Pivot-based Statistical Machine Translation for Morphologically Rich Languages

Ahmed El Kholy

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2016

ABSTRACT

Pivot-based Statistical Machine Translation for Morphologically Rich Languages

Ahmed El Kholy

This thesis describes the research efforts on pivot-based statistical machine translation (SMT) for morphologically rich languages (MRL). We provide a framework to translate to and from morphologically rich languages especially in the context of having little or no parallel corpora between the source and the target languages. We basically address three main challenges. The first one is the sparsity of data as a result of morphological richness. The second one is maximizing the precision and recall of the pivoting process itself. And the last one is making use of any parallel data between the source and the target languages.

To address the challenge of data sparsity, we explored a space of tokenization schemes and normalization options. We also examined a set of six detokenization techniques to evaluate detokenized and orthographically corrected (enriched) output. We provide a recipe of the best settings to translate to one of the most challenging languages, namely Arabic. Our best model improves the translation quality over the baseline by $\approx$1.3 BLEU points.

We also investigated the idea of separation between translation and morphology generation. We compared three methods of modeling morphological features. Features can be modeled as part of the core translation. Alternatively these features can be generated using target monolingual context. Finally, the features can be predicted using both source and target information. In our experimental results, we outperform the vanilla factored translation model.

In order to decide on which features to translate, generate or predict, a detailed error analysis should be provided on the system output. As a result, we present AMEANA, an open-source tool for error analysis of natural language processing tasks, targeting morphologically rich languages.

The second challenge we are concerned with is the pivoting process itself. We discuss several techniques to improve the precision and recall of the pivot matching. One technique to improve the

recall works on the level of the word alignment as an optimization process for pivoting driven by generating phrase pairs between source and target languages. Despite the fact that improving the recall of the pivot matching improves the overall translation quality, we also need to increase the precision of the pivot quality. To achieve this, we introduce quality constraints scores to determine the quality of the pivot phrase pairs between source and target languages. We show positive results for different language pairs which shows the consistency of our approaches. In one of our best models we reach an improvement of 1.2 BLEU points.

The third challenge we are concerned with is how to make use of any parallel data between the source and the target languages. We build on the approach of improving the precision of the pivoting process and the methods of combination between the pivot system and the direct system built from the parallel data.

In one of the approaches, we introduce morphology constraint scores which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. We compare two methods of generating the morphology constraints. One method is based on hand-crafted rules relying on our knowledge of the source and target languages; while in the other method, the morphology constraints are induced from available parallel data between the source and target languages which we also use to build a direct translation model. We then combine both the pivot and direct models to achieve better coverage and overall translation quality. Using induced morphology constraints outperformed the handcrafted rules and improved over our best model from all previous approaches by 0.6 BLEU points (7.2/6.7 BLEU points from the direct and pivot baselines respectively). Finally, we introduce applying smart techniques to combine pivot and direct models. We show that smart selective combination can lead to a large reduction of the pivot model without affecting the performance and in some cases improving it.

Table of Contents

List of Figures

List of Tables

Acknowledgments

After a long journey and years of hard work to complete my thesis, I would like to express my gratitude to all the people who supported me and gave me guidance through out the years.

First, I would like to express my gratitude to my advisor Nizar Habash. He has been a great advisor and mentor in life. His encouragement and continuous support is tremendous. Nizar is always keen to transfer all the knowledge he possess to his students and his door is always open to help with any issue. I would like also to thank the other members of the committee Kathleen McKeown, Michael Collins, Rebecca Passonneau, Smaranda Muresan and Alon Lavie for being part of this thesis and devoting time to review my dissertation document.

I would like to thank all the people at the Center for Computing Learning Systems (CCLS) for being great colleagues and friends. It was a pleasure being part of the group and special thanks to Mona Diab, Owen Rambow, Hatim Diab, Ryan Roth, Ramy Eskander, Mahmoud Ghoneim, Yassine Benajiba, Marine Carpuat, Fadi Biadsy and Mohamed Albadrashiny. I spent great time with the students at CCLS and I feel lucky being among nice and smart people like Wael Salloum, Sarah Alkuhlani, Heba ElFardy, Boyi Xie, Weiwei Guo, Vinod Kumar, Daniel Bauer, Apoorv Agarwal, Noura Farra and Mohammad Sadegh Rasooli. I hope we stay in contact and I wish them all the best in their life and career. Last but not least, I would like to thank my best friend, Mohamed Altantawy. We have known each other since undergraduate days and he is more than a brother to me. He has been always there in good and bad times and I'm grateful that I met such a person in my life.

Finally, I would like to express my gratitude to all my family and friends. I have been lucky to receive an amazing support from my dear sister Eman and several members in my family. Their support and encouragement has been vital in my overcoming several hurdles in life.

To my mother who is not here today to witness the completion of this thesis.
I owe every bit of my existence to her and this thesis is dedicated to her memory.
To my father for his ongoing support and care. He is my inspiration when it comes to overcoming times of hardship and working hard to achieve life goals.

Chapter 1

Introduction

One of the main issues in statistical machine translation (SMT) is data sparsity which is mainly a result of the shortage of parallel data for many language pairs. Morphologically rich languages (MRL), in particular, are more challenging. This is due, in part, to two phenomena. The first one is morphological richness. Words sharing the same core meaning (represented by the word lemma or lexeme) can be said to inflect different morphological features such as gender and number. These features can be shown by using concatenative (affixes and stems) and/or templatic (root and patterns) morphology. The second challenge is morphological ambiguity. Words with different lemmas can have the same inflected form. As such, a word form can have more than one morphological analysis (represented as a lemma and a set of feature-value pairs). This is especially problematic for languages with reduced orthographies such as Arabic or Hebrew. These two phenomena lead to more sparsity in data in comparison to a morphologically poor language given the same corpora size.

A common solution in the field is to pivot the translation through a third language (called pivot or bridge language) for which there exists abundant parallel corpora with the source and target languages. The literature covers many pivoting techniques. One of the best performing techniques, phrase pivoting [Utiyama and Isahara, 2007], builds an induced new phrase table between the source and the target. One of the problems of this technique is that the size of the newly created pivot phrase table is very large [Utiyama and Isahara, 2007].

In this thesis, we provide a pivoting framework to translate to and from MRL especially in the context of having little or no parallel corpora between the source and the target languages. We basically address three main challenges. The first and main challenge is sparsity of data. The second one is maximizing precision and recall of the pivoting process. The last one is making use of any parallel data between the source and the target languages.

In general, our discussed solutions can be applied to any MRL and many of our approaches are language independent, but some still requires linguistic knowledge and the availability of morphological analyzers for a given language. However, the techniques discussed can be easily adapted to any language. In most of our work, we target Arabic as it is one of the most challenging languages in the field, but we work with other languages specifically Persian and Hebrew.

1.1 Approach

To address the first challenge of data sparsity, we explore a space of tokenization schemes and normalization options with their implications on the quality of MT. Regardless of the preprocessing choices, the Arabic output is detokenized and denormalized. Anything else is comparable to producing all lower cased English or uncliticized and undiacritized French. Detokenization is not a simple task because there are several morphological adjustments that apply in the process. We examine different detokenization techniques for various tokenization schemes.

In another direction, we address these challenges through different modeling methods. In our approach, morphological features can be modeled as part of the core translation process mapping source tokens to target tokens. Alternatively, these features can be generated using target monolingual context as part of a separate generation (or post-translation inflection) step. Finally, the features can be predicted using both source and target information in a separate step before generation.

In order to help decide which features to translate, to generate or to predict, we present AMEANA (Automatic Morphological Error Analysis), an automatic error analysis tool that is designed to identify morphological errors in the output of a given system against a gold reference. AMEANA produces detailed statistics on morphological errors in the output. It also generates an oracularly modified version of the output that can be used to measure the effect of these errors using any evaluation metric. AMEANA is a language independent tool except that a morphological analyzer must be provided for a given language.

The second challenge that we are concerned with is the pivoting process itself. In the standard phrase-pivoting approach, many phrase pairs between source and target languages are not generated because of the bad matching of pivot phrases. However, the size of the newly created pivot phrase table is very large. In addition, many of the produced phrase pairs are of low quality which affects the translation choices during decoding and the overall translation quality.

We try to maximize both precision and recall of the pivoting process, and we discuss several techniques to improve the recall of the pivot matching. One of the techniques works on the level of the word alignment symmetrization. Like the common heuristics for symmetrization, we aim to find a balance between the intersection and union. But unlike the state of the art heuristics, symmetrization is carried out as an optimization process driven by the effectiveness of each alignment pair with respect to pivoting, and add or remove the word links that can maximize the pivoting process.

Despite the fact that we miss a lot of matches in pivoting and that we try to improve the recall, we also need to consider the quality precision of phrase pivoting. One of the manifestations of phrase pivoting is that the size of the newly created pivot phrase table is very large [Utiyama and Isahara, 2007]. Besides, many of the produced phrase pairs are of low quality which affects the translation choices during the decoding and the overall translation quality. We discuss different techniques to determine the quality of the pivot phrase pairs between the source and the target. In one of the language independent approaches, we generate different connectivity scores between the source and target phrase pairs based on the alignment information propagated from the source-pivot and pivot-target systems.

The third challenge we are concerned with is how to make use of any parallel data between the source and the target languages. We discuss different approaches to improve the pivot SMT system and methods of the combination between the pivot system and the direct system built from the parallel data.

In one of the approaches, we introduce morphology constraint scores which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. This morphology constraint scores are based on the connectivity scores. We compare two methods of generating the morphology constraint scores. One method is based on hand-crafted rules relying on our knowledge of the source and target languages; while in the other method, the morphology constraints are induced from available parallel data between the source and target languages which we also use to build a direct translation model. We then combine both the pivot and direct models to achieve better coverage and overall translation quality.

We also discuss applying smart techniques to combine pivot and direct models. We aim at having a better coverage and overall translation quality. The combination approach needs to be optimized in order to maximize the information gain. We maximize the information gain by selecting the relevant portions of the pivot model that do not interfere with the direct model which is in principal trusted more.

1.2 Contributions

In our research contribution, we are interested in improving the Pivot-based Statistical Machine Translation for Morphologically Rich Languages with limited resources.

Our discussed efforts will work on the pivoting framework of constructing two separated SMT systems, Source-Pivot SMT and Pivot-Target; and then perform phrase-pivoting. We discuss several methods to improve each component separately, and also discuss methods to improve the system as the whole targeting of the final pivot SMT system.

The first challenge we are concerned with is the sparsity of data. The following is a list of our approaches to solve this challenge.

- **Morphological Processing:** Explore a space of tokenization schemes and normalization options. We also examine a set of six detokenization techniques to evaluate the detokenized and orthographically corrected (enriched) output.
- **Separation between Translation and Morphology Generation:** We compare three methods of modeling morphological Features that can be modeled as part of the core translation process generated or predicted.
- **Automatic Error Analysis for Morphologically Rich Languages:** We present AMEANA an open-source tool for error analysis of natural language processing tasks targeting MRLs.

The second challenge we are concerned with is the pivoting process itself. We try to maximize both precision and recall of the pivoting process through the following approaches.

- **Pivoting Recall Maximization:** We discuss techniques to improve the recall of the pivot matching by improving the alignment symmetrization method. Symmetrization is carried out as an optimization process driven by the effectiveness of each alignment pair with respect to pivoting, and add or remove the word links that can maximize the pivoting process.

- **Pivoting Quality Maximization:** We discuss different techniques to determine the quality of the pivot phrase pairs between source and target. Next we generate different connectivity scores between the source and target phrase pairs based on the alignment information propagated from the source-pivot and pivot-target systems.

The third challenge we are concerned with is how to make use of any parallel data between the source and target languages. We discuss different approaches to improve the pivot SMT system and methods of combination between the pivot system and the direct system built from the parallel data.

- **Morpho-syntactic Constraints** In this approach, we discuss morpho-syntactic constraints between the source and target languages. We compare two methods of generating the morpho-syntactic constraints. One method is based on hand-crafted rules relying on our knowledge of the source and target languages. In the other method, the morphology constraints are induced from available parallel data between the source and target languages

- **Combination of Pivot and Direct Models:** We discuss applying smart techniques to combine pivot and direct models. We maximize the information gain by selecting the relevant portions of the pivot model that do not interfere with the direct model which is in principal trusted more.

1.3 Thesis Outline

This thesis is organized as follows. In Chapter 2, we give background information on the different languages explored in this thesis; in addition to background information on phrase-based SMT in general and phrase-based Pivot SMT in specific which is the center topic of this thesis.

In Chapter 3, we discuss our work on one of the main components of performing phrase pivoting which is the direct translation to a morphologically rich language (MRL). The main focus of this chapter is to address the challenge of data sparsity due richness in morphology. Most of our discussed and implemented approaches are focusing on Arabic as it is one of the most challenging languages in the field. Then in Chapter 4, we present AMEANA (Automatic Morphological Error

Analysis), AMEANA produces detailed statistics on morphological errors in the output. It also generates an oracularly modified version of the output that can be used to measure the effect of these errors using any evaluation metric.

In Chapter 5 we address morphology richness challenges through different modeling methods. We show how morphological features can be generated using target monolingual context as part of a separate generation (or post-translation inflection) step. Alternatively, the features can be predicted using both source and target information in a separate step before generation.

Starting Chapter 6, we move to the bigger context of phrase pivoting and discuss different approaches to improve the precision and recall of the pivot matching process. All the approaches discussed in this chapter are language independent. In Chapter 7 we explore the space of using linguistic information and make use of any parallel data between the source and target languages to improve the quality of the pivot translation model. Finally, we summarize our contributions and discuss directions for future work in Chapter 8.

Chapter 2

Background

2.1 Linguistic Background

In this section, we discuss the linguistic aspects of three languages that we worked with through out my dissertation (Arabic, Hebrew and Persian). These languages are considered morphologically rich and each pair share some aspects that differ from the third. This section shows why working with these languages is a challenge and a motivation to our approaches in the following chapters.

2.1.1 Arabic

In this thesis, we focus on Modern Standard Arabic (MSA) which is the standard language of the media, education and formal culture in the Arab world. We present relevant aspects of Arabic word orthography and morphology. See [Habash, 2010] for additional computational and non-computational linguistic aspects of the Arabic language.

2.1.1.1 Arabic Orthography

There are two main challenges to Arabic orthography

Spelling Inconsistency: Certain letters in Arabic script are often spelled inconsistently which leads to an increase in both sparsity (multiple forms of the same word) and ambiguity (same form corresponding to multiple words). In particular, variants of Hamzated Alif, أ Â[1] or إ Ǎ, are often written without their Hamza (ء '): ا *A*; and the Alif-Maqsura (or dotless Ya) ى ý and the regular dotted Ya ي *y* are often used interchangeably in word final position. This inconsistent variation in raw Arabic text is typically addressed in Arabic NLP through what is called orthographic normalization, a reductive process that converts all Hamzated Alif forms (including Alif Madda آ Ā) to bare Alif and dotless Ya/Alif Maqsura form to dotted Ya. This kind of normalization is referred to as a Reduced normalization (RED). RED normalization is contrasted with Enriched normalization (ENR), which selects the appropriate form of the Alif and Ya in context [El Kholy and Habash, 2010b]. ENR Arabic is optimally the desired form of Arabic to generate and to evaluate against. Comparing a manually enriched (ENR) version of the Penn Arabic Treebank (PATB) [Maamouri *et al.*, 2004a] to its reduced (RED) version, we find that 16.2% of the words are different. However, the raw (naturally unnormalized) version of the PATB is only different in 7.4% of the words. This suggests a major problem in the recall of the correct ENR form in raw text. In internal experiments, we noticed that BLEU-4 [Papineni *et al.*, 2002a] scores drop about 10 % absolute when comparing ENR to raw (as opposed to ENR) and about 5 % when comparing RED to raw (as opposed to RED) for the same output. As such we only evaluate results against references with their matching normalization condition (ENR or RED).

Optional Diacritics: Another orthographic issue is the optionality of diacritics in Arabic script. In particular, the absence of the Shadda diacritic (ـّ ∼) which indicates a doubling of the consonant it follows leads to a different number of letters in the tokenized and untokenized word forms (when the tokenization happens to split the two doubled consonants). For example, the tokens sequence قاضي+ي *qADy+y* ‘my judge’ is detokenized to قاضيّ *qADy*. Consequently, the detokenization task for such cases is not a simple string concatenation.

2.1.1.2 Arabic Morphology

Arabic is a morphologically complex language with a large set of morphological features producing a large number of rich word forms. While the number of (morphologically untokenized) Arabic

[1] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter transliteration scheme [Habash *et al.*, 2007].

words in a parallel corpus is 20% less than the number of corresponding English words, the number of unique Arabic word types is over twice the number of unique English word types over the same corpus size.

One aspect of Arabic that contributes to this complexity is its various attachable clitics. We define three degrees of cliticization that are applicable in a strict order to a word base [Habash and Sadat, 2006]:

`[cnj+ [prt+ [art+ BASE +pron]]]` [2]

At the deepest level, the BASE can have either the definite article (ال+ *Al*+ ‘the’) or a member of the class of pronominal enclitics, +pron, (e.g., +هم +*hm* ‘their/them’). Next comes the class of particle proclitics (prt+), e.g., ل+ *l*+ ‘to/for’. At the shallowest level of attachment we find the conjunction proclitic (cnj+), e.g., و+ *w*+ ‘and’. The attachment of clitics to word forms is not a simple concatenation process. There are several orthographic and morphological adjustment rules that are applied to the word. An almost complete list of these rules relevant to this article are presented and exemplified in Table 2.1.

It is important to make the distinction here between simple word segmentation, which splits off word substrings with no orthographic/morphological adjustments, and morphological tokenization, which does. Although segmentation by itself can have important advantages, it leads to the creation of inconsistent or ambiguous word forms: consider the words مكتبة *mktbħ* ‘a library’ and مكتبتهم *mktbthm* ‘their library’. A simple segmentation of the second word creates the non-word string مكتبت *mktbt*; however, applying adjustment rules as part of the tokenization generates the same form of the basic word in the two cases. See example of Ta-Marbuta rule in Table 2.1. For more details, see [Habash, 2007; Habash, 2010].

2.1.2 Hebrew

Similar to Arabic, Hebrew poses computational processing challenges typical of Semitic languages [Itai and Wintner, 2008; Shilon *et al.*, 2012; Habash, 2010]. In this section we briefly present relevant aspects of Hebrew word orthography and morphology. Hebrew orthography uses optional diacritics and its morphology uses both root-pattern and affixational mechanisms. Hebrew inflects for

[2] The ’+’ is a marker for the attachable clitics.

gender, number, person, state, tense and definiteness. Furthermore, Hebrew has a set of attachable clitics that are typically separate words in English, e.g., conjunctions (such as +ו *w*+ ‘and’),[3] prepositions (such as +ב *b*+ ‘in’), the definite article (+ה *h*+ ‘the’), or pronouns (such as הם+ +*hm* ‘their’). These issues contribute to a high degree of ambiguity that is a challenge to translation from Hebrew to English or to any other language.

2.1.3 Persian

Unlike Arabic and Hebrew, Persian comes from the Indo-European family and has a relatively simple nominal system. There is no case system and words do not inflect with gender except for a few animate Arabic loanwords. Unlike Arabic, Persian shows only two values for number, just singular and plural (no dual), which are usually marked by either the suffix ها+ +*hA* and sometimes ان+ +*An*, or one of the Arabic plural markers. Persian also possess a closed set of few broken plurals loaned from Arabic. Further, unlike Arabic which expresses definiteness, Persian expresses indefiniteness with an enclitic article ي+ +*y* ‘a/an’ which doesn’t have separate forms for singular and plural. When a noun is modified by one or more adjective, the indefinite article is attached to the last adjective. Persian adjectives are similar to English in expressing comparative and superlative constructions just by adding suffixes تر+ +*tar* ‘+er’ and ترين+ +*taryn* ‘+est’ respectively. Verbal morphology is very complex in Persian. Each verb has a past and present root and many verbs have attached prefix that is regarded part of the root. A verb in Persian inflects for 14 different tense, mood, aspect, person, number and voice combination values [Rasooli *et al.*, 2013].

2.1.4 Summary

We have discussed linguistic aspects of the main three languages that we worked with throughout my dissertation (Arabic, Hebrew and Persian). Table 2.2 summarizes and compares the different aspect of the three languages in addition to English in a nutshell.

[3] The following Hebrew 1-to-1 transliteration is used (in Hebrew lexicographic order): *abgdhwzxTiklmns‘pcqršt*. All examples are undiacritized and final forms are not distinguished from non-final forms.

2.2 Overview of Statistical Machine Translation

Machine Translation (MT) is one of the challenging tasks of NLP addressing translation from one language to another using computational modeling. We have seen recently a lot of progress in the field of machine translation. However, there is still a big doubt about the feasibility of having “fully automatic, high quality machine translation” [Bar-Hillel, 1964] especially when started to better understand the limits of automatic translation [Madsen, 2009].

In recent years, machine translation has been dominated by statistical approaches. This could be attributed on one hand to the fact that the world became more open and there is an increasing demand for better translation services. On the other hand, the rapid development in hardware and computing power makes it possible to benefit from the available data; for example, the UN data. Moreover, there is a growing body of development open source SMT toolkits which facilitate the implementation and the evaluation of translation systems.

Like many NLP tasks, translation is a process involving different factors. The typical approach for any translation model is to tackle each factor individually, and to model their interactions. Generally, translating any body of text requires segmenting it into smaller text units, then translating them atomically and recombining their translations afterward. Statistical approaches aim to learn such segmentation or tokenization in our case in this dissertation, translation and recombination decisions by learning them from a large collections of previously translated texts. The first step to learn these factors starts by learning word alignment which extract the the hidden relations between words from different languages.

