossia. In *Word* 15, pp. 325-340.

Fillmore, C. (1968). The Case for Case. In E. Bach and R. T. Harms (eds.) *Universals in Linguistic Theory*, New York: Holt, Rinehart, and Winston, pp. 1-88.

Firth, J.R. (1957). Modes of Meaning. In *Papers in Linguistics*. London: Oxford University Press, pp. 190-215.

Fischer, W. (2002). *A Grammar of Classical Arabic*. 3rd edition, New Haven & London: Yale University Press.

Fitschen, A., Gupta, P. (2008). Lemmatising and Morphological Tagging. In A. Lüdeling and M. Kytö (eds.), *Corpus Linguistics: An International* Handbook, vol. 1, Walter de Gruyter GmbH & Co. KG, Berlin, Germany, pp. 552-563.

Francis, W. N., and Kucera, H. (1979). *Brown Corpus Manual*. Department of Linguistics, Brown University. URL: http://icame.uib.no/brown/bcm.html.

Freeman, A. (2001). Brill's POS Tagger and a Morphology Parser for Arabic. In *ACL 39th Annual Meeting. Workshop on Arabic Language Processing; Status and Prospect*. Toulouse, France, pp. 148-154.

Gaifman, H. (1965). Dependency Systems and Phrase-Structure Systems. In *Information and Control*, 8 (3), pp. 304–337.

Gale, W. A., and Church, K. W. (1991). Identifying Word Correspondences in Parallel Texts. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop,* Pacific Grove, California, Morgan Kaufmann Publishers, San Mateo, California, pp. 152-157.

Gamallo P. (2005). Extraction of Translation Equivalents from Parallel Corpora Using Sense-Sensitive Contexts. In *Proceedings of 10th Conference of the European Association for Machine Translation (EAMT'05),* Budapest, Hungary, pp.97-102.

Gamallo P. (2007). Learning Bilingual Lexicons from Comparable English and Spanish Corpora. In *Proceedings of MT Summit XI,* Copenhagen, Denmark, pp. 191-198.

Garside, R., and Smith, N. (1997). A hybrid grammatical tagger: CLAWS4. In R. Garside, G. Leech, and A. McEnery (eds.), *Corpus Annotation: Linguistic Information from Computer Text Corpora*, Longman, London, pp. 102-121.

Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. A. (1985). *Generalized Phrase Structure Grammar*. Oxford, Blackwell.

Ghali, M. M. (2005). *Towards Understanding the Ever-Glorious Qur'an*, 4th ed. Daar al-nashr lil-Jaami^{c}aat, Cairo, Egypt.

Gough, N. (2005). *Example-Based Machine Translation Using the Marker Hypothesis*. Ph.D. Thesis, Dublin City University, Dublin, Ireland.

Goweder, A., Alhammi, H., Rashid, T., and Musrati, A. (2008). A Hybrid Method for Stemming Arabic Text. In *The 2008 International Arab Conference on Information Technology (ACIT'2008)*, Tunisia.

Green, B. B., and Rubin, G. M. (1971). *Automatic Grammatical Tagging of English*. Department of Linguistics, Brown University, Providence, R.I.

Habash, N. (2007). Arabic Morphological Representations for Machine Translation. In A. Soudi, A. Van den Bosch, and G. Neumann (eds.), *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, Springer, pp. 263-285.

Habash, N., and Rambow, O. (2005). Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop, In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, Michigan, pp. 573-580.

Habash, N., Faraj, R., and Roth. R. (2009). Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 22-23 April 2009, Cairo, Egypt, pp. 125-132.

Habash, N., Rambow, O. (2004). Extracting a Tree Adjoining Grammar from the Arabic Treebank. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*, Fez, Morocco, pp. 277-284.

Haghighi, A., Liang, P., Berg-Kirkpatrick, T., and Klein, D. (2008). Learning Bilingual Lexicons from Monolingual Corpora. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Colombus, Ohio, pp. 771-779.

Hakim, N. (1995). *Syntactic Constraints on Genitive Constructions in Arabic*. M.A. Thesis, University of Manchester, UK.

Halliday, M. (1970). Language Structure and Language Function. In J. Lyons (ed.), *New Horizons in Linguistics*, Penguin Books: Harmondsworth, pp. 140-165.

Hamada, S., and Al Kufaishi, A. (2009). Linguistic Constraints as a Sub-component of a Framework that has Multilinguistic Applications. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 22-23 April 2009, Cairo, Egypt, pp. 90-97.

Hammo, B., Sleit, A., and El-Haj, M. (2008). Enhancing Retrieval Effectiveness of Diacritized Arabic Passages Using Stemmer and Thesaurus. In *19th Midwest Artificial Intelligence and Cognitive Science Conference MAICS 2008*, Ohio, USA, pp. 189-196.

Hammo, B., Sleit, A., and El-Haj, M. (2007). Effectiveness of Query Expansion in Searching the Holy Quran. In *Proceedings of the Second International Conference on Arabic Language Processing CITALA'07*, Rabat, Morocco, pp. 1-10.

Hassan, A. (2007). *Al-Naħw Al-Waafiy*. vol. 1, (1st ed.), Maktabat Al-Muhammadi, Beirut, Lebanon.

Hays, D.G. (1964). Dependency Theory: A Formalism and Some Observations. In *Language*, 40 (4), pp.511-525.

Haywood, J.A., and Nahmad, H.M. (2005). *A new Arabic grammar of the written language*, London: Lund Humphries.

Hearne, M. (2005). *Data-Oriented Models of Parsing and Translation*. Ph.D. Thesis, Dublin City University, Dublin, Ireland.

Hearst, M. A. (1991). Noun Homograph Disambiguation Using Local Context in Large Text Corpora. In *the Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New OED and Text Research: Using Corpora*, Oxford, UK, pp.1-19.

Herslund, M. (1988). On Valence and Grammatical Relations. In F. Sørensen (ed.) *Valency: Three Studies on the Linking Power of Verbs*, Copenhagen: Forlag Arnold Busck, pp. 3-34

Hijazy, M. F., and Yusof, M. I. (1999). *Mu^{c}jam Al-Qawa^{c}id Al-Naħawiya*. Dar Al-Kitab Al Masri, Cairo & Dar Al-Kitab Al-Lubnani, Beirut.

Hirst, G. (1987). *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge: Cambridge University Press.

Hudson, R. (1995) *Word Meaning*, Routledge: London and New York.

Hudson, R. (1984). *Word Grammar*. Basil Blackwell Inc., Oxford, England.

Hudson, R. (1990). *English Word Grammar*. Blackwell.

Hudson, R. (2007). *Language Networks: The New Word Grammar*. Oxford: Oxford University Press.

Hutchins, W., and Somers, H. (1992). *An Introduction to Machine Translation*. London: Academic Press.

Hutchins, W. (1986). *Machine Translation: Past, Present, Future*. Chichester, Ellis Horwood Limited.

Ide, N., and Véronis, J. (1998). Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. In *Computational Linguistics*, 24 (1), pp. 1-40.

Izwaini, S. (2006). Problems of Arabic Machine Translation: Evaluation of Three Systems. In *The Challenge of Arabic for NLP/MT Conference*, The British Computer Society, London, UK, pp. 118-148.

Jackson, H. (1988). *Words and their Meaning*. London: Longman.

Johnson-Davies, D. (2002). *Translating the untranslatable?*, Al-Ahram Weekly, Issue no.573, 14-20 February.

Jurafsky, D., and Martin, J. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Upper Saddle River, New Jersey: Pearson/Prentice Hall.

Kaji, H., and Aizono, T. (1996). Extracting Word Correspondences from Bilingual Corpora Based on Word Co-occurrences Information. In *Proceedings of the 16th International Conference on Computational linguistics (COLING'96)*, Copenhagen, Denmark, pp. 23-28.

Kamir, D., Soreq, N., and Neeman, Y. (2002). A Comprehensive NLP System for Modern Standard Arabic and Modern Hebrew. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, PA, USA, pp. 1-9.

Kaplan, R., and Bresnan, J. (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation. In J. Bresnan (ed.) *The Mental Representation of Grammatical Relations*, Cambridge, MA: The MIT Press, pp.173-281.

Kaplan, R. M. (2005). A Method for Tokenizing Text. In A. Arppe, L. Carlson, K. Lindén, J. Piitulainen, M. Suominen, M. Vainio, H. Westerlund, and A. Yli-Jyrä (eds.), *Inquiries into Words, Constraints and Contexts: Festschrift for Kimmo Koskenniemi on his 60th Birthday*, CSLI Studies in Computational

Linguistics ONLINE, Series Editor: Ann Copestake, pp. 55-64.