In this chapter we give a brief introduction to the Phrase-based Statistical Machine Translation (PBSMT), in which we have performed our experiments and is considered a main component of phrase pivoting presented in this dissertation. For more details on phrase-based SMT and for overviews of other approaches one can refer to several references or books covering SMT [Knight and Marcu, 2005; Lopez, 2008; Koehn, 2010]; and related fundamental research in NLP [Manning and Schütze, 1999; Jurafsky and Martin, 2008], artificial intelligence [Russell and Norvig, 2009], and machine learning for NLP [Smith, 2011], and formal language theory [Hopcroft *et al.*, 2006].

2.2.1 Phrase-Based Translation Model

Phrase-based models translate several contiguous word tokens as an atomic unit, called a phrase[4]. Phrases pairs that are translation of one another are stored in a table structure referred to as the *phrase table*.

The first SMT systems were word-based [Brown *et al.*, 1993b] meaning that they used words as the units of translation. However, shifting from words to phrases has a lot of advantages and gives a lot of context. Typically the translation process requires word disambiguation and word reordering. Working on the phrase level allow us to model those things in one step. For example, the Arabic word "وسيكتبونها" which translates into a whole phrase in English "and they will write it". The word-based model will have to map one word in Arabic to five in English which is what we call the fertility of the word. This process involve many decisions that can be avoided in a phrase-based model which can perform the translation directly in one step. Larger context also helps in dealing with lexical ambiguity. Moreover, along the same lines comes the idea of translating idiomatic expressions and non-compositional phrases.

According to the phrase-based model [Zens *et al.*, 2002; Koehn *et al.*, 2003; Och and Ney, 2004], translation is performed in three steps that can be implemented by a cascade of finite state transducers (FST) [Kumar *et al.*, 2006]: a **segmentation** step, where the source sentence is first split into disjoint contiguous phrases; a **lexical translation** step, in which each source phrase is translated; and finally a **reordering** step, in which target phrases are rearranged into their final order.

One of the open source toolkits for Phrase-based translation is Moses [Koehn *et al.*, 2007a][5]. Most of our experiments in this thesis depend on this toolkit. Many variants of the phrase-based

[4] In this context, the term "phrase" has no specific linguistic meaning.

[5] The Moses toolkit is available at `http://www.statmt.org/moses`.

model have been investigated in the literature. [Och and Ney, 2004] present an alignment template approach that model word reordering based on their part-of-speech categories. [Mariño *et al.*, 2006] refer to phrase pairs as tuples and estimate the translation model as *n-gram* distributions over tuples. Other phrase-based variants [Simard *et al.*, 2005; Crego and Yvon, 2009; Galley and Manning, 2010] offer the possibility for phrases to contain gaps that are filled with other phrases during decoding.

While phrase-based models produce better results than the word-based models, they still have issues with the modeling of reordering. Long-distance reordering is complicated, and distinguishing correct reordering patterns is not an easy task. Incorporating syntax constraints is essential in these cases. Hierarchical and synchronous context-free grammar models use more expressive approaches to handle these cases which belongs to the class of context-free grammar (CFG). They are based on linguistic representation of syntax which help them better modeling long-distance reorderings.

2.2.2 Modeling and Parameter Estimation

In most translation equivalence models and specifically phrase based models, it makes it possible to enumerate all structural relationships between pairs of strings. However, the ambiguity of natural language results in a very large number of possible target sentences for any input source sentence. We will show later when we discuss pivoting that this become more severe when we pivot through a third language. As in typical statistical decisions problems, we are given an input sentence f, and the goal is to find the best translation e. Given different target hypotheses for a given source input, we have to rank those hypotheses and assign a real-valued score.

To approach this problem, we can think of a function $\omega : \Sigma^* \times \Lambda^* \rightarrow \mathbb{R}$ that maps input and output pairs in a real-valued score, is used to rank possible outputs. Given an appropriate parameterization, this scoring function can be interpreted as the conditional probability $p(e|f)$ where $e = (e_1, \ldots, e_T)$ and $f = (f_1, \ldots, f_S)$ are represented with random variables.

In the finite state model of translation, each sentence e can be derived from f in several ways according the alignment $\mathbf{d}$ established between source and target words or segments. The value of $p(e|f)$ is therefore obtained by summing the probabilities of all derivations $\mathbf{d} \in \mathcal{D}$ that yield e.

$$p(e|f) = \sum_{\mathbf{d} \in \mathcal{D}} p(e, \mathbf{d}|f). \tag{2.1}$$

However, this sum involves an exponential number of terms and hence, a common practice is to resort to directly maximizing the function $p(e, \mathbf{d}|f)$. The parameters of $p(e, \mathbf{d}|f)$ are estimated from a parallel corpus using machine learning techniques.

2.2.2.1 Translation Models

We focus in this section on discriminative translation models. They are more suitable for translation prediction because there is not need to model the source sentence which is always given. One of the most popular approaches in SMT is to use a linear model [Berger *et al.*, 1996; Och and Ney, 2002], as in Equation (2.2):

$$p(e, \mathbf{d}|f) = Z(f, \lambda)^{-1} \exp \sum_{k=1}^{K} \lambda_k h_k(e, \mathbf{d}, f), \tag{2.2}$$

where $\{\lambda\}_1^K$ are the scaling factors, associated to the feature functions $\{h\}_1^K$, and $Z(f, \lambda) = \sum_{e,\mathbf{d}} \exp \sum_{k=1}^{K} \lambda_k h_k(e, \mathbf{d}, f)$ is a normalization factor required only to make the scoring function a well-formed probability distribution.

2.2.2.2 Phrase Table Induction

The hypotheses translations for a given input sentence are constructed from preconstructed set of phrase pairs, which sometime called the *bilexicon*. These phrase pairs set is built from a sentence-aligned parallel corpus in one of two ways. Typically, a general phrase alignment is computed for each sentence pair , and the extracted phrase pairs are accumulated over the entire corpus. This method performs very well in practice and is used in most state-of-the-art translation systems.

The Phrase table is a data structure that is widely used in phrase-based systems. This structure contains all the phrase pairs included in the bilexicon. All the features used by the model are precomputed and stored in the phrase table as well. Basically a phrase table can be summarized as a data structure that represents each source phrase along with each possible translation and the associated parameter values. The pipeline used to build the phrase table is pictured in Figure 2.1.

As we mentioned, for each phrase pair in the phrase table, a set of feature functions are computed and used to score translation hypotheses. We discuss the state of the art used feature functions in Section 2.2.2.3.

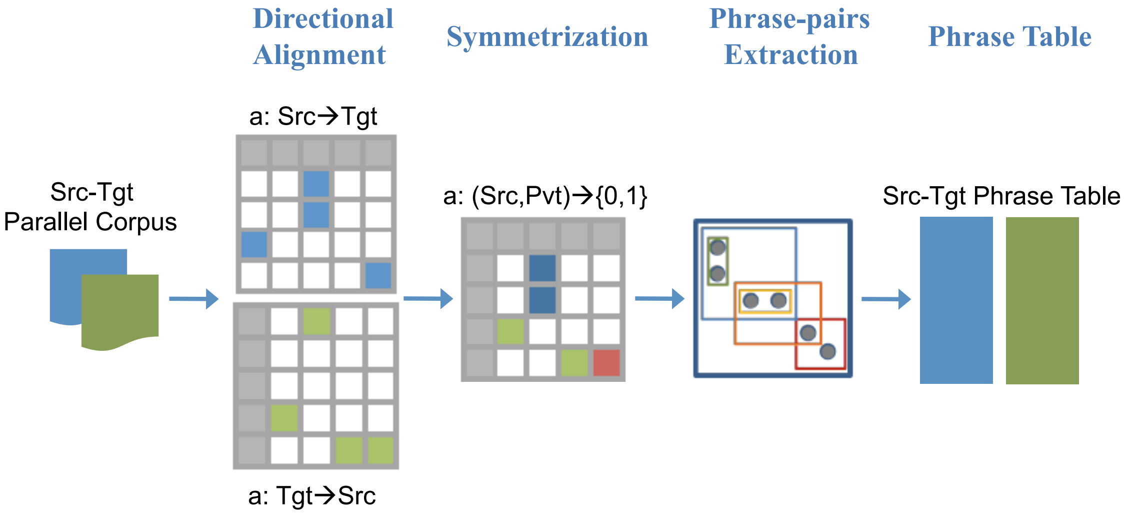


Figure 2.1: The pipeline to construct the phrase table.

2.2.2.3 Features

A feature can be any function from that maps a pair of source and target sentences to a non-negative score value. Each feature function can typically be unraveled in terms of local evaluations at the level of words and also the phrase level. First, we briefly describe the standard features introduced in [Koehn *et al.*, 2007a] and found in other approaches [Simard *et al.*, 2005; Chiang, 2005]. Global features are computed from the entire derivation or decoding process which includes:

- **Distortion:** The number of source words between two source phrases translated into consecutive target phrases.

- **Phrase penalty:** The number of phrase pairs used in the derivation $|\mathcal{D}|$.

- **Word penalty:** The number of produced target words, which controls the length of translation.

The Other features use a limited context around the individual phrase pairs:

- **Language model:** The logarithm of an n-gram target language model

$$\log p(e) = \log \prod_{j=1}^{T} p(e_j | e_{j-1} \dots e_{j-n}), \tag{2.3}$$

which requires keeping a history of n words for each position in the target sentence.

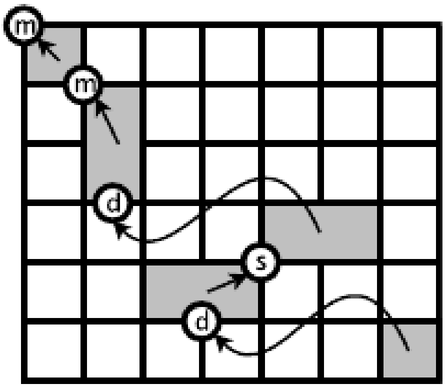


Figure 2.2: Phrase orientations in a lexicalized reordering model

The remaining features are based on each individual phrase pairs. These features include phrase translation probabilities, lexical weighting and lexical reordering.

- **Translation probabilities:** The conditional translation probability of the target phrase given the source phrase:

  $$\log \prod_{(\mathbf{t},\mathbf{s})\in\mathbf{d}} p(\mathbf{t}|\mathbf{s}), \tag{2.4}$$

  where $\mathbf{s}$ is a source phrase and $\mathbf{t}$ is a target phrase. The equivalent phrase probability for the same phrase pairs in the opposite direction $p(\mathbf{s}|\mathbf{t})$ is also computed. It follows the noisy channel approach that was proved in practice to produce a performance comparable to the direct probability $p(\mathbf{s}|\mathbf{t})$ [Och *et al.*, 1999].

  The estimation of the individual probabilities vary along with the phrase alignment model used to build the phrase table.

  $$p(\mathbf{s}|\mathbf{t}) = \frac{\text{count}(\mathbf{s},\mathbf{t})}{\text{count}(\mathbf{t})} \tag{2.5}$$

  The numerator represents the number of the joint occurrences of both phrases aligned together $(\mathbf{s},\mathbf{t})$, while the denominator represents the marginal counts of the phrase $\mathbf{t}$. $p(\mathbf{s}|\mathbf{t})$ is defined similarly.

- **Lexical weighting:** Translation probabilities that is based on relative frequency estimation between phrase pairs are always rough to depend on due do data sparsity. We use Lexical weighting as a smoothing method for infrequent phrase pairs, the probabilities of which are poorly estimated [Foster *et al.*, 2006]. Smoothing is based on word-to-word translation probabilities, for which statistics are available. The target-to-source lexical weighting is:

$$\phi(e|f, \mathbf{A}) = \log \prod_{j=1}^{T} \frac{1}{|\{i : (i,j) \in \mathbf{A}\}|} \sum_{i:(i,j)\in\mathbf{A}} p(f_i|e_j), \tag{2.6}$$

where $\mathbf{A}$ refers to some underlying word alignment. The reverse lexical weighting $\phi(f|e, \mathbf{A})$ is defined similarly. The word conditional probabilities $p(ff_i|e_j)$ are computed in a similar way as phrase conditional probabilities.

- **Lexicalized reordering:** These features are based on the orientation of a source phrase being translated with respect to the previously translated phrase. Reordering can be represented as the distance between these two source phrases. To avoid sparsity issues, orientation is limited to some heuristics and categories: the most widely used are *monotone*, *swap (s)* with the previously translated source phrase and *discontinuous (d)*. These categories are illustrated in Figure 2.2, borrowed from [Koehn, 2010]. The associated features are then computed:

$$\log \prod_{(\text{orientation},\mathbf{t},\mathbf{s})\in\mathcal{D}} p(\text{orientation}|\mathbf{s}, \mathbf{t}). \tag{2.7}$$

As we discussed earlier, there are several ways to compute the probabilities $p(\text{orientation}|\mathbf{s}, \mathbf{t})$ for all phrase pairs in the phrase table. A common practice is again to rely on relative frequencies of such events in the parallel corpus annotated with alignment. Orientation events can be defined either with respect to the word alignment [Tillmann, 2004a; Koehn *et al.*, 2005] or to the phrase alignment [Galley and Manning, 2008].

2.2.3 Summary

In this section, we have described a state-of-the-art phrase-based *SMT* system. In the following part, we will use such a system as a base for our phrase pivoting models.

This model is a weighted linear combination of feature functions. Translation hypotheses are constructed by concatenating phrase translations found in the phrase table of the translation system. This phrase table is typically built from a parallel corpus which is annotated with generalized phrase alignment. We have then discussed the standard set of features that are the state of the art and are used in current *SMT* systems.

2.3 Statistical Machine Translation for Morphologically Rich Languages

There has been active research on incorporating morphological knowledge in SMT. Most of the work done on studying the effects of morphological preprocessing on SMT quality focuses on translation from morphologically rich languages. Several approaches use pre-processing schemes, including segmentation of clitics [Lee, 2004; Habash and Sadat, 2006; Zollmann *et al.*, 2006], compound splitting [Nießen and Ney, 2004] and stemming [Goldwater and McClosky, 2005]. They show that reducing the sparsity caused by rich morphology through some form of morphological tokenization has a positive impact on the quality of SMT. There are also a growing number of publications that consider translation into morphologically rich languages such as Turkish [Oflazer and Durgar El-Kahlout, 2007], Arabic [Sarikaya and Deng, 2007; Badr *et al.*, 2008] and Persian [Kathol and Zheng, 2008].

Most of the focus here is on the efforts that studied the impact of morphological preprocessing on Arabic as a target language. In previous work on Arabic language modeling, OOV reduction was accomplished using morpheme-based models [Heintz, 2008]. [Diehl *et al.*, 2009] also used morphological decomposition for Arabic language modeling for speech recognition. They described an SMT approach to detokenization (or what they call morpheme-to-word conversion). Although the implementation details are different, their solution is comparable to one of our new (but not top performing) detokenization models (T+LM) (discussed in more details in Chapter 3. With regard to the English-to-Arabic MT, [Sarikaya and Deng, 2007] uses joint morphological-lexical language models to re-rank the output of the English-dialectal Arabic MT. [Badr *et al.*, 2008] reports results on the value of morphological tokenization of Arabic during training, and describes different techniques for the detokenizing Arabic output.

The research discussed in Chapter 3 is most closely related to that of [Badr *et al.*, 2008]. We extend their contribution in two ways: (a) We present a comparison of a larger number of tokenization schemes that yielded improved results over theirs; and (b) We discuss the technical challenges, and present solutions for producing unnormalized Arabic output through different detokenization techniques.

In another direction, there were efforts to enrich the source in word-based SMT, [Ueffing *et al.*, 2002] used POS tags, in order to deal with the verb conjugation of Spanish and Catalan. The POS tags were used to identify the pronoun+verb sequence and splice these two words into one term. The adopted this approach for single-word-based SMT which is solved like a phrase-based model. [Minkov *et al.*, 2007] suggested a post-processing system which syntactic features, in order to ensure grammatical agreement on the output. The method, using various grammatical source-side features, achieved higher accuracy when applied directly to the reference translations but it was not tested as a part of an MT system. Similarly, translating English into Turkish [Durgar El-Kahlout and Oflazer, 2006] uses POS and morph stems in the input along with rich Turkish morph tags on the target side, but improvement was gained only after augmenting the generation process with morphotactical knowledge. [Habash, 2007] also investigated case determination in Arabic. [Carpuat and Wu, 2007] approached the issue as a Word Sense Disambiguation problem.

Another method related to our approach in Section 5 is using an independent morphological prediction component such as used by [Minkov *et al.*, 2007] and [Toutanova *et al.*, 2008]. They use maximum entropy models for inflection prediction. Unlike our approach, they predict inflected word forms directly without going into a fine grained morphological feature prediction as we do. One of the main drawbacks of their approach is that they use stems as their base for translation instead of lemmas. There is also work by [Clifton and Sarkar, 2011] where they do segmentation and morpheme prediction. They also use stems as their basic word form.

2.3.1 Summary

In this section, we discussed some related work to modeling rich morphology in *SMT*. Many of the models depend on morphological preprocessing either by simplifying the morphologically rich language or enriching the morphologically poor language to match the richness of the opposite side. Other models depend on morphological-lexical language models. In addition, other efforts used POS, morph stems and various grammatical source-side features to ensure grammatical agreement on the output.

2.4 MT Evaluation and Error Analysis

One way of evaluating the output of an *SMT*system relies on a comparison between the system's output and correct translations. The problem of evaluation is usually solved either by asking a human expert to subjectively judge the quality of the system's output; or by explicitly constructing the correct answer and conceiving an objective comparison metric.

Subjective evaluation requires the annotators to judge the quality of a translation based on several criteria such as intelligibility, fluency, fidelity, adequacy and even informativity. This approach is adopted in recent evaluation campaigns [Callison-Burch *et al.*, 2008; Callison-Burch *et al.*, 2009]. Alternatively, the judgment may be based on how helpful the system's output was to the annotator to complete a specific task [Blanchon and Boitet, 2007]; or how easy was post-editing the output to obtain a correct translation [Specia, 2011].

Automatic evaluation mostly relies on a direct comparison between the system output hypothesis and the reference translations. The underlying assumption is that the closer the hypothesis is to the reference, the better its quality will be. In comparison with subjective evaluations, human annotator are involved just once in the process, when the reference is generated. The difficulty of automatic evaluation is two-fold. On the one hand, we have the difficulty of defining the correct translation. Usually one or several human experts are asked to translate the input sentence and build the set of references as an approximation of the space of correct translations. However, given the nature of translation this space is huge, and few translations are likely to cover only a small fraction of it. Recent technologies based on *meaning-equivalent semantics* tools [Dreyer and Marcu, 2012] provide the annotators with efficient ways to generate a large number of reference translations,thus resulting in a better approximation of the correct translations space.

The most widely used metric is the BLEU score [Papineni *et al.*, 2002b]. BLEU considers not only single word matches between the output and the reference sentence, but also n-gram matches, up to some maximum n. This formulation permits to reward sentences where local word order is closer to the local word order in the reference. BLEU is a precision-oriented metric; that is, it considers the number of n-gram matches as a fraction of the number of total n-grams in the output sentence.

There have been recent efforts in improving the quality of the evaluation and avoiding the harsh measures that depends on exact word matches. These efforts looked at paraphrasing and stemming of the output and the equivalent reference translation [Denkowski and Lavie, 2010; Snover *et al.*, 2009]. Stemming is not sufficient to capture similarity of syntactic structure or the similarity of semantic content. Moreover, paraphrasing focuses on matching words with different lexical choices with same meaning rather than handling the difference in the morphological choices between the output and the equivalent reference. Furthermore, none of these metrics provide detailed error analysis. Several publications defined different error classifications and typologies for the purpose of evaluation of single systems, or comparison between systems [Flanagan, 1994; Vilar *et al.*, 2006; Farrus *et al.*, 2010]. [Kirchhoff *et al.*, 2007] developed a framework for semi-automatically analyzing characteristics of input documents to MT systems that determine output performance. The framework heavily depends on human annotation.

To our knowledge, there hasn't been many efforts to build publicly available error analysis tools for MT output with focus on rich morphology which is our focus in Chapter 4.

[Popovic and Ney, 2006] provided precision and recall measures of MT output for different verbal inflections, but they only focus on Spanish verbs. Their word matching technique is a based on *PER* which may not be sufficient to apply in more general settings (i.e., not just verbs).

[A. Cuneyd Tantug and El-Kahlout, 2008] created a tool which is closely related to our work. They extended the BLEU and METEOR metrics to handle errors in Turkish morphology. Their matching algorithm uses Turkish word roots and a wordnet hierarchy, and it produces oracle score comparable to what AMEANA does.

[Stymne, 2011] presented a tool for annotation of bilingual segments intended for error analysis of MT. It utilizes a given error typology to annotate translations from an MT system. The tool does not provide detailed morphological error analysis.

2.4.1 Summary

In this section, we explored the different approaches to evaluate an MT system. We discussed the draw backs of the manual evaluation and the caveats of the most widely used automatic metrics. We also showed the lack of available detailed error analysis tools especially when targeting morphologically rich languages.

2.5 Pivoting in Statistical Machine Translation

A common solution to the data sparsity in the field is to pivot the translation through a third language (called pivot or bridge language) for which there exists abundant parallel corpora with the source and target languages. In this section, we review the three pivoting strategies that are our baselines.

2.5.1 Pivoting Strategies

Many researchers have investigated the use of pivoting (or bridging) approaches to solve the data scarcity issue [Utiyama and Isahara, 2007; Wu and Wang, 2009; Khalilov *et al.*, 2008; Bertoldi *et al.*, 2008; Habash and Hu, 2009]. The main idea is to introduce a pivot language, for which there exists large source-pivot and pivot-target bilingual corpora. Pivoting has been explored for closely related languages [Hajič *et al.*, 2000] as well as unrelated languages [Koehn *et al.*, 2009; Habash and Hu, 2009]. Many different pivot strategies have been presented in the literature. The following three are the most common ones.

2.5.1.1 Sentence Pivoting

In sentence pivoting, pivot language is used as an interface between two separate phrase-based MT systems in which we first translate the source sentence to the pivot language, and then translate the pivot language sentence to the target language [Khalilov *et al.*, 2008].

2.5.1.2 Synthetic Corpus

The second strategy is to create a synthetic source-target corpus by translating the pivot side of source-pivot corpus to the target language using an existing pivot-target model [Bertoldi *et al.*, 2008].

2.5.1.3 Phrase Pivoting

In phrase pivoting (sometimes called triangulation or phrase table multiplication), we train a source-pivot and an pivot-target translation models, such as those used in the sentence pivoting technique. Based on these two models, we induce a new source-target translation model.

Translation Model Since we build our models on top of Moses phrase-based SMT [Koehn *et al.*, 2007b], we need to provide the same set of phrase translation probability distributions. We follow [Utiyama and Isahara, 2007] in computing the probability distributions. The following are the set of equations used to compute the phrase-based SMT feature which are equivalent to the features discussed in Section 2.2.2.3. We compute the lexical probabilities (ϕ) and the phrase probabilities (p_w) according the following equations:

$$\phi(f|e) = \sum_{e} \phi(f|p)\phi(p|e) \tag{2.8}$$

$$\phi(e|f) = \sum_{e} \phi(e|p)\phi(p|f) \tag{2.9}$$

$$p(f|e) = \sum_{e} p(f|p)p(p|e) \tag{2.10}$$

$$p(e|f) = \sum_{e} p(e|p)p(p|f) \tag{2.11}$$

where f is the source phrase. p is the pivot phrase that is common in both source-pivot translation model and pivot-target translation model. e is the target phrase.

Since the underlying word alignment $\mathbf{A}$ doesn't exist, these equations are good approximation of the original features. Figure 2.3 illustrates the phrase pivoting process.

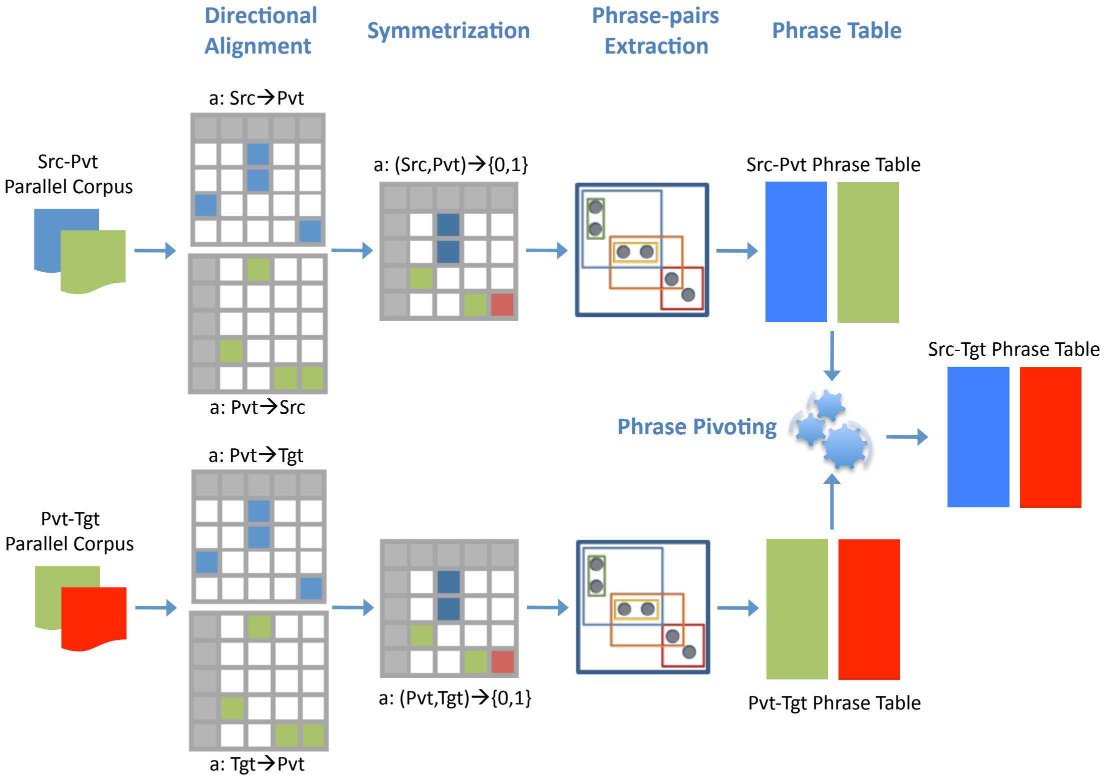


Figure 2.3: Phrase Pivoting process.

Reordering Model Following the reordering strategy in Moses phrase-based SMT system [Tillmann, 2004b; Koehn *et al.*, 2007b], we generate lexical reordering weights based on [Henriquez *et al.*, 2010] approach. Three different moves a phrase can make related to the previous and following phrase are considered: monotonous move, swap move and discontinuous move. There are three consideration to have in mind to calculate the reordering weights for phrase pivoting:

- A swap move on the Source-Pivot system is dissolved if the same phrase is swapped again on the Pivot-Target system which is then considered a monotonous move.
- A monotonous move followed by a swap means a swap from Source phrase to Target phrase. The same applies is the same if the swap if performed first and then the monotonous move.
- A discontinuous moves always generates a final discontinuous move no matter which move is performed before it.

Figure 2.4 shows a graphical example of the rules explained above. Following these rules, the monotonous weights for the Source-Target system $m(f|e)$ is calculated using this formula:

$$m(f|e) = \sum_p m(f|p)m(p|e) + \sum_p s(f|p)s(p|e) \tag{2.12}$$

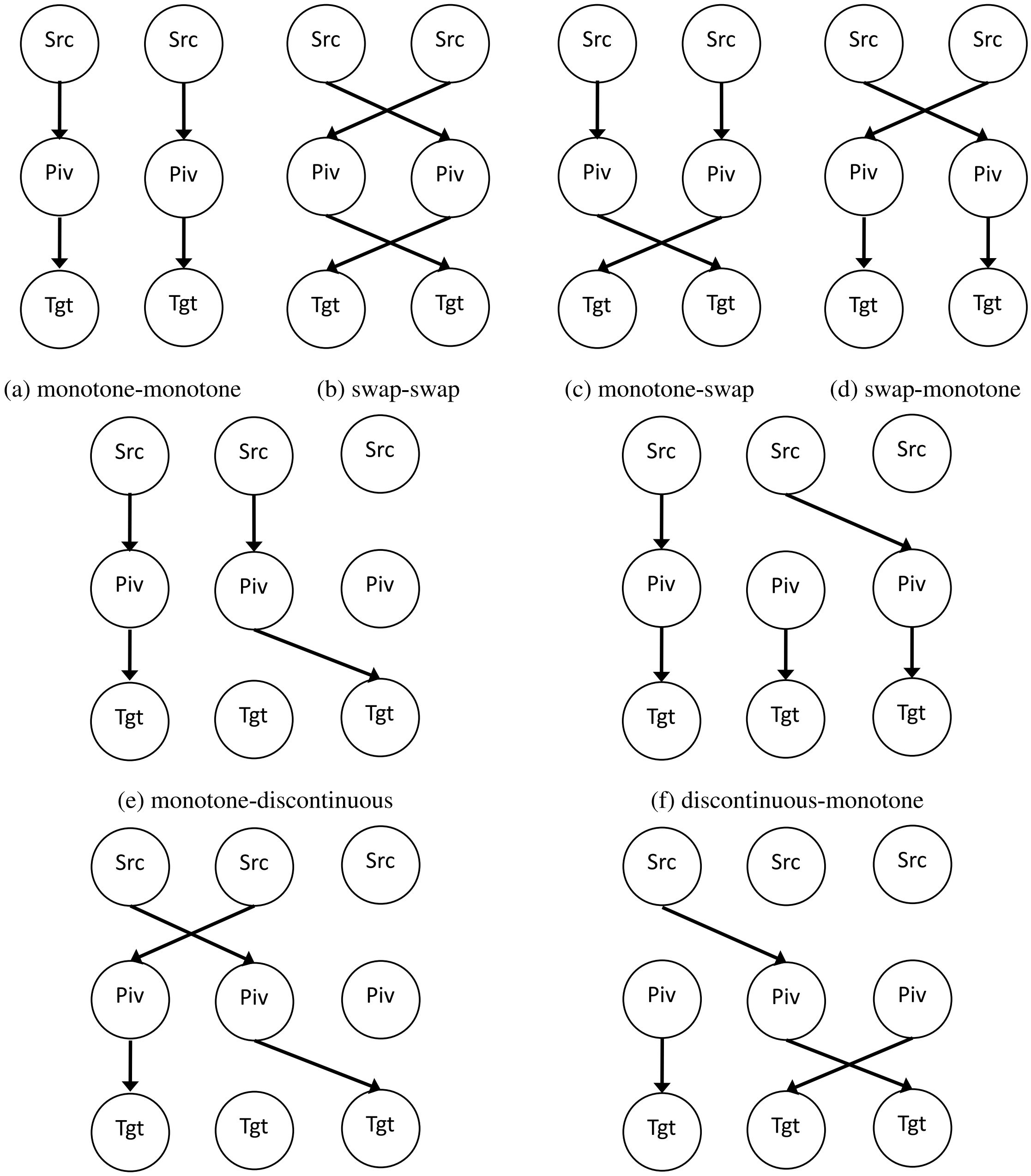
The swap weights $s(f|e)$ is calculated using this formula:

$$s(f|e) = \sum_p m(f|p)s(p|e) + \sum_p s(f|p)m(p|e) \tag{2.13}$$

And the discontinuous weights $d(f|e)$ calculated using the following formula:

$$\begin{aligned} d(f|e) = \sum_p m(f|p)d(p|e) + \sum_p d(f|p)m(p|e) \\ + \sum_p s(f|p)d(p|e) + \sum_p d(f|p)s(p|e) \\ + \sum_p d(f|p)d(p|e) \end{aligned} \tag{2.14}$$

where f is the source phrase. p is the pivot phrase that is common in both source-pivot translation model and pivot-target translation model. e is the target phrase.



(a) monotone-monotone

(b) swap-swap

(c) monotone-swap

(d) swap-monotone

(e) monotone-discontinuous

(f) discontinuous-monotone

(g) swap-discontinuous

(h) discontinuous-swap

Figure 2.4: Plots of different pivot lexical reordering scenarios.

2.5.2 Related Work

There have been some efforts on enhancing the recall and quality of pivot based SMT. In one effort by [Kumar and Franz, 2007], they utilized a bridge language to create a word alignment system and a procedure for combining word alignment systems from multiple bridge languages. The final translation is obtained by consensus decoding that combines hypotheses obtained using all bridge language word alignments.

[Paul *et al.*, 2009] examined the the effect of pivot language in the final translation system. He showed that in some cases if training data is small the pivot should be more similar to the source language, and if training data is large the pivot should be more similar to the target language. In Addition, it is more suitable to use a pivot language whose structure is similar to both of source and target languages.

In a recent work by [Zhu *et al.*, 2013], they focus on the problem having some useful source target translations not generated because the corresponding source phrase and target phrase connect to different pivot phrases. To alleviate the problem, they utilize Markov random walks to connect possible translation phrases between source and target language.

One of the manifestations of pivoting is that the size of the newly created pivot phrase table is very large. There has been some recent effort in improving the precision on pivoting. [Saralegi *et al.*, 2011] show that there is not transitive property between three languages. So many of the translations produced in the final phrase table might be wrong. Therefore for pruning wrong and weak phrases in the phrase table two methods have been used. One method is based on the structure of source dictionaries and the other is based on distributional similarity.

There is another recent work that uses context vectors to build a pruning method to remove those phrase pairs that connect to each other by a polysemous pivot phrase or by weak translations [Tofighi Zahabi *et al.*, 2013].

2.5.3 Summary

A common solution to the data sparsity in the field is to pivot the translation through a third language (called pivot or bridge language) for which there exists abundant parallel corpora with the source and target languages. In this section, we reviewed the three pivoting strategies and some related work to our approaches to improve the precision and recall of pivot-based models. Through out this thesis, we build on phrase pivoting since it was shown to be the best approach for pivoting [Utiyama and Isahara, 2007].

Rule Name	Tokenized	Untokenized	Example		
			Tokenized	**Untokenized**	**Gloss**
Li+Definite Article	ل+ال+ل? *l+Al+l?*	لل+ *ll+*	ل+ال+مكتب *l+Al+mktb*	للمكتب *llmktb*	‘for the office’
			ل+ال+لجنة *l+Al+ljnħ*	للجنة *lljnħ*	‘for the committee’
Ta-Marbuta	-ة+<ضمير> -*ħ*+<pron>	-ت+<ضمير> -*t*+<pron>	مكتبة+هم *mktbħ+hm*	مكتبتهم *mktbthm*	‘their library’
Alif-Maqsura	-ى+<ضمير> -*ý*+<pron>	-ا+<ضمير> -*A*+<pron>	روى+ه *rwý+h*	رواه *rwAh*	‘he watered it’
	exceptionally	-ي+<ضمير> -*y*+<pron>	على+ه *ςlý+h*	عليه *ςlyh*	‘on him’
Hamza	-ء+<ضمير> -’+<pron>	-ئ+<ضمير> -*ŷ*+<pron>	بهاء+ه *bhA’+h*	بهائه *bhAŷh*	‘his glory [gen.]’
	less frequently	-ؤ+<ضمير> -*ŵ*+<pron>	بهاء+ه *bhA’+h*	بهاؤه *bhAŵh*	‘his glory [nom.]’
	less frequently	-ء+<ضمير> -’+<pron>	بهاء+ه *bhA’+h*	بهاءه *bhA’h*	‘his glory [acc.]’
Y-Shadda	-ي+ي *-y+y*	-ي *-y*	قاضي+ي *qADy+y*	قاضي *qADy*	‘my judge’
N-Shadda	-ن+ن- *-n+n-*	-ن- *-n-*	من+نا *mn+nA*	منا *mnA*	‘from us’
N-Assimilation	من+م- *mn+m-*	م+م- *m+m-*	من+ما *mn+mA*	مما *mmA*	‘from which’
	عن+م- *ςn+m-*	ع+م- *ς+m*	عن+ما *ςn+mA*	عما *ςmA*	‘about which’
	أن+لا *Ân+lA*	ألّا *ÂlA*	أن+لا *Ân+lA*	ألّا *ÂlA*	‘that ... not’

Table 2.1: Arabic orthographic and morphological adjustment rules. <pron>/<ضمير> is a shorthand for *pronominal clitic*. The rules above are simplified in that they ignore short vowels which may affect the conditions of rule application, e.g., the Shadda rules assume that there are no short vowels intervening between the repeated letter, and the Hamza rule ambiguity is all the result of intervening unwritten short vowels. All examples are undiacritized.

	English	**Persian**	**Arabic**	**Hebrew**
Family	Indo-European	Indo-European	Semitic	Semitic
Script	Roman	Arabic	Arabic	Hebrew
Word Order	SVO	SOV	SVO/VSO	SVO
Morphology	Poor	Rich	Rich	Rich

Table 2.2: Language comparison between Arabic, Hebrew, Persian and English.

Chapter 3

Orthographic and Morphological Processing

In this chapter, we work on one component of performing phrase pivoting which is the direct translation to a morphologically rich language (MRL). The main focus of this section is to address the challenge of data sparsity due to richness in morphology. Most of our discussed and implemented approaches are focusing on Arabic as it is one of the most challenging languages in the field. We also focus on translating from English as it is the typical pivot language due to the existence of an abundance of resources and parallel corpora with many other languages.

We study the value of a variety of tokenization schemes and orthographic normalizations on English-Arabic SMT. However, since our goal is to always produce correctly detokenized and orthographically enriched Arabic words, we also consider different detokenization techniques and normalization settings. In this chapter, we discuss the various settings we studied for each of these three issues. First, various morphological tokenization schemes are considered to improve the SMT process. This is followed by detokenization techniques used to stitch the word parts back together. Finally we discuss the issue of enriched and reduced normalization which is orthogonal to the tokenization/detokenization question.

3.1 Morphological Tokenization

Morphological tokenization has been shown to be very helpful for machine translation involving morphologically rich languages. We consider five tokenization schemes discussed in the literature, in addition to a baseline no-tokenization scheme (D0). The D1, D2, TB and D3 schemes were first presented by [Habash and Sadat, 2006] and the S2 scheme was presented by [Badr *et al.*, 2008]. The D1, D2 and D3 schemes are named to indicate the degree of decliticization applied to the text. D1 separates conjunction proclitics; D2 extends D1 by separating prepositional clitics and particles (other than the definite article *Al*+); and D3 separates all clitics including the definite article and the pronominal enclitics. The S1 scheme used by [Badr *et al.*, 2008] is the same as [Habash and Sadat, 2006]'s D3 scheme. S2 is the same as D3 except that all proclitics are put together in a single proclitic cluster. TB is the Penn Arabic Treebank (PATB) [Maamouri *et al.*, 2004a] tokenization scheme. For more details on alternative tokenization schemes, see [Habash, 2010]. We use the Morphological Analysis and Disambiguation for Arabic (MADA)[1] toolkit [Habash and Rambow, 2005; Roth *et al.*, 2008] to produce the various tokenization schemes.

Figure 3.1 illustrates the different tokenization schemes with an example. Table 3.1 presents definitions and various relevant statistics for each tokenization scheme. The schemes differ widely in terms of the increase of number of tokens and the corresponding type count reduction. The more verbose schemes, i.e., schemes with more splitting and higher number of word tokens, have a lower number of token types, which leads to lower out-of-vocabulary (OOV) rates and lower perplexity; however, they are also harder to predict correctly. The increase in the number of tokens has consequences on word alignment, translation models and language models (LM). We control for these effects in our experiments in Section 3.4.

3.2 Detokenization

We compare the following six techniques for detokenization that vary in their degree of complexity and dependence on training data. The data used and the experiments setup are described in Section 3.4.1. For a baseline technique, we simply concatenate clitics to word without applying any orthographic or morphological adjustments; this is the **simple (S)** technique. Second, a **rule-based**

[1] We use MADA version 2.32 in the basic experiments and MADA version 3.0 in the final scaled up experiments

Arabic	وسينهى الرئيس جولته بزيارة الى تركيا.						
	wsynhý	Alrŷys	jwlth	bzyArħ	Alý	trkyA	.
Gloss	and will finish	the president	tour his	with visit	to	Turkey	.
English	The president will finish his tour with a visit to Turkey.						
Scheme							
D0	wsynhy	Alrŷys	jwlth	bzyArħ	Ǎlý	trkyA	.
D1	w+ synhy	Alrŷys	jwlth	bzyArħ	Ǎlý	trkyA	.
D2	w+ s+ ynhy	Alrŷys	jwlth	b+ zyArħ	Ǎlý	trkyA	.
TB	w+ s+ ynhy	Alrŷys	jwlħ +h	b+ zyArħ	Ǎlý	trkyA	.
S2	w+s+ ynhy	Al+ rŷys	jwlħ +h	b+ zyArħ	Ǎlý	trkyA	.
D3	w+ s+ ynhy	Al+ rŷys	jwlħ +h	b+ zyArħ	Ǎlý	trkyA	.
LEM	Ânhý	rŷys	jwlħ	zyArħ	Ǎlý	trkyA	.

Figure 3.1: A sentence in the various tokenization schemes. All tokenizations are in ENR normalization, but the original Arabic is in raw normalization. **LEM** is the lemma form of each word (discussed in Section 3.4.3.2).

	Definition	**Change Relative to D0**			**OOV**		**Perplexity**		**Prediction Error Rate**		
		Token#	ENR Type#	RED Type#	ENR	RED	ENR	RED	ENR	RED	SEG
D0	word				2.22	2.17	412.3	410.6	0.62	0.09	0.00
D1	cnj+ word	+7.2	-17.6	-17.8	1.91	1.89	259.3	258.2	0.76	0.23	0.14
D2	cnj+ prt+ word	+13.3	-32.3	-32.6	1.50	1.50	185.5	184.7	0.89	0.37	0.25
TB	cnj+ prt+ word +pron	+17.9	-43.9	-44.2	1.22	1.22	142.2	141.5	1.07	0.57	0.42
S2	cnj+prt+art word +pron	+40.6	-53.0	-53.3	0.91	0.91	69.3	69.0	1.20	0.73	0.60
D3	cnj+ prt+ art+ word +pron	+44.2	-53.0	-53.3	0.90	0.90	61.9	61.7	1.20	0.73	0.60

Table 3.1: A comparison of the different tokenization schemes studied in this article: tokenization scheme definition; the relative change from no-tokenization (D0) in tokens (Token#) and enriched and reduced word types (ENR Type# and RED Type#, respectively); out-of-vocabulary (OOV) rate; perplexity; MADA's prediction error rate for enriched tokens, reduced tokens and just segmentation (SEG). OOV rates and perplexity values are measured against the NIST MT04 test set while prediction error rates are measured against a Penn Arabic Treebank development set.