Karlsson, F. (2008). Early Generative Linguistics and Empirical Methodology. In A. Lüdeling, and M. Kytö (eds.), *Corpus Linguistics: An International Handbook*, vol. 1, Walter de Gruyter GmbH & Co. KG, Berlin, Germany, pp. 14-32.

Karlsson, F., Voutilainen, A., Heikkila, J., and Anttila, A. (eds.) (1995). *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Kermes, H. (2008). Syntactic Preprocessing. In A. Lüdeling, and M. Kytö (eds.), *Corpus Linguistics: An International Handbook*, vol. 1, Walter de Gruyter GmbH & Co. KG, Berlin, Germany, pp. 598-612.

Khoja, S. (2001a). APT: Arabic Part-of-Speech Tagger, *In Proceedings of the Student Workshop at NAACL-2001*, pp. 20-25.

Khoja, S., Garside, R., and Knowles, G. (2001b). A tagset for the morphosyntactic tagging of Arabic. In *Proceedings of the Corpus Linguistics 2001 Conference*, Lancaster University.

Khoja. S. (2003). *APT: An Automatic Arabic Part-of-Speech Tagger*. Ph.D. thesis, University of Lancaster, UK.

Kilgarriff, A. (2007). Word Senses. In E. Agirre, and P. Edmonds (eds.), *Word Sense Disambiguation: Algorithms and Applications*, Springer, pp. 29-46.

Koehn, P. (2004). Pharaoh: A Beam Search Decoder for Phrase-based Statistical machine Translation Models. In Proceedings of AMTA, Washington, DC.

Koehn, P., and Knight, K. (2002). Learning a Translation Lexicon from Monolingual Corpora. *In Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX),* Philadelphia, pp. 9-16.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Human Language Technology Conference (HLT-NAACL),* Edmonton, Canada, pp. 48–54.

Kübler, S., and Mohamed, E. (2008). Memory-Based Vocalization of Arabic. In *Proceedings of the LREC Workshop on HLT and NLP within the Arabic World*, Marrakesh, Morocco.

Kumano, A., and Hirakawa, H. (1994). Building an MT Dictionary from Parallel Texts Based on Linguistic and Statistical Information. In *Proceedings of the*

15th International Conference on Computational linguistics (COLING'94), Kyoto, Japan, pp. 76-81.

Lager, T. (1999). μ-TBL Lite: A Small, Extendible Transformation-Based Learner. In *Proceedings of the 9th European Conference on Computational Linguistics (EACL-99)*, Bergen, pp. 279-280.

Larkey, L., Ballesteros, L., and Connell, M. (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In *Proceedings of th 25th International Conference on Research and Development (SIGIR)*, Tampere, Finland: ACM, pp. 275-282.

Larkey, L., Ballesteros, L., and Connell, M. (2007). Light Stemming for Arabic Information Retrieval. In A. Soudi, A. Bosch, and G. Neumann (eds.) *Arabic Computational Morphology: Knowledge-based and Empirical Methods,* volume 38 of *Text, Speech and Language Technology,* Springer Verlag, pp. 221-243.

Lee, H. A., park, J. C., and Kim, G. C. (1999) Lexical Selection with a Target Language Monolingual Corpus and an MRD. In *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in MT*, Chester, UK, pp. 150-160.

Lee, H. A., Yoon, J., and Kim, G. C. (2003). Translation Selection by Combining Multiple Measures for Sense Disambiguation and Word Selection. In *the International Journal of Computer Processing of Oriental Languages*, 16 (3), pp. 219-239.

Lee, H. A. (2006). Translation Selection Through Machine Learning with Language Resources. In *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*. Lecture Notes in Computer Science, vol. 4285, Springer-Verlag Berlin Heidelberg, pp. 370-377.

Lee, H.K. (2002). Classification Approach to Word Selection in Machine Translation. In S. D. Richardson (ed.), *Machine Translation: From Research to Real Users,* (AMTA 2002), Springer-Verlag Berlin Heidelberg, pp. 114-123.

Leech, G. (1997). Grammatical Tagging. In R. Garside, G. Leech, and A. McEnery (eds.), *Corpus Annotation: Linguistic Information from Computer Text Corpora,* Longman, London, pp. 19-33.

Lovins, J. B. (1968). Development of a Stemming Algorithm. *In Mechanical Translation and Computational Linguistics*, 11, pp. 22-31.

Lyons, J. (1995). *Linguistic Semantics: An Introduction*. Cambridge: Cambridge University Press.

Lyons, J. (1977). *Semantics*. vol. 2, Cambridge: Cambridge University Press.

Lyons. J. (1968). *Introduction to theoretical linguistics*. Cambridge: Cambridge University Press.

Maamouri, M., Bies , A., and Kulick, S. (2006) Diacritization: A Challenge to Arabic Treebank Annotation and Parsing, In *The Challenge of Arabic for NLP/MT Conference*, The British Computer Society, London, UK, pp. 35-47.

MacDonald, M. C., Pearlmutter, N. J., and Seidenberg, M. S. (1994). Lexical Nature of Syntactic Ambiguity Resolution. In *Psychological Review*, 101 (4), pp. 676-703.

Mace, J. (1998). *Arabic Grammar: A Reference Guide*. Edinburgh University Press.

Majdi, B. (1990). Word Order and Proper Government in Classical Arabic. In M. Eid (ed.) *Perspectives on Arabic Linguistics I: Papers from the First Annual Symposium on Arabic Linguistics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp. 127-153.

Manning, C., and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Mass.; London: MIT Press.

Marcu, D., and Wong, W. (2002). A Phrase-Based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP-2002),* University of Pennsylvania, Philadelphia, PA, pp. 133–139.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. In *Computational Linguistics*, 19 (2), pp. 313-330.

Marsi, E., Bosch, A., and Soudi, A. (2005) Memory-based morphological analysis generation and part-of-speech tagging of Arabic, In *ACL Workshop on Computational Approaches to Semitic Languages*, Michigan, USA.

Matsumoto, Y., Ishimoto, H., and Utsuro, T. (1993). Structural Matching of Parallel Texts. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 23-30.

McCarthy, J. (1981). A Prosodic Theory of Nonconcatenative Morphology. In *Linguistic Inquiry*, 12 (3), pp. 373–418.

McCarthy, J., and Prince, A. (1990). Prosodic Morphology and Templatic Morphology. In M. Eid and J. McCarthy (eds.) *Perspectives on Arabic Linguistics II: Papers from the Second Annual Symposium on Arabic Linguistics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp. 1-54.

McEnery, T. (1992). *Computational Linguistics: a handbook & toolbox for natural language processing*. Sigma Press, England.

Mel'čuk, I. (1979). *Studies in Dependency Syntax*. Karoma Publishers, Inc.

Mel'čuk, I. (1988). *Dependency Syntax: Theory and Practice*. State University Press of New York.

Melamed, I. D. (2000). Models of Translational Equivalence among Words. In *Computational Linguistics*, 26 (2), pp. 221-249.

Melamed, I. D. (1995). Automatic Evaluation and Uniform Filter Cascades for Inducing N-Best Translation Lexicons. In *Proceedings of the 3rd Workshop on Very Large Corpora (WVLC3)*. Boston, MA, U.S.A, pp. 184-198.

Melamed, I. D. (1997). A Word-to-Word Model of Translational Equivalence. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain, pp. 490-497.

Melamed, I. D. (2004). Statistical Machine Translation by Parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, Barcelona, Spain, pp. 653–660.

Melamed, I. D., Green, R., and Turian, J. P. (2003). Precision and Recall of Machine Translation. In *Proceedings of the HLT-NAACL 2003: Short Papers*, Edmonton, Canada, pp. 61–63.

Menezes, A. (2002). Better Contextual Translation Using Machine Learning. In S. D. Richardson (ed.) *Machine Translation: From Research to Real Users* (AMTA 2002), Springer-Verlag Berlin Heidelberg, pp. 124-134.

Miller, G.A., and Charles, W. G. (1991). Contextual Correlates of Semantic Similarity. In *Language and Cognitive Processes*, 6 (1), pp.1-28.

Miller, G.A., and Teibel, D. A. (1991). A Proposal for Lexical Disambiguation. In

Proceedings of DARPA Workshop on Speech and Natural Language, Pacific Grove, California, pp. 395-399.