(R) technique uses deterministic rules to handle all of the cases described in Table 2.1. We pick the most frequent decision for ambiguous cases. The determination of frequency was done against the language model corpus [2]. We tokenized the whole corpus and for each tokenized word we counted the frequency of each equivalent untokenized form. Then we constructed rules that leads to the most frequent decision for the cases described in Table 2.1. The third technique is a **table-based (T)** technique that uses a lookup table mapping tokenized forms to detokenized forms. The table is based on pairs of tokenized and detokenized words from our language model data which had been processed by MADA. We pick the most frequent decision for ambiguous cases. Words not in the table are handled with the (S) technique. This technique essentially selects the detokenized form with the highest conditional probability $P(detokenized|tokenized)$. We also consider a variant of **T** technique that backs off to **R** not **S**: **Table+Rule (T+R)** technique.

The above-mentioned four techniques are the same as those used by [Badr *et al.*, 2008]. In this work, we introduce two new techniques that use a 5-gram untokenized-form language model

[2] Language model corpus is composed of 200M words from the Arabic Gigaword Corpus (LDC2007T40)

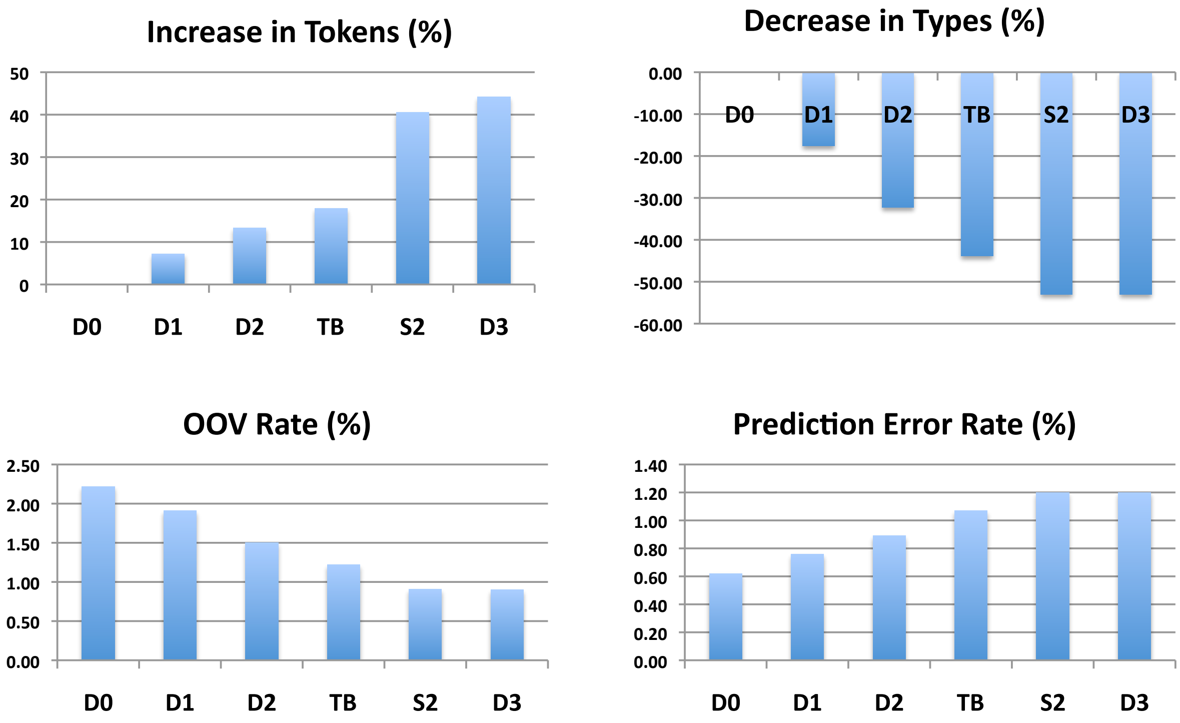


Figure 3.2: A comparison of the different tokenization schemes studied in this article: out-of-vocabulary (OOV) rate; perplexity; MADA's prediction error rate for enriched tokens, reduced tokens and just segmentation (SEG). OOV rates and perplexity values are measured against the NIST MT04 test set while prediction error rates are measured against a Penn Arabic Treebank development set.

(LM) and the `disambig` utility in the SRILM toolkit [Stolcke, 2002] to decide among different alternatives. First is **T+LM**; here we use all the forms in the **T** approach. Alternatives are given different conditional probabilities, $P(detokenized|tokenized)$, derived from the tables. Back-off is to the **S** technique. This technique essentially selects the detokenized form with the highest $P(detokenized|tokenized) \times P_{LM}(detokenized)$. Second is **T+R+LM**, a technique similar to **T+LM** but with **R** as back-off.

3.3 Orthographic Normalization

We consider two kinds of orthographic normalization schemes, enriched Arabic (ENR) and reduced Arabic (RED). For tokenized enriched forms, the detokenization produces the desired output. In case of reduced Arabic, we consider two alternatives to automatic orthographic enrichment. First, we use the Morphological Analysis and Disambiguation for Arabic (MADA)[3] toolkit [Habash and Rambow, 2005; Roth *et al.*, 2008] to enrich Arabic text after detokenization (MADA-ENR). MADA can predict the correct enriched form of Arabic words at 99.4%.[4] Alternatively, we jointly detokenize and enrich using detokenization tables that map reduced tokenized words to their enriched detokenized form (Joint-DETOK-ENR).

In terms of evaluation, we report our results in both reduced and enriched Arabic forms. We only compare in the matching form, i.e., reduced hypothesis to reduced reference and enriched hypothesis to enriched reference.

3.4 Evaluation

In this section we study the value of a variety of detokenization techniques over different tokenization schemes and orthographic normalization. We report results on naturally occurring Arabic text in the first two subsections. Then in the last subsection, we report results on English-Arabic SMT outputs with extended analysis.

3.4.1 Detokenization

We compare the performance of the different detokenization techniques discussed in Section 6.3.4 for the ENR and the RED normalization conditions. The performance of the different techniques is measured against the Arabic side of the NIST MT evaluation set for 2004 and 2005 (henceforth, MT04+MT05) which together have 2,409 sentences comprising 64,554 words. We report the results in Table 3.2 in terms of sentence-level detokenization error rate defined as the percentage of sentences with at least one detokenization error. The best performer across all conditions is the

[3] We use MADA version 2.32

[4] Statistics are measured on a development set from the Penn Arabic Treebank [Maamouri *et al.*, 2004a].

T+R+LM technique. The previously reported best performer was T+R [Badr *et al.*, 2008], which was only compared with D3 and S2 tokenizations only.

As illustrated in the results, the more complex the tokenization scheme, the more prone it is to detokenization errors. Moreover, RED has equal or worse results than ENR under all conditions except for the S detokenization technique with the TB, S2 and D3 schemes. This is a result of the S detokenization technique not performing any adjustments, which leads to the never-word-internal Alif-Maqsura character appearing incorrectly in word-internal positions in ENR. While for RED, the Alif-Maqsura is reductively normalized to Ya, which is the correct form in some of the cases.

The results for S2 and D3 are identical because these two schemes only superficially differ in whether proclitics are space-separated or not. Similarly, TB results are identical to D3 for the S and R techniques. This can be explained by the fact that the only difference between the D3 and TB schemes is that the definite article is attached to the word (in TB and not D3), a difference that does not produce different results under the deterministic S and R techniques.

We analyze the errors (14 cases) for the T+R+LM technique on D3 scheme and classify them into two categories. The first category comprises 11 cases (≈ 80% of the errors) and is caused by ambiguity resulting from the lack of diacritical marks. Seven (50% overall) of these errors involve the selection of the correct Hamza form before a pronominal enclitic. For example, the tokenized word و+أشقاء+ها *w+ÂšqA'+hA* 'and+siblings+her' can be detokenized to وأشقاءها *wÂšqA'hA* or وأشقائها *wÂšqAŷhA* or وأشقاؤها *wÂšqAŵhA* depending on the grammatical case of the noun أشقاء *ÂšqA'*, which is only expressible as a diacritical mark. The other four cases involve two closed class words, إن *Ăn* 'that/indeed' and لكن *lkn* 'however', each of which corresponding to two diacritized forms that require different adjustments. For example, the tokenized word إن+ني *Ăn+ny* can be detokenized to إني *Ăny* (إِن+نِي *Ăin+niy* → إِنِّي *Ăin∼iy*) or إنني *Ănny* (إِنَّ+نِي *Ăin∼a+niy* → إِنَّنِي *Ăin∼aniy*). In many cases, the n-gram language model is able to choose the correct form, but it is not always successful. The second category of errors compromises 3 cases (≈ 20% of the errors) which involve automatic tokenization failures producing tokens that are impossible to map back to the correct detokenized form.

	S		**R**		**T**		**T+R**		**T+LM**		**T+R+LM**	
	ENR	RED	ENR	RED	ENR	RED	ENR	RED	ENR	RED	ENR	RED
D1	0.17	0.17	0.17	0.17	0.08	0.08	0.08	0.08	0.08	0.08	**0.08**	**0.08**
D2	22.50	22.50	0.58	0.79	0.37	0.37	0.21	0.21	0.37	0.37	**0.21**	**0.21**
TB	38.36	35.53	1.41	3.03	1.33	1.49	0.75	0.91	1.16	1.25	**0.58**	**0.66**
S2	38.36	35.53	1.41	3.03	1.37	1.54	0.79	0.95	1.20	1.29	**0.62**	**0.71**
D3	38.36	35.53	1.41	3.03	1.37	1.54	0.79	0.95	1.20	1.29	**0.62**	**0.71**

Table 3.2: Detokenization results in terms of sentence-level detokenization error rate (SER).

3.4.2 Orthographic Enrichment and Detokenization

As previously mentioned, it is desirable for automatic applications generating Arabic to produce orthographically correct Arabic. As such, reduced tokenized output should be enriched and detokenized to produce proper Arabic. We compare next the two different enrichment techniques discussed in Section 6.3.4: using MADA to enrich detokenized reduced text (MADA-ENR) versus detokenizing and enriching in one joint step (Joint-DETOK-ENR). We consider the effect of applying these two techniques together with the various detokenization techniques when possible. The comparison is presented for D3 in Table 3.3. D3 has the highest number of tokens per word and it's the hardest to detokenize as shown in Table 3.2. The MADA-ENR enrichment technique can be applied to the output of all detokenization techniques; however, the Joint-DETOK-ENR enrichment technique can only be used as part of table-based detokenization techniques. The results for basic ENR and RED detokenization are in columns two and three (same values as the last row in Table 3.2). Columns four and five present the two approaches to enriching the tokenized reduced text. Although the Joint-DETOK-ENR technique does not outperform MADA-ENR for T and T+R, it significantly benefits from the use of the LM extension to these two techniques. In fact, Joint-DETOK-ENR produces the best results overall under T+R+LM, with an error rate that is 20% lower than the best performance by MADA-ENR. Overall, however, enriching and detokenizing RED text yields output that has almost 10 times the error rate compared to detokenizing ENR. This is expected since ENR is far less ambiguous than RED. The best performer across all conditions for

Detokenization	**Input Form**			
	ENR	RED	RED	
			Enrichment	
			MADA-ENR	Joint-DETOK-ENR
S	38.36	35.53	39.73	
R	1.41	3.03	10.59	
T	1.37	1.54	8.92	9.46
T+R	0.79	0.95	8.68	9.22
T+LM	1.20	1.29	9.34	6.23
T+R+LM	**0.62**	**0.71**	**7.39**	**5.89**

Table 3.3: Detokenization and enrichment results for D3 tokenization scheme in terms of sentence-level detokenization error rate.

detokenization and enrichment is the T+R+LM approach.

All experiments reported so far in this article start with a perfect pairing between the original and tokenized words. The real challenge is applying the detokenization techniques on automatically produced (noisy) text. The next section discusses the effect of detokenization on SMT output.

3.4.3 Tokenization and Detokenization for SMT

In this section we present the effect of tokenization in improving the quality of English-to-Arabic SMT. Then, we show the performance of the different detokenization techniques on the output and their reflections over the overall performance.

3.4.3.1 Experimental Data

All of the training data we use is available from the Linguistic Data Consortium (LDC).[5] We use an English-Arabic parallel corpus of about 142K sentences and 4.4 million words for translation model training data. The parallel text includes Arabic News (LDC2004T17), eTIRR (LDC2004E72), En-

[5] http://www.ldc.upenn.edu

glish translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18). Word alignment is done using GIZA++ [Och and Ney, 2003b]. For language modeling, we use 200M words from the Arabic Gigaword Corpus (LDC2007T40) together with the Arabic side of our training data. Twelve LMs were built for all combinations of normalization and tokenization schemes. We used 5-grams for all LMs unlike [Badr *et al.*, 2008], who used different n-grams sizes for tokenized and untokenized variants. All LMs are implemented using the SRILM toolkit [Stolcke, 2002].

MADA is used to preprocess the Arabic text for translation modeling and language modeling to produce enriched forms and tokenizations. English preprocessing simply includes down-casing, separating punctuation and splitting off "'s".

Standard use of GIZA++ includes filtering out sentences over a certain length (typically 100) and sentences with high ratio of source-to-target or target-to-source length (typically 9-to-1). We will refer to this as basic filtering. Due to the fact that the number of tokens per sentence changes from one tokenization scheme to another, GIZA++'s basic filtering will drop more sentences from the more verbose schemes. The percentage of sentences dropped due to the filtration process can be up to 2.3% in D3 (versus D0) for a generic cut off of 100 tokens per sentence in Arabic. It may seem like a small percentage; but since all dropped sentences are very long, this leads to D0 having access to 6.6 % extra words in training over D3. To control for this issue, we filter the training data so that all experiments are done on the same sentences. We use the D3 tokenization scheme as a reference and set the cutoff at 100 D3 tokens. We will refer to this as sentence length bias filtering.

All experiments are conducted using the Moses phrase-based SMT system [Koehn *et al.*, 2007b]. The decoding weight optimization was done using a set of 300 sentences from the 2004 NIST MT evaluation test set (MT04). The tuning is based on tokenized Arabic without detokenization. We use a max phrase length of size 8 for all tokenizations. For alignment symmetrization, we use the *grow-diag-final-and* method. And for the reordering parameter, we use the *monotonicity-bidirectional-fe* setting.

We report results on the 2005 NIST MT evaluation set (MT05). These test sets were created for Arabic-English MT and have 4 English references. We use only one Arabic reference in reverse direction for both tuning and testing. We evaluate using BLEU-4 [Papineni *et al.*, 2002a] although we are aware of its caveats [Callison-Burch *et al.*, 2006].

Align	Translation	Post	Reference
	Model	Process	Matching
Lemma	ENR	ENR	**25.3**
		RED	**25.3**
	RED	Joint-DETOK-ENR	24.9
		RED	25.0
Surface	ENR	ENR	24.9
		RED	25.0
	RED	Joint-DETOK-ENR	24.6
		RED	24.7

Table 3.4: Baseline SMT experiments with D0 tokenization. All results are in BLEU.

3.4.3.2 Baseline System

For our baseline system, using D0 tokenization, we compare the value of using lemmas for automatic word alignment as opposed to word surface forms (ENR or RED). In both cases, the phrase tables are built using the surface forms. We compare different combinations of settings for translation models and post-processing. For translation models, we either train on ENR or RED text. As for post-processing, we either keep the output as is, reduce it or enrich it. The enrichment is done using a variant of the Joint-DETOK-ENR technique discussed in Section 3.3. In this experiment set, we did not use the D3-based sentence length bias filtering described above. The results in Table 3.4 show that lemma-based alignment consistently yields superior results to surface-based alignment for the same translation model and post-processing conditions. The rest of the experiments in this article will all use lemma-based alignment in the following manner: when aligning a verbose tokenization, the lemma form will be used instead of the base word and the separated clitics will not be modified. Table 3.4 also shows that ENR training is better than RED training; however, since automatic enrichment error increases with tokenization verbosity (see Table 3.1, column 10), it is not clear which normalization settings is best to use with verbose schemes. We explore these combinations next.

Tokenization	**System Output**			
	Enr		Red	
	Enr	**Reduction**	**Enrichment**	Red
		Red	Joint-DeTok-Enr	
D0	24.6	24.7	24.7	24.7
D1	25.9	26.0	26.1	26.1
D2	26.4	26.5	26.1	26.2
TB	**26.5**	**26.5**	**26.7**	**26.8**
S2	25.7	25.8	26.1	26.2
D3	25.7	25.8	25.0	25.1

Table 3.5: Comparing different tokenizations schemes on 4 M data sets in BLEU scores

3.4.4 Tokenization Experiments

We compare the performance of the different tokenization schemes and normalization conditions. The results are presented in Table 3.5. The best performer across all conditions is the TB scheme. The previously reported best performer was S2 [Badr *et al.*, 2008], which was only compared against D0 and D3 tokenizations. Our results are consistent with [Badr *et al.*, 2008]'s results regarding D0 and D3. However, our TB result outperforms S2. The differences between TB and all other conditions are statistically significant above the 95% level. Statistical significance is computed using paired bootstrap resampling [Koehn, 2004]. Training over Red Arabic then enriching its output sometimes yields better results than training on Enr directly which is the case with the TB tokenization scheme. However, sometimes the opposite is true as demonstrated in the D3 results. This is likely due to a tradeoff between the quality of translation and the quality of detokenization.

3.4.5 Learning Curve Experiments

We also compare the value of different schemes across a learning curve where we consider smaller sets of our data: 2M, 1M and 0.5M words. We only show results for the reduced-then-enriched systems in Table 3.6. As expected, the increase in training data size causes an increase in BLEU

	0.5M	**1M**	**2M**	**4M**
D0	19.7	22.3	24.0	24.7
D1	21.0	23.2	23.7	26.1
D2	21.3	**23.7**	24.2	26.1
TB	**21.7**	23.6	**25.2**	**26.7**
S2	20.6	23.0	24.8	26.1
D3	20.5	23.0	24.5	25.0

Table 3.6: Comparing different tokenizations schemes over a learning curve using reduced-then-enriched systems. All results are in BLEU.

scores. Both TB and S2 at the 2M level outperform D0 at the 4 M level. The TB scheme is almost always the top performer. The S2 scheme goes from being ranked fourth in the smallest condition to being second in the largest. Further experiments considering the same learning curve with ENR training may be necessary to understand how different normalization settings interact with training size.

3.4.5.1 SMT Sensitivity to Different Detokenization Techniques

We measure the performance of the different detokenization techniques discussed in Section 3.2 against the SMT output for the TB tokenization scheme. We report results in terms of BLEU scores in Table 3.7. The results for basic ENR and RED detokenization are in columns two and three. Column four presents the results for the Joint-DETOK-ENR approach to joint enriching and detokenization of tokenized reduced output discussed in Section 6.3.4.

When comparing Table 3.7 (in BLEU scores) with the corresponding cells in Table 3.3 (in sentence-level detokenization error rate), we observe that the wide range of performance in Table 3.3 is not reflected in BLEU scores in Table 3.7. This is expected given the different natures of the tasks and metrics used. Although the various detokenization techniques do not preserve their relative order completely, the S technique remains the worst performer and T+R+LM remains the best in both tables. However, the R and T+LM techniques perform relatively much better with MT output than they do with naturally occurring text. The most interesting observation is perhaps that under the best performing T+R+LM technique, joint detokenization and enrichment (Joint-DeTok-Enr) outperforms Enr detokenization despite the fact that Joint-DeTok-Enr has over nine times the error rate in Table 3.3. This shows that improved MT quality using Red training data out-weighs the lower quality of automatic enrichment.

3.4.5.2 SMT Detokenization Error Analysis

Since we do not have a gold detokenization reference for our MT output, we automatically identify detokenization errors resulting in non-words (i.e., invalid words). We analyze the SMT output for the D3 tokenization scheme and T+R+LM detokenization technique using the morphological analyzer component in the MADA toolkit,[6] which provides all possible morphological analyses for a given word and identifies words with no analysis. We find 94 cases of words with no analysis out of 27,151 words (0.34%), appearing in 84 sentences out of 1,056 (7.9%). Most of the errors come from producing incompatible sequences of clitics, such as having a definite article with a pronominal clitic. For instance, the tokenized word ال+علاقة+نا *Al+ςlAqħ+nA* 'the+relation+our' is detokenized to العلاقتنا *AlςlAqtnA* which is grammatically incorrect. This is not a detokenization problem per se but rather an MT error. Such errors could still be addressed with specific detokenization extensions such as removing either the definite article or the pronominal clitic.

3.5 Improving and Scaling Up

In this section we present results demonstrating the effect of scaling up the training data and the relative gain in the quality of English-to-Arabic SMT using an updated version of MADA. We report results on the baseline system (D0) and our best system (TB). Then, we provide a detailed error analysis of the different types of morphological errors in the output.

3.5.1 Experiment Setup and Results

We use the same setup used for all the previous experiments explained in Section 3.4.3.1 but we scale up the English-Arabic parallel corpus ≈15 times. The corpus size is about 2.8m sentences

[6] This component uses the databases of the Buckwalter Arabic Morphological Analyzer [Buckwalter, 2004].