Mohamed, E., and Kübler, S. (2009). Diacritization for Real-World Arabic Texts. In *Proceedings of RANLP 2009*, Borovets, Bulgaria, pp. 251-257.

Mohamed, E., and Kübler, S. (2010a). Arabic Part of Speech Tagging. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC),* Valetta, Malta, pp. 2537-2543.

Mohamed, E., and Kübler, S. (2010b). Is Arabic Part of Speech Tagging Feasible Without Word Segmentation. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL (NAACL-HLT 2010)*, pp. 705-708.

Mohammad, M.A. (1990). The Problem of Subject-Verb Agreement in Arabic: Towards a Solution. In M. Eid (ed.) *Perspectives on Arabic Linguistics I: Papers from the First Annual Symposium on Arabic Linguistics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp. 95-125.

Mubarak, H., Al Sharqawy, M., and Al Masry, I. (2009a). Diacritization and Transliteration of Proper Nouns from Arabic to English. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools,* 22-23 April 2009, Cairo, Egypt, pp. 256-259.

Mubarak, H., Shaban, K., and Adel, F. (2009b). Lexical and Morphological Statistics of an Arabic POS-Tagged Corpus. In *Proceedings of the 9th Conference on Language Engineering ESOLEC'2009,* 23-24 December 2009, Cairo, Egypt, pp. 147-161.

Nagao, M. (1984). A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In A. Elithorn, and R. Banerji (eds.), *Artificial and Human Intelligence,* Amsterdam: North-Holland, pp. 173–180.

Nelken, R., and Shieber, SM (2005). Arabic Diacritization Using Weighted Finite-State Transducers, In *Proceedings of the 2005 ACL Workshop on Computational Approaches to Semitic Languages*, Michigan, pp.79-86.

Newmark, P. (1988). *Approaches to Translation*. Prentice Hall International (UK) Ltd.

Ney, H. (1997). Corpus-Based Statistical Methods in Speech and Language Processing. In S. Young, and G. Bloothooft (eds.), *Corpus-Based Methods in*

Language and Speech Processing, Kluwer Academic Publishers.

Nirenburg, S., Domashnev, C., and Grannes, D. J. (1993). Two Approaches to Matching in Example-Based Machine Translation. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation TMI '93: MT in the Next Generation*, Kyoto, Japan, pp. 47–57.

Nirenburg, S., Levin, L. (1992). Syntax-Driven and Ontology-Driven Lexical Semantics. In J. Pustejovsky, and S. Bergler (eds.) *Lexical Semantics and Knowledge Representation*, Heidelberg: Springer-Verlag, pp. 5-20.

NIST (2002). Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. *Technical report*.

Nivre, J. (2005). Dependency Grammar and Dependency Parsing. *MSI report 05133*. Växjö University: School of Mathematics and Systems Engineering.

Nivre, J. (2006). *Inductive Dependency Parsing*. Text, Speech and Language Technology. Dordrecht: Springer.

Nivre, J. and Scholz, M. (2004). Deterministic Dependency Parsing of English Text. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pp. 64-70.

Nivre, J., Hall, J., and Nilsson, J. (2006). MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, May 24-26, 2006, Genoa, Italy, pp. 2216-2219.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryiğit, G., Kübler, S., Marinov, S. And Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. In *Natural Language Engineering*, 13(2), Cambridge University Press, pp. 95-135.

Och, F. J., and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. In *Computational Linguistics*, 29 (1), pp. 19-52.

Och, F. J., Tillmann, C. and Ney, H. (1999). Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora*, University of Maryland, College Park, MD, pp. 20–28.

Omar, A., Z., M., Abdellatif, M. (1994). *Al-Naħw Al-Asaasi*. Daar Al-Salaasel lil-nashr, Kuwait.

Palmer, F. R. (1981). *Semantics*. 2nd edition, Cambridge: Cambridge University Press

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02),* Philadelphia, PA, pp. 311–318.

Parkinson, D. (1990). Orthographic Variations in Modern Standard Arabic: The Case of the Hamza. In M. Eid and J. McCarthy (eds.) *Perspectives on Arabic Linguistics II: Papers from the Second Annual Symposium on Arabic Linguistics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp. 269-295.

Pedersen, B.S. (2000). Lexical Ambiguity in Machine Translation: Using Frame Semantics for Expressing Regularities in Polysemy. In N. Nicolov, and R. Mitkov (eds.), *Recent Advances In Natural Language Processing II,* Amsterdam/Philadelphia: J. Benjamins, pp. 207-218.