Detokenization	**SMT Output Form**		
	Enr	Red	
		Red	Enrichment
			Joint-DeTok-Enr
S	25.6	26.0	N/A
R	26.5	26.8	N/A
T	26.4	26.8	22.4
T+R	26.4	26.8	22.4
T+LM	26.5	26.8	26.7
T+R+LM	**26.5**	26.8	**26.7**

Table 3.7: BLEU scores for SMT outputs with different detokenization techniques over TB tokenization scheme

($\approx$60 million words). All data we use is available from LDC[7] and GALE[8] constrained data. We also use an updated version MADA (v 3.0) instead of MADA (v 2.32), to pre-process the Arabic text for the translation model and language model (LM). To control for the change in the MADA version and to compare the results of the scaled up systems (D0-60m & TB-60m) to the basic systems trained on 4 M words, we re-conducted the basic experiments for the baseline system (D0-4m) and our best system (TB-4m) using the new version of MADA. We replicated the basic experiments once with D3-based sentence length bias filtering in addition to the basic filtering discussed in Section 3.4.3.1 and once with just the basic filtering.

We report results in terms of BLEU scores in Table 3.8. The first two rows are the old results of the basic systems D0 (baseline) & TB (our best system) trained on 4 M words and using the old version of MADA (v 2.32) and applying D3-based sentence length bias filtering. The following two rows are the results based on the same systems setup except for using the newer version of

[7] LDC Catalog IDs: LDC2005E83, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006G05, LDC2007E06, LDC2007E101, LDC2007E103, LDC2007E46, LDC2007E86, LDC2008E40, LDC2008E56, LDC2008G05, LDC2009E16, LDC2009G01.

[8] Global Autonomous Language Exploitation, or GALE, is a DARPA-funded research project.

Data	**Filtering**	**MADA**	**BLEU**
D0-4m	Basic+D3	v2.32	24.7
TB-4m	Basic+D3	v2.32	**26.7**
D0-4m	Basic+D3	v3.0	25.4
TB-4m	Basic+D3	v3.0	**27.1**
D0-4m	Basic	v3.0	26.0
TB-4m	Basic	v3.0	**27.3**

Table 3.8: BLEU scores for the basic SMT systems outputs under different filtering conditions and different MADA versions.

Data	**Filtering**	**MADA**	**BLEU**	**METEOR**	**TER**
D0-60m	Basic	v3.0	31.3	48.9	48.9
TB-60m	Basic	v3.0	**32.3**	**49.5**	**48.5**

Table 3.9: BLEU, METEOR and TER scores for the scaled up SMT systems outputs.

MADA (v 3.0). The results show that the new MADA with the improved quality of tokenization and enrichment leads to a boost in the quality of the translation by 0.7 BLEU point in D0 and 0.4 BLEU point in TB. In addition, results in rows five and six show that even without the sentence length bias filtering, TB still outperforms D0 in the basic systems. In Table 3.9, we report results in terms of BLEU, METEOR and TER. We show that when we scale up the training data, the relative improvement that we get in the basic systems between D0 and TB is still maintained although slightly reduced (from 1.3 BLEU to 1 BLEU difference). We also noted improvement with other metrics that are not part of our optimization process like METEOR and TER. This suggests that tokenization can still help even with a much larger data set. This result (comparing TB to D0) is also corroborated using a much larger data set (150 M words) by [Al-Haj and Lavie, 2010].

Data	BLEU	Reduced BLEU	Lemmatized BLEU
D0-4m	26.0	26.0	33.6
TB-4m	27.3	27.3	34.7
D0-60m	31.3	31.5	39.3
TB-60m	32.2	32.4	40.3

Table 3.10: Three different BLEU metrics for the basic and scaled up SMT systems' outputs with the basic filtering.

3.5.2 BLEU Score Analysis

In order to overcome the limitations of BLEU [Callison-Burch *et al.*, 2006], in Table 3.10 we produce three sets of numbers for the baseline (D0) and our best system (TB) across different systems' setups; basic and scaled up. The first set of numbers are the vanilla BLEU scores that we use in all our experiments. The second set of numbers are Reduced BLEU where the system output and the reference are reduced (RED) during evaluation. The last set of numbers are what we call Lemmatized BLEU where we try to match the words in the output with words in the reference and if there is no exact match, we try to match based on the lemma with the corresponding reference words. This way we factor out the errors resulting from incorrect morphological generation and focus on the lexical choice of words in the translation process.

The Reduced BLEU results are higher than the vanilla BLEU scores as we expected but not by much. This shows the robustness of the denormalization process. The interesting results here are the Lemmatized BLEU scores which show a potential increase of ≈8 BLEU points by improving the output morphological form across all different systems. In the next sub-section, we investigate the Lemmatized BLEU scores in more details.

3.5.3 Unigram Error Analysis

In computing the Lemmatized BLEU scores, we divide the output into three categories. The first category includes words which are correct and exactly match words in the reference. The second category includes the Lemma Match words where the words of the output are matched with words in the reference based on the lemma. The last category includes words which can't be matched automatically with any of the reference words. Table 3.11 shows the distribution of the output words on the three different categories across different systems' setups; basic and scaled up. Following are the main conclusions:

- ≈57% of the output is correct in the basic systems while ≈63% of the output is correct in the scaled up systems which is the expected effect of scaling up the data
- ≈12-13% of the output could be corrected by making better morphological choices across all different systems
- ≈29% of the basic systems' outputs and ≈24% of the scaled up systems' outputs can not be matched automatically with any of the reference word
- Drop in unmatchable words is almost 4 times the drop in Lemma Match as we scale up. This suggest that scaling up the data helps in increasing the recall of the output but there's still a big margin of improvement by making better morphological choices.

Another interesting observation is that despite the fact that TB systems always produce more Exact Match words than D0 systems, it's not reflected in the ratio over all words. This is the result of D0 systems producing less number of words than TB systems which in return affect the brevity penalty values when computing the BLEU scores. The brevity penalty of the four systems D0-4m, TB-4m, D0-60m and TB-60m are 0.9923, 1.0 , 0.9683 and 0.9807 respectively.

3.5.4 Lemma Match Errors

In the previous section, we showed that ≈12-13% of the output could be corrected by making better morphological choices. In this section, we examine the different types of morphological errors in the Lemma Match words of our scaled up systems (TB-60m & D0-60m). In Table 3.12, we report results in terms of the percentage of Lemma Match words affected by each morphological error. The total doesn't sum up to 100% because a word could have more than one error which counts for 19% of the words in TB-60m and 25% in D0-60m which shows that tokenization helps in reducing the number of morphological errors per word. An interesting point in the results is that ≈40% of the errors come from TB clitics in both systems which shows that tokenization helps but we still need

Data	Exact Match	Lemma Match	Unmatchable	Total
D0-4m	16092 (57.01%)	3895 (13.80%)	8242 (29.20%)	28229
TB-4m	16335 (57.23%)	3941 (13.81%)	8265 (28.96%)	**28541**
D0-60m	17394 (63.08%)	3498 (12.69%)	6682 (24.23%)	27574
TB-60m	17534 (62.80%)	3542 (12.69%)	6846 (24.52%)	**27922**

Table 3.11: Unigram error analysis for the basic and scaled up SMT systems' outputs: # of words which match exactly words in the reference; # of words which match words in the reference based on the lemma; # of unmatched words with any in the reference automatically; total number of words in the translation output.

to work more on making better morphological choices in Arabic generation. The results also show that deleting or adding a determiner is the most common error in ≈30% of the Lemma Match words in both systems. Moreover, we noticed that gender, stem and number (in addition to stem, which reflects number change in the form of a broken plural) count for the biggest percentage of the errors. This could be explained by the fact that Arabic is highly inflected with these three morphological features unlike English which leads to many errors. The rest of the morphological features do not contribute much in the errors for many reasons. For example, case in Arabic is only marked by diacritics except for the plural form of some nouns and since all the data are non-diacritized, the effect of case is very small. On the contrary, person and aspect features are explicitly determined in Arabic but it's also explicitly determined in English which helps in the translation process.

3.5.5 Unmatchable Words Analysis

We took a sample of 50 sentences (1,224 words) from the output of the upscaled best system (TB-60m) and conducted a manual error analysis on the Unmatchable Words. We divide the errors into four categories. The first category compromises words which are considered a correct paraphrase of words in the reference and hold the same meaning. The second category compromises the incorrectly translated words. The third category includes all punctuations errors; for example, adding or deleting periods and commas. The last category compromises the out of vocabulary words (OOV). Table 3.13 shows the results of the analysis. We can see from the results that punctuation errors

	Incorrect Feature	**% Words Affected**	
		TB-60m	**D0-60m**
TB Clitics	Conjunction Proclitics	24.07%	21.85%
	Prepositional Proclitics	18.29%	18.83%
	Pronominal Enclitics	13.34%	12.15%
	Determiner	**29.84%**	**32.10%**
	Gender	17.52%	17.07%
	Stem	12.86%	12.98%
	Number	9.90%	10.07%
	State	2.67%	2.94%
	Aspect	2.28%	2.22%
	Case	2.02%	2.19%
	Person	0.83%	0.89%
	Mood	0.54%	0.52%
	Voice	0.11%	0.12%
	Multiple Features	19%	25%

Table 3.12: Lemma Match morphological errors for TB-60m and D0-60m systems' outputs.

Category	**% of Unmatchable Words**	**% of all Words**	**Example**
Correct Paraphrase	71.12%	17.44%	ÂðAr vs. mArs ‘March’ tHdθ vs. tklm ‘speak’
Incorrect Translation	18.63%	4.57%	θAlθ vs. θlθ ‘Third vs. [One] Third’
Punctuation Errors	7.76%	1.90%	Adding commas, periods, etc.
OOV	2.48%	0.61%	La Picota [proper name]

Table 3.13: Unmatchable Words Analysis for our best scaled up system TB-60m.

and the out of vocabulary words have the least share in the errors. The big bulk of the errors which counts for ≈70% of the unmatchable words and ≈17% of total words are correct paraphrases of words in the reference which could be accounted for by having multi reference. Moreover, there are some word order errors which are not captured by these numbers. We plan to investigate these types of errors in the future work.

3.6 Conclusions

We presented experiments studying a large number of variables for English-Arabic SMT systems that produce correctly tokenized and enriched Arabic text. The results show that lemma based alignment leads to a better output quality. Our best system uses the Penn Arabic Treebank (TB) tokenization scheme and reduced Arabic word forms followed by a language-model based joint detokenization and enrichment step.

Chapter 4

AMEANA Error Analysis

Error analysis is a central part of the research process in natural language processing (NLP). Through error analysis, researchers and developers can better understand the strengths and weaknesses of their systems. The more detailed the analysis, the more specific the insights can be. Morphologically rich languages, such as Arabic, Turkish or German, are particularly challenging since there is a large space of possible details to explore at the word morphology level. Human evaluation is an attractive solution; however, it typically involves qualitative measures, such as fluency or adequacy, which are very generic and do not capture nor quantify word-level errors. Fine-grained human analysis suffers from low speed, high cost and low consistency due to fatigue and adaptation to machine-generated language. Automating error analysis is a good solution, although simple matching techniques can be too coarse to be helpful.

We present AMEANA (Automatic Morphological Error Analysis), AMEANA produces detailed statistics on morphological errors in the output. It also generates an oracularly modified version of the output that can be used to measure the effect of these errors using any evaluation metric. AMEANA is a language independent tool except that a morphological analyzer must be provided for a given language.

4.1 Motivation

Most MT automatic evaluation metrics, such as BLEU [Papineni *et al.*, 2002a], focus on comparing an MT output against a set of references in order to assign a similarity score. The scores are typically based on exact word matching, a particularly harsh measure especially for morphologically rich languages. This is due in part to two phenomena. First is **Morphological Richness:** words sharing the same core meaning (represented by the lemma or lexeme) can be said to inflect for different morphological features, e.g., gender and number. These features can realize using concatenative (affixes and stems) and/or templatic (root and patterns) morphology. Second is **Morphological Ambiguity:** words with different lemmas can have the same inflected form. As such, a word form can have more than one morphological analysis (represented as a lemma and a set of feature-value pairs). This is especially problematic for languages with reduced orthographies such as Arabic or Hebrew.

Using an abstraction of the word, such as the stem or the lemma, to match output and reference words can address the harshness of exact form matching. Stemming has been shown to be helpful in MT evaluation [Denkowski and Lavie, 2010]; but simple stemming is not sufficient when dealing with morphologically rich languages as it suffers from errors of omission and errors of commission [Krovetz, 1993]: words with the same core meaning not sharing the same stem, and words with different core meanings sharing the same stem. This is especially problematic for words with templatic morphology, e.g., broken plurals in Arabic.[1] Furthermore, simple stemming does not properly address ambiguity as most shallow stemmers do not provide more than one stem for a given word. A more sophisticated stemming approach using a morphological analyzer can address this limitation. AMEANA can be used with stems, lemmas or even higher abstractions relating different lemmas to each other. In the case study we present on Arabic, we use the lemma representations produced by a morphological analyzer because of the above-mentioned limitations of stemming. We plan to study the use of higher abstractions in the future.

Form abstraction, however, is a double edged sword since it will lead to numerous matching points between the output and reference. To address this concern, AMEANA uses a word matching (alignment) algorithm that minimizes the number of morphological differences and sentence-relative word position.

[1] In broken plurals, the functional number (plural) is inconsistent with the morphological ending (singular suffix) [Alkuhlani and Habash, 2011]. Plurality is indicated using a word template realized as a stem that is different from the singular stem.

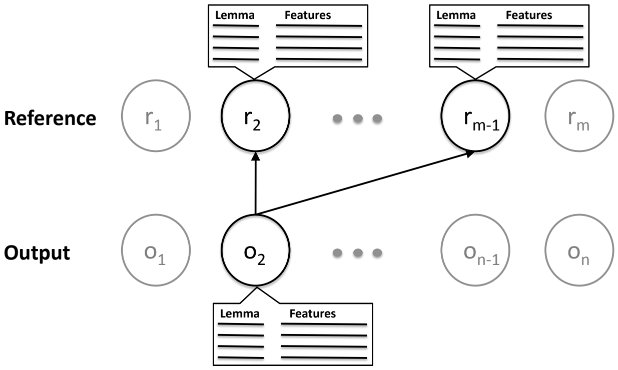


Figure 4.1: Output word o_2 has at least one common lemma with reference words r_2 and r_{m-1}. Our alignment algorithm selects edges minimizing differences in features (primarily) and relative positions (secondarily), while maximizing the number of paired output-reference words.

4.2 Alignment Algorithm

In this section, we describe the algorithm used in aligning the output words with their matching reference words. The alignment is then used to produce detailed morphological-error diagnostics and an oracularly modified output to use with MT evaluation metrics. A sample of these diagnostics is shown in Section 4.6.1.2. Our approach is close to efforts by [Denkowski and Lavie, 2010]. However, we focus on morphology while the other approach is focused on paraphrase matching.

For every sentence pair of MT output and its reference translation, we apply the following alignment algorithm (see Figure 4.1):

First: Morphological Analysis We run a morphological analyzer on all output and reference words producing a set of lemmas and their associated analyses for each word. A morphological disambiguator or part-of-speech (POS) tagger can be used to limit the choices given to AMEANA, e.g., [Habash and Rambow, 2005]. This is not required and perhaps even not desirable given error propagation resulting from running a disambiguator on automatically generated text.

Second: Graph Construction We build a graph where each word is represented by a node. We draw an edge for each output-reference word pair if there is at least one common lemma between them. Each edge receives a weight based on the following equation:

$$W \;=\; \min\left(D_{ab}\right) + \frac{\left(\left|\frac{P_a}{S_a} - \frac{P_b}{S_b}\right|\right)}{2}$$

We define a and b as words in output and reference. For each pair of morphological analyses for a

and b sharing the same lemma, we compute the count of features with different values. We define D_{ab} as the set of all feature-difference counts. Consequently, $\min(D_{ab})$ is the minimum feature difference possible between words a and b. We define P_a and P_b as the position of words a and b in their respective sentences. We also define S_a and S_b as the lengths of the sentences in which a and b appear, respectively. The absolute difference in relative word position $\left|\frac{P_a}{S_a} - \frac{P_b}{S_b}\right|$ is used as a tie breaker. It is divided by 2 to account for the extreme case of matching words at opposite ends of their respective sentences. The smaller the value W, the closer the two words a and b are to each other. In this equation, we give more weight to feature differences by giving a whole point for each mismatching feature, while word position distance is used as a tie breaker.

Third: Bipartite Matching Once the graph is constructed, the search space for the alignment is defined as a maximum bipartite matching problem constrained on the weights of the edges. We use a modified version of the Ford-Fulkerson algorithm [Ford and Fulkerson, 1956] to solve the matching problem and select a number of edges that maximizes the number of aligned output-reference words.

After alignment, each output word receives a matching category based on the reference word it is paired with. If the output and reference words have the same form, the category is an *Exact Match*, otherwise, it is a *Lemma Match*. Unpaired output words receive the category *Unmatchable*.

4.3 Morphological Diagnostics

We sum over all the feature differences in the *Lemma Match* category words. In cases with multiple analyses with the same lemma and same minimum feature-difference count, we assign equal partial error to each analysis so that they sum up to 1 instead of choosing among them. The partial errors are aggregated for each feature difference in all analyses. We will generically refer to feature differences as *errors* with respect to the reference, although some may not actually be erroneous (albeit not directly matching).

AMEANA produces general statistics such as the number and percentage of *Exact Match*, *Lemma Match* and *Unmatchable* words; the average number of errors per sentence; and the number of sentences with a certain number of errors. Detailed statistics are produced for errors in *Lemma Match* words including the number and percentage of errors for all features, feature-value pairs, and their combination. Additionally, AMEANA produces precision, recall and F-scores for correctly generating the various features. See Section 4.6.1.2 for some examples of these statistics.

4.4 Use for MT Evaluation

One of the side benefits provided by AMEANA is the production of an oracularly modified MT output where output words with a *Lemma Match* are replaced with the reference words they are aligned to. The modified output can be run through any evaluation metric such as BLEU or METEOR to get the upper limit of improvement the system can reach by just making better morphological choices. AMEANA also gives the user the option of restricting the oracle generation such that certain morphological features, in addition to the lemma, must correctly match the reference.

4.5 AMEANA **Language Independence**

As mentioned above, AMEANA is a language-independent error-analysis tool. To use AMEANA for a particular language, the user must specify the following parameters in a simple and easy to use configuration file:

- The output of a morphological analyzer run on the MT output and the reference.
- The tag marking the lemma in the morphological analyzer output and the list of morphological features to consider in the error analysis.
- A list of prior probabilities of each value for each morphological feature. If not provided, a uniform value is used and when there's a conflict, the first option is always selected.
- A list of features to focus on in oracle generation, if desired.

Once these parameters are specified in the config file, the user can use AMEANA seamlessly with any language. There are other options and configuration values that the user can work with. They are described in the user manual provided with the tool.

4.6 Case Studies

4.6.1 AMEANA for Arabic

In this section and the following section, we work with an English-to-Arabic Statistical MT system as a case study to show the different error-analysis outputs of AMEANA and to verify its performance. Since Arabic is the target language of the MT system we use, we first discuss relevant aspects of Arabic morphology, and how we adapt AMEANA to work with Arabic.

4.6.1.1 Adapting AMEANA to work with Arabic

In order to make AMEANA work for Arabic, we have to modify the configuration file to specify the parameters mentioned in Section 4.5. For the morphological analyzer, we use ALMORGEANA (ALMOR) [Habash, 2007]. ALMOR is a morphological analysis and generation system for Arabic. It provides analyses of a given word based on the lemma-and-features level of representation which is what we want as an input for AMEANA.

4.6.1.2 Evaluation

In this section, we present two sets of results: a demonstration of the use of AMEANA for MT error analysis and a study verifying its behavior.

Machine Translation Error Analysis We ran AMEANA on the output of three SMT systems based on previous work on English-Arabic SMT [El Kholy and Habash, 2010a]. We present next the experimental settings of the MT systems. Then we present four sets of results produced by AMEANA to demonstrate its usability.

MT Experimental Settings All systems share the following settings. They use the Moses phrase-based SMT decoder [Koehn *et al.*, 2007b] trained on an English-Arabic parallel corpus of about 135k sentences (4 million words). Phrase-table maximum phrase length is 8. Word alignment is done using GIZA++ [Och and Ney, 2003b] run on the lemma level of representation. Lemmatization as well as tokenization (discussed below) is done using the MADA+TOKAN toolkit [Habash and Rambow, 2005]. A 5-gram language model is based on 200M words from the Arabic Gigaword together and the Arabic side of the training data [Stolcke, 2002]. Decoding weight optimization [Och, 2003a] is done using 300 sentences from the 2004 NIST MT evaluation test set (MT04). Systems are compared on their performance on the 2005 NIST MT evaluation set (MT05). This Arabic-English test set has four English references. We invert it by selecting the first English reference to be our input and use the Arabic side as the only reference.

The three systems vary as follows: the D0 system uses no morphological tokenization whatsoever; the TB system uses the PATB tokenization scheme [Maamouri *et al.*, 2004b]; and the LEM system uses PATB tokenization and keeps the main word in lemma form. We have shown in Chapter 3 that TB outperforms D0; and LEM is the lemmatized version of TB used in TB's word alignment [El Kholy and Habash, 2010a]. We expect LEM to under perform compared to the other two systems. The first three columns of Table 4.4 show automatic evaluation scores in three metrics for all systems.

Overall Lemma Match Statistics Table 4.1 shows the AMEANA output of one sentence from the TB system. The first four lines are the English input sentence, Arabic translation reference, MT output, and the AMEANA modified MT output, respectively. Following that is word-by-word analysis in the following format. The first row is the original MT output words in sequence and the second row is the modified MT words. Third row is the matching category while the fourth and fifth rows are the morphological features differences between the MT output word and its reference translation word when the matching category is *Lemma Match*.