Piperidis, S. Dimitrakis, P., and Balta, I. (2005). Lexical Transfer Selection Using Annotated Parallel Corpora. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria.

Platzack, C. (2003). Agreement and Null Subjects. In *The 19th Scandinavian Conference of Linguistics*, University of Tromsø, Norway, pp. 326-355.

Platzack, C. (1988). Valency and GB Grammar. In F. Sørensen (ed.), *Valency: Three Studies on the Linking Power of Verbs*, Copenhagen: Forlag Arnold Busck, pp.57-71.

Pollard, C., and Sag, I. (1994). *Head-driven Phrase Structure Grammar*. Center for the Study of Language and Information, University of Chicago Press.

Poole, G. (2002). *Syntactic Theory*. Basingstoke: Palgrave.

Porter, M. F. (1980). An algorithm for suffix stripping. In *Program*, 14 (3), pp. 130-137.

Rambow, O. (2010). The Simple Truth about Dependency and Phrase Structure Representations: An Opinion Piece. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL (NAACL-HLT 2010),* Los Angeles, pp. 337-340

Ramsay, A., and Sabtan, Y. (2009). Bootstrapping a Lexicon-Free Tagger for Arabic.

In *Proceedings of the 9th Conference on Language Engineering ESOLEC'2009*, 23-24 December 2009, Cairo, Egypt, pp. 202-215.

Ramsay, A., and Mansour, H. (2004). The parser from an Arabic text-to-speech system, *Traitement automatique du language naturel* (TALN'04)}, F`es, Morroco.

Ramsay, A., and Mansour, H. (2007). Towards including prosody in a text-to-speech system for modern standard Arabic. In *Computer Speech and Language*, 22, pp. 84–103.

Ratcliffe, R. (1998). *The Broken Plural Problem in Arabic and Comparative Semitic: Allomorphy and Analogy in Non-concatenative Morphology*. Amsterdam studies in the theory and history of linguistic science. Series IV, Current issues in linguistic theory; v. 168. Amsterdam; Philadelphia: J. Benjamins.

Ratcliffe, R. (1990). Arabic Broken Plurals: Arguments for a two-fold classification of morphology. In M. Eid, and J. McCarthy (eds.) *Perspectives on Arabic Linguistics II: Papers from the Second Annual Symposium on Arabic Linguistics*, Amsterdam/Philadelphia: John Benjamins Publishing Company, pp. 94-119.

Reifler, E. (1955). The mechanical determination of meaning. In W. N. Locke, and A. D. Booth (eds.), *Machine Translation of Languages*, John Wiley & Sons, New York, pp. 136-164.

Resnik, P., and Melamed, I. D. (1997). Semi-Automatic Acquisition of Domain-Specific Translation Lexicons. In *Proceedings of the 5th ACL Conference on Applied natural Language Processing*, Stroudsburg, PA, USA, pp. 340-347.

Rimon, M., McCord, M., Schwall, U., and Martinez, P. (1991) Advances in Machine Translation Research in IBM. *In MT Summit III*, Washington, DC, USA, pp. 11-18.

Robinson, J. J., (1967). Methods for Obtaining Corresponding Phrase Structure and Dependency Grammars. In *Proceedings of the 1967 International Conference on Computational linguistics*, USA, pp. 1-25.

Robison, H. R. (1970). Computer-Detectable Semantic Structures. In *Information Storage and Retrieval*, 6, pp. 273-288.

Rouhani, J. (1994). *An Applied Research into the Linguistic Theory of Collocation: English-Arabic Dictionary of Selected Collocations and Figurative Expressions* *with an Arabic Index*. Ph.D. Thesis, University of Exeter, UK.

Ryding, K. C. (2005). *A Reference Grammar of Modern Standard Arabic*. Cambridge: Cambridge University Press.

Saeed, J. I. (2003). *Semantics*. 2nd edition. Malden, Mass.; Oxford: Blackwell Publishing.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword Expressions: A Pain in the Neck for NLP. In A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing*, pp.1-15: Springer.