Table 4.2 shows the numbers and percentage of words in each matching category for the three systems. *Exact Match* is the simplest statistics that can be obtained using any MT evaluation metric, e.g., it is a sub-score used in BLEU. AMEANA allows us to distinguish a subset of no matches that can be matched at the lemma level. This allows to quantify the percentage of words that have no lexical translation problems (since they have matching lemmas) and identify the subset that has feature problems even though the lemma is correct. Such distinction may be useful for techniques involving post-editing or word-repair. The D0 and TB systems have similar *Exact Match* percentages. In both systems about one-third of the *non-Exact Match* cases have matchable lemmas. LEM has a much lower *Exact Match* but also a much higher *Lemma Match*. LEM overall has the highest *Any Match* (includes *Exact Match* & *Lemma Match*), which suggests it has the highest lexical translation quality despite its low fluency.

Lemma Match Error Distribution Table 4.3(a) presents the percentage of matching errors among the *Lemma Match* words, which are only about a seventh (D0, TB) or a third (LEM) of all words. The errors are classified by feature, e.g., conjunction, determiner, or gender; and by two feature-classes: *PATB clitics* and *other features*. This table allows us to study the distribution of various error types per system. Comparing across systems must take into account the size of the *Lemma Match* set of words. For example, although LEM has a lower percentage of pronominal clitics than TB or D0, it actually has 40% more instances of errors. Overall, these numbers show that the determiner is the biggest single feature error across systems. Non-PATB clitic errors collectively constitute a smaller proportion of matching errors than other word features, although the difference between the two sets gets smaller in our best performer, TB. The PATB clitics together with determiner, gender and number are biggest culprits overall. This analysis suggests targeting them may be most beneficial. Some features have low counts because they are associated with specific POS which are less frequent, e.g., verbal mood, voice and aspect.

Morphological Feature Correctness Table 4.3(b) presents the F-measure (balanced harmonic mean of the precision and recall) of words matching between the output and the reference for a variety of matching criteria of morphological features. The last two rows are for *Exact Match* and *Any Match*. These two can be interpreted as the lowest and highest limits on matching given the space of morphological errors. While *Exact Match* requires the lemma and all features to match, *Any Match* only requires the lemma to match – of course, in *Exact Match*, the lemma matches by definition. The rest of the rows are for matching subsets that include the lemma together with a particular feature, such as conjunction or determiner. These numbers are not oracle scores, they reflect the correctness of the text on different morphological features even if the final word form is not matchable.

Across all features, TB outperforms D0. This is consistent with their overall BLEU scores; however, it is interesting to see that the improvement in features other than PATB clitics is actually more than in PATB clitics overall (by 1.9% compared to 1.4%). The main area LEM is suffering compared to TB and D0 is in non-PATB clitics. This is expected given the lack of inflections in the output of LEM. Lemma plus determiner matching yields the lowest single F-score over than *Exact*

Match. That said, it is about 70% of the way between *Exact Match* and *Any Match* (for D0 and TB) (and 40% for LEM).

Generation for MT Evaluation We evaluate the oracularly modified output using several MT evaluation metrics. Table 4.4 shows the difference in scores between the original MT output and the modified one. There are ≈7 and 10.5, points difference in BLEU [Papineni *et al.*, 2002a] and METEOR [Denkowski and Lavie, 2010] scores, respectively. These differences are the upper limits that a system can reach by making better morphological choices on the unigram level. It is important to keep in mind that some of these improvements are very hard to achieve and some are incorrect linguistically although they maximize the reference matching.

English	Erdogan states Turkey to reject any pressures to urge it to recognize Cyprus.
Reference	أردوغان يؤكد بأن تركيا سترفض أي ضغوطات لحثها على الاعتراف بقبرص . *ÂrdwγAn yŵkd bÂn trkyA strfD Ây DγwTAt lHθhA ςlý AlAςtrAf bqbrS .*
MT Output	اردوجان أن تركيا دولة ترفض أي ضغط لدفعها للاعتراف بقبرص . *ArdwjAn Ân trkyA dwlħ trfD Ây DγT ldfςhA llAςtrAf bqbrS .*
Modified MT	اردوجان بأن تركيا دولة سترفض أي ضغوطات لدفعها الاعتراف بقبرص . *ArdwjAn bÂn trkyA dwlħ strfD Ây DγwTAt ldfςhA AlAςtrAf bqbrS .*

MT Output	ArdwjAn	Ân	trkyA	dwl*ħ*	trfD	Ây	DγT	ldfςhA	llAςtrAf	bqbrS	.
Modified MT	ArdwjAn	bÂn	trkyA	dwl*ħ*	strfD	Ây	DγwTAt	ldfςhA	AlAςtrAf	bqbrS	.
Match Category	UM	LEM	Exact	UM	LEM	Exact	LEM	UM	LEM	Exact	Exact
MT Features		Part:ϕ			Part:ϕ		Gen:M,Num:S		Part:*li*+		
Reference Features		Part:*bi*+			Part:*sa*+		Gen:F,Num:P		Part:ϕ		

Table 4.1: AMEANA word-by-word error analysis. The first four rows specify the English input, Arabic reference, MT output and oracularly modified MT output, respectively. The second half of the table lists every word in the MT output (column 1) with the reference word used to modify it (column 2). Column 3 specifies the reference-match category: exact match indicates the MT output word appears in the reference; unmatchable indicates no match is found; and lemma match indicates a lemma-level match. For lemma match cases, the differences in MT output and reference morphological features are specified in columns 4 and 5.

	D0	TB	LEM
Output Word Count	28,126	28,816	28,759
Exact Match (%)	58.0	59.0	38.7
Lemma Match (%)	13.9	13.3	33.8
Any Match (%)	72.0	72.3	72.6
Unmatchable (%)	28.0	27.7	27.4

Table 4.2: Unigram analysis of three English-to-Arabic SMT systems

	(a)			(b)		
	Error Type %			**Match F-score %**		
	D0	TB	LEM	D0	TB	LEM
PATB Clitics	52.9	53.6	24.3	63.9	65.3	64.4
Conjunction	20.2	18.6	7.4	68.4	69.9	70.1
Particle	23.1	24.3	10.5	68.0	69.2	69.0
Pronoun	15.1	15.7	8.7	69.1	70.3	69.6
Other Features	61.1	57.8	84.6	62.8	64.7	43.9
Determiner	31.0	29.7	60.0	66.9	68.4	52.3
Gender	14.3	12.8	20.5	69.2	70.7	65.7
Number	11.8	10.8	14.0	69.6	71.0	67.9
Person	4.0	4.0	3.4	70.7	71.9	71.4
Stem	3.6	4.1	3.9	70.7	71.9	71.3
Case	3.5	3.0	2.4	70.7	72.0	71.8
Aspect	2.2	2.1	3.1	70.9	72.1	71.5
State	1.0	0.8	0.8	71.1	72.3	72.3
Mood	0.8	0.7	0.4	71.1	72.3	72.5
Voice	0.2	0.2	0.4	71.2	72.4	72.5
Exact (Lemma+Feature) Match				57.4	59.1	38.7
Any (Lemma) Match				71.2	72.4	72.6

Table 4.3: (a) Comparison between the different morphological errors in the MT output in terms of their percentage of the total number of morphological errors and the percentage of total words in the given document. (b) Comparison of F-scores of words matching between the output and the reference for a list of morphological features.

	Basic MT Output			**Oracle MT Output**		
	D0	TB	LEM	D0	TB	LEM
BLEU	25.5	29.5	10.2	33.4	35.6	35.7
METEOR	42.2	45.7	22.6	53.6	55.4	55.4

Table 4.4: Comparison between the original and the modified MT output in BLEU and METEOR metrics. METEOR is used in language-independent mode.

Chapter 5

Lexical Translation versus Morphology Generation

In this chapter, we address these challenges through different modeling methods. In our approach, morphological features can be modeled as part of the core translation process mapping source tokens to target tokens. Alternatively these features can be generated using target monolingual context as part of a separate generation (or post-translation inflection) step. Finally, the features can be predicted using both source and target information in a separate step before generation. We focus in our experiments on English-Arabic SMT and we work on three morphological features that we found, through a manual error analysis, to be most problematic for English-Arabic SMT: gender, number and the determiner clitic. In our approach, the process of translating English words to Arabic words is broken into a pipeline consisting of four steps:

- **Lexical Translation** from English words to tokenized Arabic lemmas and any subset of Arabic linguistic features.
- **Morphology Prediction** of linguistic features to inflect Arabic lemmas.
- **Morphology Generation** of inflected Arabic tokens from Arabic lemmas and any subset of Arabic linguistic features.
- **Detokenization** of inflected Arabic tokens into surface Arabic words.

Arabic tokenization and lemmatization are done before training the translation models. Both lexical translation and generation are implemented as phrase-based SMT systems [Koehn *et al.*, 2007b]. Morphology prediction is an optional step implemented using a supervised discriminative learning model. Generation can be done from lemmas and any subset of Arabic inflectional features. Detokenization simply stitches the words and clitics together as a post-processing step [Badr *et al.*, 2008; El Kholy and Habash, 2010a].

We build on our resolutions from Chapter 3 and focus on the question of how to improve the translation of tokenized words using deeper representations, namely lemmas and features. Within our framework, we can model the translation of different Arabic linguistic features as part of the lexical translation step, as part of the generation step, or model them using an independent morphology prediction step. Some features, such as clitics, can be modeled well through simple tokenization and detokenization (which can be thought of as part of lexical translation).

We use the best performing tokenization scheme (PATB) and the best detokenization technique on the output as our baseline (discussed in Chapter 3). Consequently, in this section we focus on the first three components of the pipeline and we keep the tokenization a constant across all experiments. We study different options of including three morphological features (GEN, NUM and DET) in the first three steps of the pipeline and their implications on the quality of English-to-Arabic SMT. These three features are considered the most problematic from our error analysis in Chapter 4. We discuss the three steps in the following subsections.

Our approach is based on an aggregate model of translation from English into Arabic. This model consists of a pipeline of five components including pre and post processing, lexical translation, morphology prediction and generation. Figure 5.1 show the full pipeline. We will discuss each component separately in the following sections but the main idea here is to separate between lexical translation and morphology generation. In essence, the generation of the different morphological features can be done in any of the components in the pipeline.

5.1 Lexical Translation

Lexical translation is the first step in our decoding pipeline. It is trained on pre-processed text: tokenized, lemmatized and disambiguated Arabic words and English words (with limited processing)

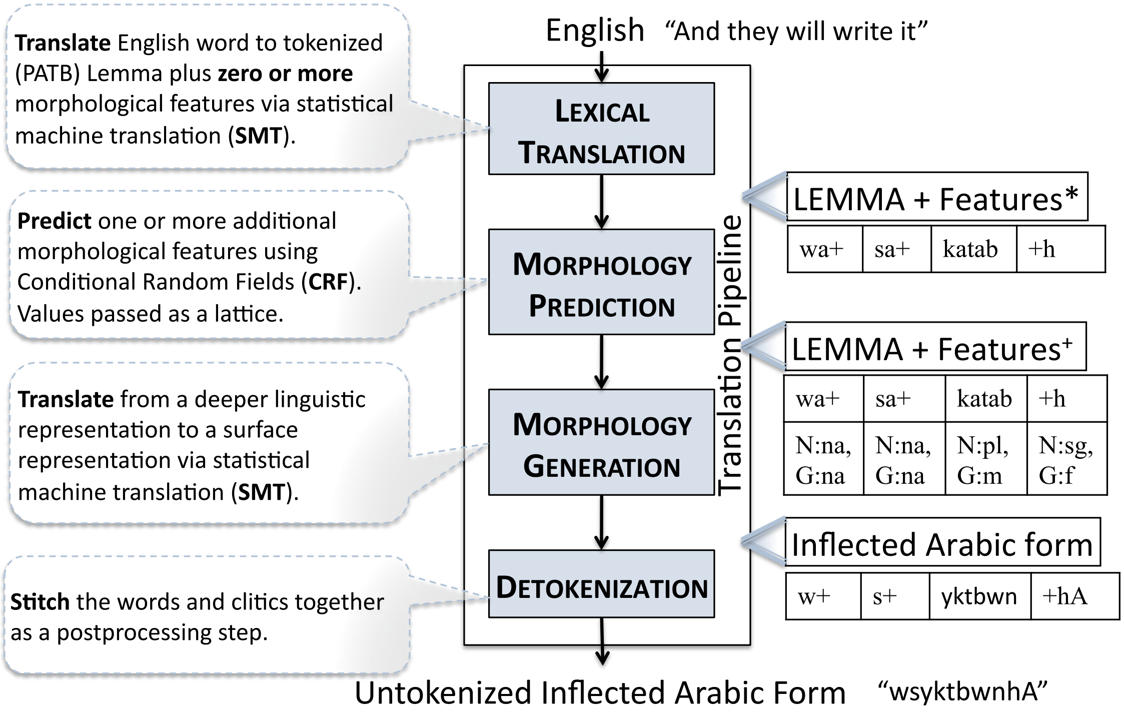

Translate English word to tokenized (PATB) Lemma plus **zero or more** morphological features via statistical machine translation (**SMT**).
Predict one or more additional morphological features using Conditional Random Fields (**CRF**). Values passed as a lattice.
Translate from a deeper linguistic representation to a surface representation via statistical machine translation (**SMT**).
Stitch the words and clitics together as a postprocessing step.
English "And they will write it"
LEXICAL TRANSLATION
MORPHOLOGY PREDICTION
MORPHOLOGY GENERATION
DETOKENIZATION
Translation Pipeline
LEMMA + Features*
wa+ | sa+ | katab | +h
LEMMA + Features+
wa+ | sa+ | katab | +h
N:na, G:na | N:na, G:na | N:pl, G:m | N:sg, G:f
Inflected Arabic form
w+ | s+ | yktbwn | +hA
Untokenized Inflected Arabic Form "wsyktbwnhA"

Figure 5.1: Lexical Generation vs. Morphology Generation Pipeline.

and their POS tags. We use an SMT system to translate from English words (ENGWORD) and POS tags (POS) to tokenized Arabic lemmas (ARALEM) plus zero or more morphological features. We use an abstract representation for the morphological features so that each word is represented as a lemma and a set of feature-value pairs. Table 5.1 shows a sample sentence in the above-mentioned representations. This way we simplify the translation task by targeting a less complex output. The key point here is to keep the morphological features that help the translation task and then try to generate the rest of the morphological features and inflected forms in later steps. The output of lexical translation is input to the morphological generation step directly or is first enriched by additional morphological features predicted in the morphology prediction step.

5.2 Morphology Prediction

Morphology prediction takes the output of lexical translation and tries to enrich it by predicting one or more morphological features. Unlike Toutanova *et al.*, who predict full inflected forms and [Clifton and Sarkar, 2011] who predict morphemes, we predict morphological feature. This task is,

Representation	Example
ENGWORD	saddam hussein 's half-brother refuses to return to iraq
ENGWORD+POS	saddam#NN hussein#NN 's#POS half-brother#NN refuses#VBZ to#TO return#VB to#TO iraq#NN
ARALEM	Âax γayor šaqiyq li+ Sad∼Am Husayon rafaD ςawodaħ Iilaý ςirAq
ARALEM+DET	Âax#det γayor#0 šaqiyq#det li+#na Sad∼Am#0 Husayon#0 rafaD#0 ςawodaħ#det Ǎilaý#na ςirAq#det
Arabic Tokenized	AlÂx γyr Alšqyq l+ SdAm Hsyn yrfD Alςwdħ Ǎlý AlςrAq
Arabic Script	الأخ غير الشقيق لصدام حسين يرفض العودة الى العراق

Table 5.1: A sample sentence showing the different representations used in our experiments.

in sense, a form of POS tagging. However, unlike typical tagging, which is done on fully inflected word forms, this task is applied to uninflected or semi-inflected forms – lemmas with zero or more morphology features. As such, we do not expect it to do as well as normal POS tagging/morphology disambiguation for Arabic [Habash and Rambow, 2005].

We use a Conditional Random Field (CRF) toolkit [Lafferty *et al.*, 2001] to train a prediction module with a variety of learning features (not to be confused with the tagged linguistic features). We also make use of the alignment information produced by the MT system in the lexical translation step to get the equivalent aligned English word of each translated word. We then use this information in addition to some syntactic information on the English side as CRF learning features.

We group the CRF learning features into two sets: *Basic* and *Syntax*. The *Basic* features consist of the Arabic output from the lexical translation step (lemma plus certain features), the equivalent aligned English word, English POS and English context (+/- two words). The *Syntax* features consist of the English parent word in a dependency tree, the dependency relation and the equivalent Arabic output word of the English parent. English is parsed using the Stanford Parser [Klein and Manning, 2003].

In training the CRF model, we use the same data used in training the lexical translation step (Section 5.4). We create three datasets from this data. The first is the original gold data where we train the CRF module on clean Arabic text and gold feature values that are determined using a state-of-the-art POS tagger for Arabic [Habash and Rambow, 2005]. Although the automatic tagging does produce errors, we still call this data set *gold* since the Arabic is correctly inflected naturally occurring text. The second dataset is created by translating the whole data using the translation

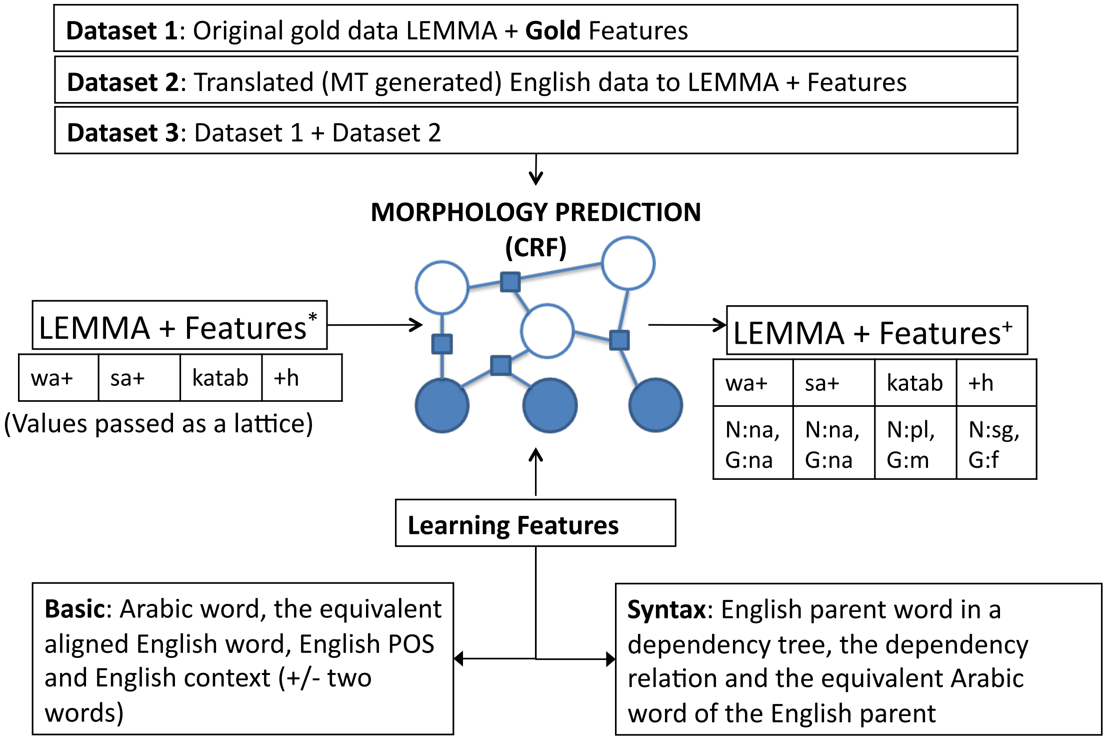

Dataset 1: Original gold data LEMMA + **Gold** Features
Dataset 2: Translated (MT generated) English data to LEMMA + Features
Dataset 3: Dataset 1 + Dataset 2
MORPHOLOGY PREDICTION (CRF)
LEMMA + Features*
wa+
sa+
katab
+h
(Values passed as a lattice)
LEMMA + Features+
wa+ N:na, G:na
sa+ N:na, G:na
katab N:pl, G:m
+h N:sg, G:f
Learning Features
Basic: Arabic word, the equivalent aligned English word, English POS and English context (+/- two words)
Syntax: English parent word in a dependency tree, the dependency relation and the equivalent Arabic word of the English parent

Figure 5.2: Morphology Prediction.

model created by the lexical translation step. The intuition here is to model lexical translation errors by training the CRF models on data similar in quality to its expected input. The last dataset is the combination of gold and translated dataset.

Table 5.2 shows the accuracy of the CRF module on a test set of 1000 sentences. CRF in general achieves a high accuracy across the different training datasets and the different training parameters. Using translated data does not outperform using gold data; however, the accuracy of predicting NUM and GEN seems to benefit from adding the translated data to the gold data. That could be explained by the fact that NUM and GEN are more affected by translation adequacy unlike DET which is more coupled with translation fluency. Overall the results are about 10-14% absolute lower than MADA [Habash and Rambow, 2005] tagging of the same features on fully inflected text; and are 20-30% absolute better than a degenerate baseline using the most common feature value.

The morphology prediction step produces a lattice with all possible feature values each having an associated confidence score. The morphology generation module discussed next will decide on the best option.

Prediction Training		**Predicted Feature Accuracy**		
Data Set	**Model**	Gen	Num	Det
Gold	**Basic**	84.65	88.76	**88.00**
	Basic+Syntax	84.22	89.11	87.85
Translated	**Basic**	84.46	86.11	85.98
	Basic+Syntax	84.08	86.79	85.41
Gold +Translated	**Basic**	**85.96**	89.43	87.40
	Basic+Syntax	85.49	**89.52**	86.91

Table 5.2: Accuracy (%) of feature prediction starting from Arabic lemmas. A most-common-tag degenerate baseline would yield 67.4%, 70.6% and 59.7% accuracy for Gen, Num, and Det, respectively. Reported MADA classification accuracy starting from fully inflected Arabic is as follows: Gen 98.2% , Num 98.8%, Det 98.3%

5.3 Morphology Generation

Morphology generation maps Arabic lemmas (AraLem) plus morphological features to Arabic inflected forms. This step is implemented as an SMT system that translate from a deeper linguistic representation to a surface representation of each token. This step is conceptually similar to the generation expansion component in Factored SMT, but it is implemented as a complete SMT system. The main advantage of this approach is that the training data is not restricted to parallel corpora. We can use all the monolingual data we have in building the system. We avoid the alignment and symmetrization errors by constructing a one-to-one alignment matrix instead of building it through an EM algorithm (provided by Moses SMT toolkit).

To evaluate the performance of this approach in generating Arabic inflected forms, we built several SMT systems translating from AraLems plus zero or more morphological features to Arabic inflected form. We use the same tools and setup as discussed in Section 5.4. Table 5.3 shows the BLEU scores of generating the MT05 set starting from Arabic lemmas plus different morphological features (Gen, Num, Det), and their combinations. As expected, the more features are included

Gold Generation Input	**BLEU%**
ARALEM	82.2
ARALEM+DET	86.6
ARALEM+NUM	86.9
ARALEM+GEN	87.3
ARALEM+GENNUM	90.2
ARALEM+GENNUMDET	94.8

Table 5.3: Results of generation from gold ARALEM plus different sets of morphological features. Results are in (% BLEU) on the MT05 set.

the better the results. Here comes the trade off between the lexical translation quality and morphological generation. The BLEU scores are very high because the input is golden in terms of word order and lemma choice. These scores should be seen as the upper limit on correctness that can be expected from this step, rather than its actual performance in an end-to-end pipeline.

The morphology generation step can take the output of lexical translation directly or after predicting certain morphological features using the morphology prediction step.

5.4 Evaluation

In this section, we present our results comparing the modeling of GEN, NUM and DET features, first as part of lexical translation versus morphological generation, and then as part of morphological prediction versus morphological generation. We also present results on a blind test set MT06, a much larger training corpus, and discuss our findings.

5.4.1 Experimental Setup

All of the training data we use is available from the Linguistic Data Consortium (LDC).[1] We use an English-Arabic parallel corpus of about 142K sentences and 4.4 million words for translation model

[1] http://www.ldc.upenn.edu

training data. The parallel text includes Arabic News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18). Word alignment is done using GIZA++ [Och and Ney, 2003b]. For language modeling, we use 200M words from the Arabic Gigaword Corpus (LDC2007T40) together with the Arabic side of our training data. We used 5-grams for all LMs implemented using the SRILM toolkit [Stolcke, 2002].

MADA is used to tokenize the Arabic text and produce lemmas and their accompanied morphological features. English preprocessing simply includes down-casing, separating punctuation and splitting off "'s".

All experiments are conducted using the Moses phrase-based SMT system [Koehn *et al.*, 2007b]. The decoding weight optimization was done using a set of 300 sentences from the 2004 NIST MT evaluation test set (MT04). The tuning is based on tokenized Arabic without detokenization. We use a maximum phrase length of size 8. We report results on the 2005 NIST MT evaluation set (MT05). These test sets were created for Arabic-English MT and have four English references. We arbitrarily picked the first English reference to be source and used the Arabic source as the only reference. We evaluate using BLEU-4 [Papineni *et al.*, 2002a].

Our baseline replicates the work presented in Chapter 3, which concluded that tokenizing Arabic into the PATB tokenization scheme is optimal for phrase-based SMT models. The baseline BLEU score is 29.48% using exactly the same data sets used in the rest of the experiments.

5.4.2 Translation vs. Generation

We compare the performance of translating English and English plus POS into Arabic lemmas plus different morphological feature combinations followed by generation of the final Arabic inflected form using the morphology generation step directly under the same conditions. The results are presented in Table 5.4. The best performer across all conditions is translating English words to Arabic lemmas plus DET. This is the only setup that beats the baseline system. The difference in BLEU scores between this setup and the baseline is statistically significant above the 95% level. Statistical significance is computed using paired bootstrap resampling [Koehn, 2004]. This shows the importance of DET in lexical translation. English POS oddly does not help. This is perhaps a result of the added sparsity in how we modeled them (as ENGWORD+POS). It is possible a factored MT model can give different results. We plan to explore this question in the future.

Input	A'	**BLEU%**
ENGWORD	ARALEM	**29.5**
ENGWORD+POS	ARALEM	29.3
ENGWORD	ARALEM+NUM	29.0
ENGWORD+POS	ARALEM+NUM	28.5
ENGWORD	ARALEM+GEN	28.8
ENGWORD+POS	ARALEM+GEN	28.7
ENGWORD	ARALEM+DET	**30.1**
ENGWORD+POS	ARALEM+DET	29.3
ENGWORD	ARALEM+GENNUM	28.8
ENGWORD+POS	ARALEM+GENNUM	28.7
ENGWORD	ARALEM+GENNUMDET	29.2
ENGWORD+POS	ARALEM+GENNUMDET	29.0

Table 5.4: End-to-end MT results for different settings of English input and Intermediate Arabic. Results are in (% BLEU) on our MT05 set.

5.4.3 Prediction vs. Generation

We compare results of two translation settings and a variety of added predicted features. The results are presented in Table 5.5. We can see from the results that using predicted GEN by itself does not help across the board yet it could be helpful when combined with other features. It also seems that predicting NUM when lexical translation is done with lemmas only helps the performance but that is not the case when the lexical translation is done using Lemma plus DET. Another observation is that combining GEN and NUM degrades the overall performance more than the GEN by itself; however, we get the best scores when DET is combined with them. This shows that some synergies come out when different features are combined together even if they perform badly on their own. The only fact that seems very robust is that translating English to Lemma plus DET and then predicting both GEN and NUM gives the highest scores. Predicting features using models trained on translated texts seem to also consistently do better than using models that are trained on original Arabic. The

Translation	EngWord→AraLem					EngWord→AraLem+Det		
No Prediction	29.5					30.1		
Prediction	**Predicted Morphological Features**							
Training	Gen	Num	Det	Gen+Num	Gen+Num+Det	Gen	Num	Gen+Num
Gold Basic	28.6	29.5	29.7	28.4	29.8	29.9	29.9	30.4
+Syntax	28.6	29.5	29.7	28.4	29.9	29.9	29.9	30.4
Trans Basic	28.9	29.6	**29.8**	28.3	29.9	29.9	29.9	30.4
+Syntax	28.9	29.6	**29.8**	28.8	29.9	**30.0**	29.9	30.4
Gold+Trans Basic	**29.0**	29.6	29.8	**28.8**	30.0	30.0	**30.0**	30.4
+Syntax	28.9	**29.6**	29.8	28.8	**30.0**	30.0	**30.0**	**30.4***

Table 5.5: End-to-end MT results for two translation settings and a variety of added predicted features. Results are in (% BLEU) on our MT05 set. The best result in each column is bolded. The best overall result is marked with *.

best result obtained is statistically significant compared with the best reported score in the previous section (AraLem+Det translation).

5.4.4 Blind Test

We performed a blind test using the 2006 NIST MT evaluation set (MT06) and compared the results to (MT05). MT06 is a harder set to translate than MT05. However, the relative performance is maintained (around 3% relative BLEU) as shown in Table 5.6. Translating through Lemma plus Det and then predicting Gen and Num is still the best option.

We found out that the percentage of the Exact matches increases while the Unmatched words decreases as an inherent effect of using more data but the Lemma match percentage decreases across the different options. This shows the applicability of our approach in predicting the morphology in the case of absence of exact evidence in the training data.

Model	MT05			MT06		
	BLEU	METEOR	TER	BLEU	METEOR	TER
Baseline	29.5	46.3	52.7	19.1	32.8	68.6
Factored	30.1	46.6	52.3	19.3	32.6	67.1
EngWord→AraLem	29.5	46.2	53.3	18.9	31.7	67.0
EngWord→AraLem+Det	30.1	46.5	52.5	19.4	32.4	67.5
EngWord→AraLem+Det with Gen+Num Prediction	**30.4**	**46.8**	**52.0**	**19.7**	**32.9**	**67.0**

Table 5.6: Results comparing our baselines and best performing setup on MT05 and MT06 (blind). Results are in (BLEU, METEOR and TER).

5.4.5 Scaling Up

We performed experiments using a larger amount of data (15 times the size of the original dataset; also available from the LDC). Not surprisingly, the effect of our approach diminished. Although the general trends remained the same, none of the alternative settings were able to beat the baseline. We compared the percentage of the Exact Match, Lemma Match and Unmatchable words with the reference of the basic and scaled up systems. We found out that the percentage of exact matches increases while the percentage of unmatched words decreases. This is not a surprising result of using more data. The lemma match percentage decreases across the different systems. This suggests that our approach is more effective for conditions with low and medium resource size.

5.5 Conclusions

The generation of fully inflected forms from uninflected lemmas (Table 5.4) in a purely monolingual setting such as our morphological generation step is very hard – we get only 82.2% BLEU starting with gold lemmas. Adding different combinations of gold values of the three most problematic morphological features improves the score by over 12% absolute BLEU to a higher performance ceiling (94.8% BLEU).

Automatically modeling these features at a high accuracy for SMT, however, turns out to be rather hard. If we consider using them as part of the translation step together with lemmas, we find that they almost always hurt the end-to-end (translation-generation) MT system except for the DET feature which improves over an inflected tokenized baseline by about 0.6% BLEU.

Predicting the feature values using an independent supervised learning step that has access to the English word, POS and syntax features produces accuracy scores ranging in mid to high 80s%. Comparing the prediction accuracy of GEN, NUM and DET (Table 5.2), we find NUM is the easiest to predict, followed by DET and then GEN. This makes sense given the information provided from English, which is inflected for NUM, but not GEN.

The results in Table 5.5 show that DET, as a single feature, helps more when it is part of the translation step (30.1 BLEU) compared to being predicted (29.7∼29.8). In both cases, it fares better than simply leaving determining DET to the generation step (29.5).

Neither GEN nor NUM, as single features, help much (or at all) over the baselines when part of the translation step or when predicted. However, when both are combined with DET they consistently help only when GEN and NUM are predicted, not translated. It is possible that the lower performance we see as part of the translation is a product of how we translate: we do not factor these features in the translation – a direction we plan to consider in the future. We postulate that the prediction step helps because it has access to more information than used in our translation step, e.g., source language syntax.

Chapter 6

Phrase Pivoting Quality and Recall Maximization

In previous chapters, we focused on the main building block of the pivoting process which is the translation model. In particular when targeting a morphologically rich language. In this chapter, we look at the bigger picture and we discuss our approaches to improve the phrase pivoting process. In the standard phrase-pivoting approach, many phrase pairs between source and target languages are not generated because of bad matching of pivot phrases. Additionally, the size of the newly created pivot phrase table is very large. Many of the produced phrase pairs are of low quality which affects the translation choices during decoding and the overall translation quality. To overcome these problems, we try to maximize both precision and recall of the pivoting process where we try to add phrase pairs to the final translation model (more coverage) and make sure they are of good quality (precision).

We discuss two language independent techniques. One of the techniques works on the level of the word alignment symmetrization. Another approach to maximize the precision is based on connectivity scores between the source and target phrase pairs based on the alignment information propagated from the source-pivot and pivot-target systems. We start by explaining the baseline phrase-pivot system that is the base for the rest of the dissertation discussion. We then discuss each approach separately.

6.1 Phrase Pivoting Baseline

In this section, we show the basic pivoting setup that we use as baseline for all the pivoting experiments in this chapter and following one. We start with the linguistic preprocessing decision that we made for different languages and then we illustrate a filtering process that we conduct before performing phrase pivoting to overcome the massive combinatorial expansion in generating source to target phrase pairs.

6.1.1 Linguistic Preprocessing

As we mentioned earlier, we work with four different languages; Arabic, Hebrew, Persian and English. We always target Arabic as the final output and we start with either Persian or Hebrew where there are limited resources for those language. While we use English as the pivot language due to the abundance of resources with all the other languages. We present our choices for preprocessing the data of each language which is consistent across all pivoting experiments. For Arabic, we follow our resolution from Chapter 3 and use the PATB tokenization scheme [Maamouri *et al.*, 2004b] in our experiments which separates all clitics except for the determiner clitic *Al*+. We use MADA v3.1 [Habash and Rambow, 2005; Habash *et al.*, 2009] to tokenize the Arabic text.

We only evaluate on detokenized and orthographically correct (enriched) output as discussed in Chapter 3. For Hebrew, we use the best preprocessing scheme for Hebrew (HTAG) identified by [Singh and Habash, 2012] which is very close to the Arabic PATB tokenization scheme. Furthermore, for Persian, we use Perstem [Jadidinejad *et al.*, 2010] for segmenting Persian text. Perstem mainly focuses on verbs which inflect for 14 different features. Finally for English, it is much easier for preprocessing because it is morphologically poor and barely inflects for number, person and tense. English preprocessing simply includes down-casing, separating punctuation and splitting off "'s".

6.1.2 Phrase Pairs Filtering

As a result of phrase pivoting (discussed in more details in Chapter 2.5, the final translation model between the source and target language is usually huge due to the combinatorial expansion from almost multiplying two translation models. Table 6.1 illustrates how big a pivot phrase table can

	Training Corpora	**Phrase Table**	
Translation Model	**Size**	**# Phrase Pairs**	**Size**
Persian-English	≈4M words	96,04,103	1.1GB
English-Arabic	≈60M words	111,702,225	14GB
Pivot_Persian-Arabic	N/A	39,199,269,195	≈2.5TB

Table 6.1: Translation Models Phrase Table comparison in terms of number of line and sizes for Persian-Arabic SMT.

reach when we combine two relatively small models through phrase pivoting.

The main idea of the filtering process is to select the top [*n*] English candidate phrases for each source phrase from the Source-English phrase table and similarly select the top [*n*] target phrases for each English phrase from the English-Target phrase table and then perform the pivoting process to create a pivoted Source-Target phrase table. To select the top candidates, we first rank all the candidates based on the log linear scores computed from the phrase translation probabilities and lexical weights of each system multiplied by the optimized decoding weights then we pick the top [*n*] pairs.

6.1.3 Evaluation

We compare the performance of sentence pivoting against phrase pivoting with different filtering thresholds for Persian-Arabic pivot translation model. The results are presented in Table 6.2. In general, phrase pivoting outperforms the sentence pivoting even when we use a small filtering threshold of size 100. Moreover, the higher the threshold is the better the performance will be but with a diminishing gain. In all our experiments, the default filtering threshold in our baselines is 1K and it is consistent across all settings. We use the suffix “_F1K" to indicate 1K filtering.[1]

[1] We tried to do an experiment with all the options without filtering but we couldn’t because of computational limitations.

Pivot Scheme	**BLEU**	**METEOR**	**TER**
Sentence Pivoting	19.2	36.4	62.7
Phrase_Pivot_F100	19.4	37.4	61.4
Phrase_Pivot_F500	20.1	38.1	59.0
Phrase_Pivot_F1K	**20.5**	**38.6**	**58.8**

Table 6.2: Comparing sentence pivoting against phrase pivoting with different filtering thresholds (100/500/1000) for Persian-Arabic SMT.

6.2 Alignment Connectivity Strength

As we mentioned in the beginning of this chapter, one of the main challenges in phrase pivoting is the very large size of the induced phrase table. It becomes even more challenging if either the source or target language is morphologically rich. The number of translation candidates (fanout) increases due to ambiguity and richness which in return increases the number of combinations between source and target phrases. Since the only criteria of matching between the source and target phrase is through a pivot phrase, many of the induced phrase pairs are of low quality. These phrase pairs unnecessarily increase the search space and hurt the overall quality of translation.

To solve this problem, we introduce two language-independent features which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. We call these features *connectivity strength features*.

6.2.1 Connectivity Strength Features

"Connectivity Strength Features" consists of two scores, Source Connectivity Strength (SCS) and Target Connectivity Strength (TCS). These two scores are similar to precision and recall metrics. They depend on the number of alignment links between words in the source phrase to words of the target phrase. SCS and TSC are defined in equations 6.1 and 6.2 where $\mathcal{S} = \{i : 1 \leq i \leq S\}$ is the set of source words in a given phrase pair in the pivot phrase table and $\mathcal{T} = \{j : 1 \leq j \leq T\}$ is the set of the equivalent target words. The word alignment between $\mathcal{S}$ and $\mathcal{T}$ is defined as $\mathcal{A} = \{(i, j) : i \in \mathcal{S} \text{ and } j \in \mathcal{T}\}$.

$$SCS = \frac{|\mathcal{A}|}{|\mathcal{S}|} \tag{6.1}$$

$$TCS = \frac{|\mathcal{A}|}{|\mathcal{T}|} \tag{6.2}$$

We get the alignment links by projecting the alignments of source-pivot to the pivot-target phrase pairs used in pivoting. If the source-target phrase pair are connected through more than one pivot phrase, we take the union of the alignments. Figure 6.1 illustrates the projection process to get the alignment links between words of source and target phrases.

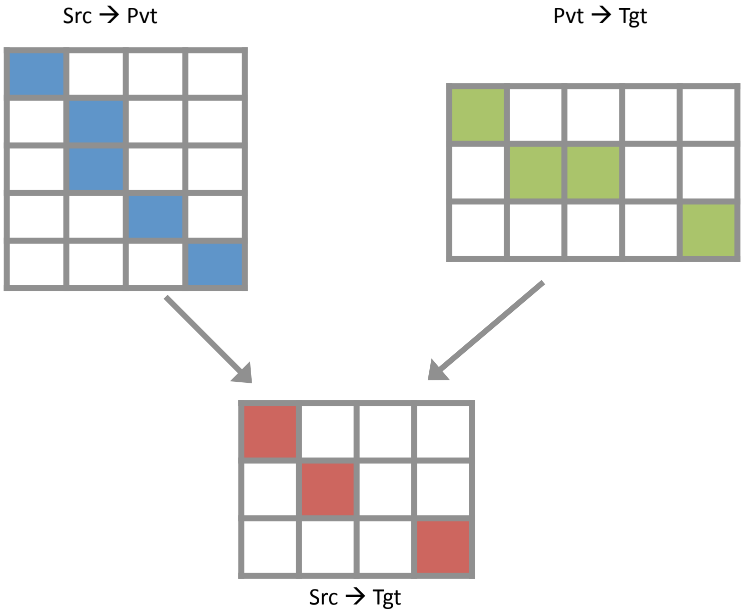


Figure 6.1: An illustration of the projection of alignment links from source-pivot to the pivot-target phrase pairs used in pivoting

In contrast to the aggregated values represented in the lexical weights and the phrase probabilities, connectivity strength features provide additional information by counting the actual links between the source and target phrases. They provide an independent and direct approach to measure how good or bad a given phrase pair are connected.

Figure 6.2 and 6.3 are two examples (one good, one bad) of Persian-Arabic phrase pairs in a pivot phrase table induced by pivoting through English.[2] In the first example, each Persian word is aligned to an Arabic word. The meaning is preserved in both phrases which is reflected in the SCS and TCS scores. In the second example, only one Persian word is aligned to one Arabic word in the equivalent phrase and the two phrases convey two different meanings. The English phrase is not a good translation for either, which leads to this bad pairing. This is reflected in the SCS and TCS scores which are presented in the captions of the figures.

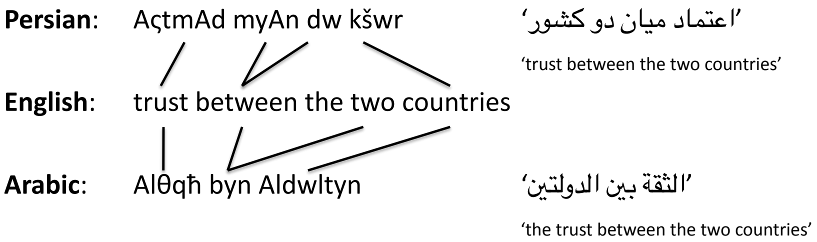

Persian: AςtmAd myAn dw kšwr
'اعتماد میان دو کشور'
'trust between the two countries'
English: trust between the two countries
Arabic: Alθqħ byn Aldwltyn
'الثقة بين الدولتين'
'the trust between the two countries'

Figure 6.2: An example of strongly connected Persian-Arabic phrase pair through English. All Persian words are connected to one or more Arabic words. SCS=1.0 and TCS=1.0.

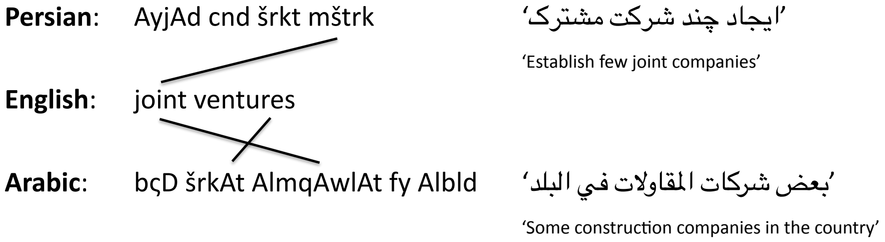

Persian: AyjAd cnd šrkt mštrk
'ایجاد چند شرکت مشترک'
'Establish few joint companies'
English: joint ventures
Arabic: bςD šrkAt AlmqAwlAt fy Albld
'بعض شركات المقاولات في البلد'
'Some construction companies in the country'

Figure 6.3: An example of weakly connected Persian-Arabic phrase pairs through English. Only one Persian word is connected to an Arabic word. SCS=0.25 and TCS=0.2.