Saleh, I. M., and Habash, N. (2009). Automatic Extraction of Lemma-based Bilingual Dictionaries for Morphologically Rich Languages. In *the 3rd Workshop on Computational Approaches to Arabic Script-based languages at the MT Summit XII*, Ottawa, Ontario, Canada,

Sato, K., and Saito, H. (2002) Extracting Word Sequence Correspondences with Support Vector Machines. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pp. 870–876

Sawalha, M., and Atwell, E. (2010). Fine-Grain Morphological Analyzer and Part-of-Speech Tagger for Arabic Text. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, 19-21 May 2010, Valletta, Malta, pp. 1258-1265.

Sawalha, M., and Atwell, E. (2009). Linguistically Informed and Corpus Informed Morphological Analysis of Arabic. In *Proceedings of the 5th International Corpus Linguistics Conference CL2009*, 20-23 July 2009, Liverpool, UK.

Schafer, C., and Yarowsky, D. (2003). A Two-Level Syntax-Based Approach to Arabic-English Statistical Machine Translation. In *Proceedings of MT Summit IX -- workshop: Machine translation for Semitic languages*, New Orleans, USA, pp. 45-52.

Schenk, A. (1995). The Syntactic Behavior of Idioms. In M. Everaert, E. Linden, A. Schenk, and R. Schreuder (eds.), *Idioms: Structural and Psychological Perspectives*, New Jersey: Lawrence Erlbaum Associates, pp. 253-271.

Schubert, K., and Maxwell, D. (1989). *Metataxis in Practice: Dependency Syntax for Multilingual Machine Translation*. Dordrecht: Floris.

Smrž, O. (2007*). Functional Arabic Morphology: Formal System and Implementation*. Ph.D. thesis, Charles University in Prague.

Smrž, O. and Hajič, J. (2006). The Other Arabic Treebank: Prague Dependencies and Functions. In A. Farghaly (ed.), *Arabic Computational Linguistics: Current Implementations,* CSLI Publications.

Smrž, O., Bielický, V., Kouřilová, I., Kráčmar, J., Hajič, J., and Petr Zemánek, P. (2008). Prague Arabic Dependency Treebank: A Word on the Million Words. In *Proceedings of the Workshop on Arabic and Local Languages (LREC 2008),* Marrakech, Morocco, pp. 16–23.

Somers, H. (1987). *Valency and Case in Computational Linguistics*, Edinburgh: Edinburgh University Press.

Somers, H. (1999). Review Article: Example-based Machine Translation. In *Machine Translation*, 14: pp.113-157.

Somers, H. (2003a). Sublanguage. In H. Somers (ed.) *Computers and Translation: A Translator's Guide*, Amsterdam: John Benjamins Publishing, pp. 283-295.

Somers, H. (2003b) Machine translation: latest developments. In R. Mitkov (ed.) *The Oxford Handbook of Computational Linguistics*, Oxford: Oxford University Press, pp.512-528.

Somers, H. (2003c). An Overview of EBMT. In M. Carl, and A. Way (eds.), *Recent Advances in Example-Based Machine Translation*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 3–57.

Somers, H., and Diaz, G. F. (2004) Translation Memory vs. Example-based MT: What is the difference? In *International Journal of Translation* 16 (2), pp. 5-33.

Soudi, A., Cavalli-Sforza, V. and Jamari, A. (2001). A Computational Lexeme-Based Treatment of Arabic Morphology. In *Proceedings of the Arabic Natural Language Processing Workshop, Conference of the Association for Computational Linguistics (ACL 2001)*, Toulouse, France.

Stubbs, M. (2002). *Words and Phrases: Corpus Studies of Lexical Semantics*. Oxford: Blackwell.

Swart, H. D. (1998). *Introduction to Natural Language Semantics*. Stanford, Calif.: CSLI Publications.

Taghva, K., Elkhoury, R., Coombs, J. (2005). Arabic Stemming Without A Root Dictionary. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)* - Volume I, IEEE Computer

Society, Washington, DC, USA, pp. 152 - 157

Tapanainen, P., Järvinen, T. (1997). A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington, D.C., pp. 64-71.

Tesnière, L. (1959) *El´ements de Syntaxe Structurale*. Paris: C. Klincksieck.

Thabet, N. (2004). Stemming the Qur'an. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-Based Languages, COLING-04,* Geneva, Switzerland, pp. 85-88.

Tiedemann, J. (1998) Extraction of Translation Equivalents from Parallel Corpora. In *Proceedings of the 11th Conference on Computational Linguistics*, Copenhagen, Denmark.

Tlili-Guiassa, Y. (2006). Hybrid Method for Tagging Arabic Text. In *Journal of Computer Science* 2 (3): pp. 245-248.

Tounsi, L. Attia, M., van Genabith, J. (2009). Automatic Treebank-Based Acquisition of Arabic LFG Dependency Structures. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, Athens, Greece, pp. 45-52.

Trujillo, A. (1999). *Translation Engines: Techniques for Machine Translation*. London: Springer.

Tufi¸s, D. and Barbu, A.M. (2001a). Automatic Construction of Translation Lexicons. In V.V. Kluev, C.E.D'Attellis, and N.E. Mastorakis (eds.), *Advances in Automation, Multimedia and Video Systems, and Modern Computer Science*, Electrical and Computer Engineering Series, WSES Press, pp. 156–161.

Tufi¸s, D. and Barbu, A.M. (2001b). Accurate Automatic Extraction of Translation Equivalents from Parallel Corpora. In P. Rayson, A. Wilson, T. McEnery, A. Hardie, and S. Khoja (eds.), In *Proceedings of the Corpus Linguistics 2001 Conference*. Lancaster: Lancaster University, pp. 581–586.

Tufi¸s, D. and Barbu, A.M. (2002). Revealing Translators' Knowledge: Statistical Methods in Constructing Practical Translation Lexicons for Language and Speech Processing. In *the International Journal of Speech Technology*, 5 (3), Kluwer Academic Publishers, The Netherlands, pp.199-209.

Turian, J. P., Shen, L., and Melamed, I. D. (2003). Evaluation of Machine

Translation and its Evaluation. In *Proceedings of MT Summit IX*, New Orleans, LA, pp. 386-393.

Uí Dhonnchadha, E. (2008). *Part-of-Speech Tagging and Partial Parsing for Irish Using Finite-State Transducers and Constraint Grammar*. Ph.D. Thesis, Dublin City University, Ireland.

Van den Bosch, A., Marsi, E., and Soudi, A. (2007). Memory-based Morphological Analysis and Part-of-speech Tagging of Arabic. In A. Soudi, A. Van den Bosch, and G. Neumann (eds.), *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, Springer, pp. 201-217.

Venkatapathy, S., and Bangalore, S. (2009). Discriminative Machine Translation Using Global Lexical Selection. In ACM Transactions on Asian Language Information Processing (TALIP), 8 (2).

Voutilainen, A. (1999a). Orientation. In H. Halteren (ed.), *Syntactic Wordclass Tagging*, Kluwer Academic Publishers, pp. 3-7.

Voutilainen, A. (1999b). A Short History of Tagging. In H. Halteren (ed.), *Syntactic Wordclass Tagging*, Kluwer Academic Publishers, pp. 9-21.

Weaver, W. (1949). Recent Contributions to the Mathematical Theory of Communication. In C. E. Shannon, and W. Weaver (eds.) *The Mathematical Theory of Communication*, The University of Illinois Press, Urbana, IL, pp. 94-117.

Willett, P. (2006). *The Porter stemming algorithm: then and now*, Program: electronic library and information systems, 40 (3), pp. 219-223.

Wright, W. (1967). *A Grammar of the Arabic Language*. Cambridge: Cambridge University Press.

Wu, Y., Zhang, Q., Huang, X. and Wu, L. (2009). Phrase Dependency Parsing for Opinion Mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, 6-7 August 2009, pp. 1533–1541.

Wu, Z. and Palmer, M. (1994). Verb Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, pp. 133-138.

Xia, F. and Palmer, M. (2001). Converting Dependency Structures to Phrase Structures. In *Proceedings of the Human Language Technology Conference*

(HLT-2001), San Diego, CA.

Yamada, K. and Knight, K. (2001). A Syntax-Based Statistical Translation Model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01),* Toulouse, France, pp. 523–530.

Yu, K., and Tsujii, J. (2009). Extracting Bilingual Dictionary from Comparable Corpora with Dependency Heterogeneity. In *Proceedings of NAACL HLT 2009: Short Papers*, Boulder, Colorado, pp. 121-124.

Žabokrtský, Z. Smrž, O. (2003). Arabic Syntactic Trees: from Constituency to Dependency. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03) – Research Notes*, Budapest, Hungary.