6.2.2 Evaluation

In our pivoting experiments, we work on two language pairs, Persian-Arabic and Hebrew-Arabic, pivoting through English. The English-Arabic parallel corpus is about 2.8M sentences (≈60M

[2] We use the Habash-Soudi-Buckwalter Arabic transliteration [Habash *et al.*, 2007] in the figures with extensions for Persian as suggested by [Habash, 2010].

words) available from LDC[3] and GALE[4] constrained data. We use an in-house Persian-English parallel corpus[5] of about 170K sentences and 4M words. The Hebrew-English corpus is about ($\approx$ 1M words) and is available from sentence-aligned corpus produced by [Tsvetkov and Wintner, 2010].

Word alignment is done using GIZA++ [Och and Ney, 2003b]. For Arabic language modeling, we use 200M words from the Arabic Gigaword Corpus [Graff, 2007] together with the Arabic side of our training data. We use 5-grams for all language models (LMs) implemented using the SRILM toolkit [Stolcke, 2002]. For English language modeling, we use English Gigaword Corpus with 5-gram LM using the KenLM toolkit [Heafield, 2011].

All experiments are conducted using the Moses phrase-based SMT system [Koehn *et al.*, 2007b]. We use MERT [Och, 2003b] for decoding weight optimization. For Persian-English translation model, weights are optimized using a set 1000 sentences randomly sampled from the parallel corpus while the Hebrew-English weights are optimized using a tuning set of 517 sentences developed by [Shilon *et al.*, 2010]. The common English-Arabic translation model weights are optimized using a set of 500 sentences from the 2004 NIST MT evaluation test set (MT04). The optimized weights are used for ranking and filtering.

We use a maximum phrase length of size 8 across all models. We report results on an in-house Persian-Arabic evaluation set of 536 sentences with three references. While for Hebrew-Arabic, we report results on an evaluation set of 300 sentences with three references developed by [Shilon *et al.*, 2010]. We evaluate using BLEU-4 [Papineni *et al.*, 2002a], METEOR v1.4 [Lavie and Agarwal, 2007] and TER [Snover *et al.*, 2006].

6.2.2.1 Connectivity Strength Features Evaluation

In this experiment, we test the performance of adding the connectivity strength features (+*Conn*) to the best performing phrase pivoting model (*Phrase_Pivot_F1K*).

[3] LDC Catalog IDs: LDC2005E83, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006G05, LDC2007E06, LDC2007E101, LDC2007E103, LDC2007E46, LDC2007E86, LDC2008E40, LDC2008E56, LDC2008G05, LDC2009E16, LDC2009G01.

[4] Global Autonomous Language Exploitation, or GALE, is a DARPA-funded research project.

[5] available from SAIC http://www.saic.com

Model	**Persian-Arabic**			**Hebrew-Arabic**		
	BLEU	METEOR	TER	BLEU	METEOR	TER
Phrase_Pivot_F1K	20.5	38.6	70.6	19.8	33.7	64.4
Phrase_Pivot_F1K+Conn	**21.1**	**38.9**	**69.3**	**20.3**	**33.9**	**63.5**

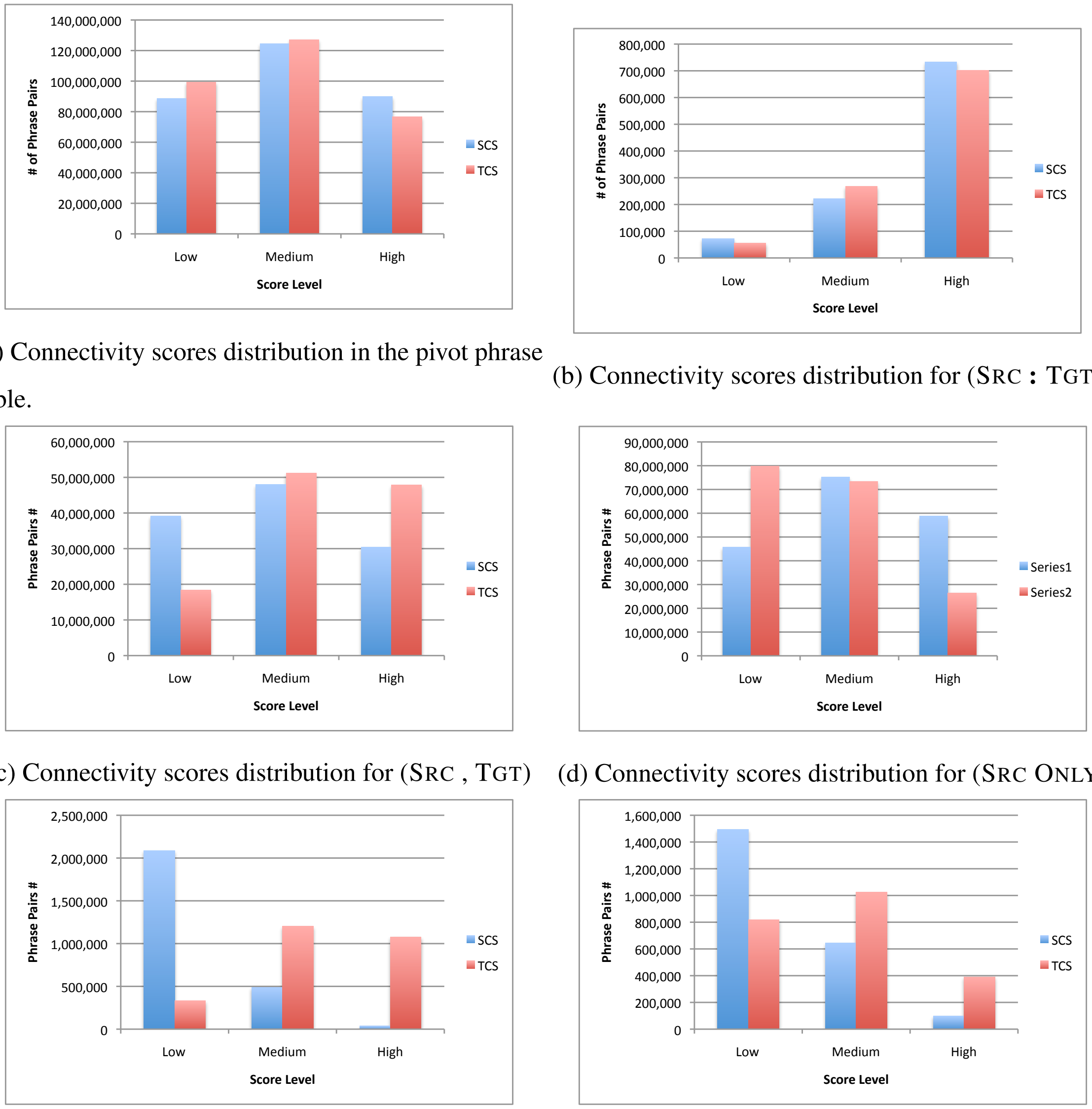
Table 6.3: Connectivity strength features experiment result for Persian-Arabic and Hebrew-Arabic SMT.

The results in Table 6.3 show that we get a good improvement in all three metrics for both models (Pesrian-Arabic and Hebrew-Arabic) by adding the connectivity strength features. The differences in BLEU scores (≈0.6/0.5) between this setup and the baseline is statistically significant above the 95% level. Statistical significance is computed using paired bootstrap resampling [Koehn, 2004].

6.2.3 Intrinsic Evaluation

In this section, we perform an intrinsic evaluation of the performance of the connectivity strength features using external parallel data between the source and target languages. To achieve this goal, we study the correlation between scores and phrase extracted from parallel text. We classify the pivot phrase pairs into five different classes based on the existence of source and/or target phrases in the direct model trained on the parallel data. The first class contains the phrase pairs where the source and target phrases are in the direct system together. The second class is the same as the first class except that the source and target phrases exist but not together as a phrase pair in the direct system. The third, forth and fifth classes are for the existence of source phrase only, target phrase only and neither in the direct system, respectively.

Figure 6.4b shows that when the source and target phrase exist in the direct model, the connectivity scores are mostly high which reflects how the connectivity scores are highly correlated with the quality of the phrase pair. Moreover, Figure 6.5 shows that when the SCS and TCS scores are equal one,i.e. highly connected phrase pairs in both directions, the phrase pairs are mostly classified as both existing together in the direct model. This is also another strong indication that the connectivity scores can be trusted.



(a) Connectivity scores distribution in the pivot phrase table.

(b) Connectivity scores distribution for (SRC : TGT)

(c) Connectivity scores distribution for (SRC , TGT)

(d) Connectivity scores distribution for (SRC ONLY)

(e) Connectivity scores distribution for (TGT ONLY)

(f) Connectivity scores distribution for (NEITHER)

Figure 6.4: Plots of connectivity scores of the different phrase pairs classifications

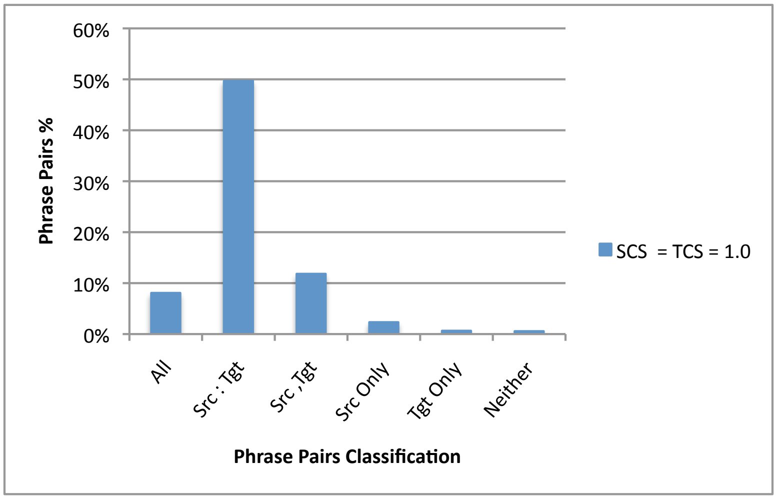


Figure 6.5: Connectivity scores across all phrase pairs categories when the SCS and TCS are equal to one i.e. highly connected phrase pairs in both directions.

6.2.4 Conclusions

We presented an experiment showing the effect of using two language independent features, source connectivity score and target connectivity score, to improve the quality of pivot-based SMT. We showed that these features help to improve the overall translation quality. We also performed intrinsic and extrinsic evaluation to show the effectiveness of our approach.

6.3 Alignment Symmetrization Optimization

In this Section, we focus on word alignment to improve translation quality. Word alignment is an essential step in building an SMT system. The most commonly used alignment models, such as IBM Model serial [Brown *et al.*, 1993a] and HMM [Och and Ney, 2003a], all assume one-to-many alignments. However, the target is to produce a many-to-many word alignment model. A common practice solution in most state-of-the-art MT systems is to create two sets of one-to-many word alignments (bidirectional alignments), source-to-target and target-to-source, and then combine the two sets to produce the final many-to-many word alignment model. This combination process is called “Symmetrization".

We discuss a symmetrization relaxation method targeting phrase-pivot SMT. Unlike the typical symmetrization methods, the process is carried out as an optimization for phrase-pivot SMT and eventually increase the matching on the pivot phrases. We show positive results (1.2 BLEU points) on Hebrew-Arabic phrase-pivot SMT (pivoting through English).

6.3.1 Background

In this section, we briefly describe different symmetrization heuristics. We then explain how symmetrization affects phrase extraction and discuss the motivation for our approach.

6.3.2 Symmetrization Heuristics

The simplest approach is to merge the two directional alignment functions using a symmetrization heuristic to produce a many-to-many alignment matrix [Och *et al.*, 1999; Och and Ney, 2003a; Koehn *et al.*, 2003].

One of the approaches is to take the intersection (I) of the two directional alignments. Intersection alignment matrices are very sparse and express only one-to-one relationship between words. As a result, we get a high precision in alignment due to the agreement of both models and a very low recall.

An alternative approach is to look at the two alignments as containing complementary information. Therefore, the union (U) of the two models can capture all complementary information. Unlike the intersection (I), many-to-many relationship between words are covered and the resulting matrices are dense. As a result, we get the opposite effect of intersection where we have a higher recall of alignment points but at the cost of losing in precision.

Many mid-way solutions between intersection (I) and (U) can be achieved which aim to balance between precision and recall. Some solutions start from high precision intersection points, and progressively add reliable links from the union to increase recall. Other solutions start from a high recall union points and remove unreliable links to increase precision. One of most commonly used heuristic is **Grow-diag-final-and** (GDFA) [Koehn *et al.*, 2003].

The GDFA heuristic is composed of two steps and one constraint. The first step (**Grow-diag**) starts from the intersection of two directional alignments then gradually considers the neighborhood of each alignment point between the source and target words. The considered neighbors of an alignment point at position (i, j) span over the range of $[i-1, i+1]$ for source words and $[j-1, j+1]$

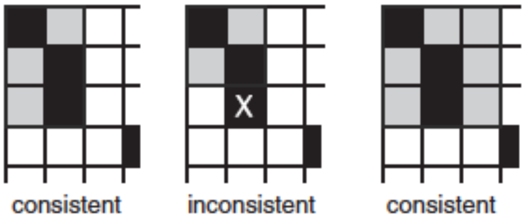


Figure 6.6: Phrase-pairs consistency constraints with word alignment (black squares are alignment points and the shaded area is a proposed phrase pair): The first example from the left obeys the consistency heuristic, which is violated in the second example (one alignment point in the second column is outside the phrase pair). The third example obeys the consistency heuristic despite the fact that it includes an unaligned word on the right.

for target words. Points in this neighborhood are progressively added to the alignment if neither the source word nor the target word is already aligned and the corresponding point exists in the union (U). The second step (**-final**) adds alignment points that are not neighbor intersection alignment points. This is done for alignment points between words, of which at least one is currently unaligned and exists in the union (U). Adding the constraint (**-and**), only allows alignment points between two unaligned words to be added.

6.3.3 Symmetrization vs. Phrase Extraction

There is a direct relationship between the final alignment matrix after symmetrization and the phrase extraction process. One way to look at the role of alignment points in extracting phrases is that they act as constraints for which phrase pairs can be extracted. In the standard heuristic [Koehn *et al.*, 2003] for phrase pair extraction, the extracted phrase pair should be consistent and contain at least one word-based link. In addition, no word inside the phrase pair is aligned to a word outside it. Figure 6.6 shows examples of phrase pairs that obey or violate the consistency constraints.

The consistency constraint leads to an inverse relationship between the number of alignment points and the number of phrase pairs extracted; the fewer alignment points, the more phrase pairs can be extracted. This relationship is not valid in the extreme situation with no alignment points at all; in this extreme case, no phrase pairs are extracted.

A major issue in this heuristic is its sensitivity to word alignment errors. Since the consistency

Algorithm 1 Symmetrization Relaxation Algorithm (starting with union symmetrization). Symbols used are explained in Section 6.3.4.

```
{ generate the list of possible pivot unigram $L_p$}
$A^U_{pt} = \overrightarrow{A_{pt}} \cup \overleftarrow{A_{pt}}$
$A^F_{pt} = A^U_{pt}$
for $(i, j) \in A^F_{pt}$ do
    if $W_i \notin L_p$ then
        $A^F_{pt} = A^F_{pt} - \{(i, j)\}$
    end if
end for
return $A^F_{pt}$
```

constraint is based on the alignment, an error could prevent the extraction of many good phrase pairs. In the context of phrase pivoting, this eventually leads to much less chances to pivot on potential good phrases. This problem motivates our approach to relax the symmetrization process (discussed in Section 6.3.4) and generate new pivot phrases in both systems used in pivoting. These new pivot phrases can connect potential source to target phrase pairs.

6.3.4 Relaxation

In this section, we explain our approach in relaxing the symmetrization process to improve the matching in phrase-pivot SMT. We then discuss our approach in combining the phrase pairs extracted from the basic pivot system and a pivot system using our relaxation approach which leads to our best results.

6.3.5 Symmetrization Relaxation

Our approach is based on two parts. The first part is constructing a list of all possible pivot unigram phrases L_p that can be used in the pivoting process. This can simply be done by getting the intersection of all the pivot unigrams extracted from both the source-pivot and the pivot-target corpora.

In the second part, we start by building two directional alignment models: pivot-to-target $\overrightarrow{A_{pt}}$

	He-En				En-Ar				He-En-Ar	
Symm.	$\|A_{sp}\|$	$\frac{\|A_{sp}\|}{\|A^U_{sp}\|}$	$\|PT_{sp}\|$	$\frac{\|PT_{sp}\|}{\|PT^I_{sp}\|}$	$\|A_{pt}\|$	$\frac{\|A_{pt}\|}{\|A^U_{pt}\|}$	$\|PT_{pt}\|$	$\frac{\|PT_{pt}\|}{\|PT^I_{pt}\|}$	$\|PT_{st}\|$	$\frac{\|PT_{st}\|}{\|PT^I_{st}\|}$
I	0.6M	45%	15.0M	100%	0.7M	57%	11.4M	100%	1707M	100%
U	1.4M	100%	0.9M	6%	1.3M	100%	1.3M	12%	1M	0.1%
U_R	1.2M	89%	1.7M	11%	1.2M	91%	2.3M	21%	245M	14%
GDFA	1.1M	79%	3.0M	20%	1.0M	85%	3.0M	27%	267M	16%
GDFA_R	1.0M	73%	4.4M	30%	1.0M	78%	4.6M	40%	1105M	65%

Table 6.4: Comparison of symmetrization methods in terms of alignment set size, resulting phrase tables size (in millions) for each size of SMT systems used in pivoting (He-En & En-Ar) and the final pivot phrase table (He-En-Ar).

and target-to-pivot $\overleftarrow{A_{pt}}$. Following Algorithm 1, we can start with union A^U_{pt} or grow-diag-final-and A^{GDFA}_{pt} alignment symmetrization. We then relax the symmetrization to allow the extraction of many new pivot phrases by removing a given word link that links a target word to a pivot word that is NOT in L_p. The final alignment matrix after all the deletions is A^F_{pt}. To remind the reader, alignment points deletion (a.k.a alignment symmetrization relaxation) allows the extraction of more phrases.

We repeat the whole process in the other language pair of the pivoting, source-pivot, to get the final alignment set A^F_{sp}. Then, these final alignment matrices are used to extract two phrase tables PT_{sp} and PT_{pt} which are used in the phrase pivoting process to produce the final pivot phrase table PT_{st}.

Table 6.4 shows the impact of different word alignment symmetrization methods on phrase tables for each system used in Hebrew-Arabic phrase-pivot SMT (He-En & En-Ar) and the final phrase table (He-En-Ar). We compare each method with and without our relaxation approach. The first row in the table is the intersection (I). The next two are union (U) without relaxation and then union with relaxation (U_R). The next two methods are heuristic grow-diagonal-final-and (GDFA) without relaxation and with relaxation (GDFA_R).

For each particular symmetrization method and each system used in pivoting, we compute the output alignment set size in first & fifth columns of table 6.4 and their percentage of the union in second & sixth columns. We also compute the size of the resulting phrase tables. The numbers show the inverse relationship between the alignment set size and the phrase table sizes. The most sparse matrix in intersection leads to huge phrase tables which consequently leads a exponentially huge final pivot phrase table with potentially a lot of low quality phrase pairs. The union has an opposite effect. It has a higher recall of alignment points including some bad alignment points that can prevent the extraction of good pivoting phrase pairs.

Figure 6.7 illustrates how the symmetrization relaxation approach can lead to good and bad English-Arabic phrase pairs.[6] The English-Arabic phrase pair (B1) is extracted into the original baseline phrase table. The word “phased" is erroneously aligned to the Arabic word وفق *wfq* ‘according to/under’ which prevents the extraction of smaller phrase pairs because of the consistency constraint (discussed in Section 6.3.3). Since the word “phased" does not appear in the English side of the Hebrew-English corpus, our relaxation method will drop all the alignment points which are connected to the word “phased". This allows the extraction of a couple of new phrase pairs (R1a & R1b). (R1a) is not a good phrase pair since it includes an extra word (“phased") in the English side that is absent in the Arabic. That said, it will not be used in the pivoting. (R1b), on the other hand, is a good phrase pair that could lead to a pivot match.

The lower half of Figure 6.7 illustrates how symmetrization relaxation does not always lead to good phrase pairs. The English-Arabic phrase pair (B2), which appears in the original baseline phrase table, is a perfectly good phrase pair. However, since the word “Saloniki" doesn’t appear in the English side of the Hebrew-English corpora, deleting it leads to the creation of two bad phrase pairs (R2a & R2b) where the English and Arabic side do not have the same meaning.

6.3.6 Model Combination

The alignment symmetrization relaxation explained in Section 6.3.5 leads to an increase in the number of phrase pairs extracted in the translation model. Some of these phrase pairs would be useful but many others are of low quality which affects the translation choices during decoding and the overall translation quality as shown in Figure 6.7.

As a solution, we construct a combined phrase table using phrase pairs from the best baseline pivoting system without relaxation and then add any additional phrase pairs extracted after relax-

[6] We use the Habash-Soudi-Buckwalter Arabic transliteration [Habash *et al.*, 2007].

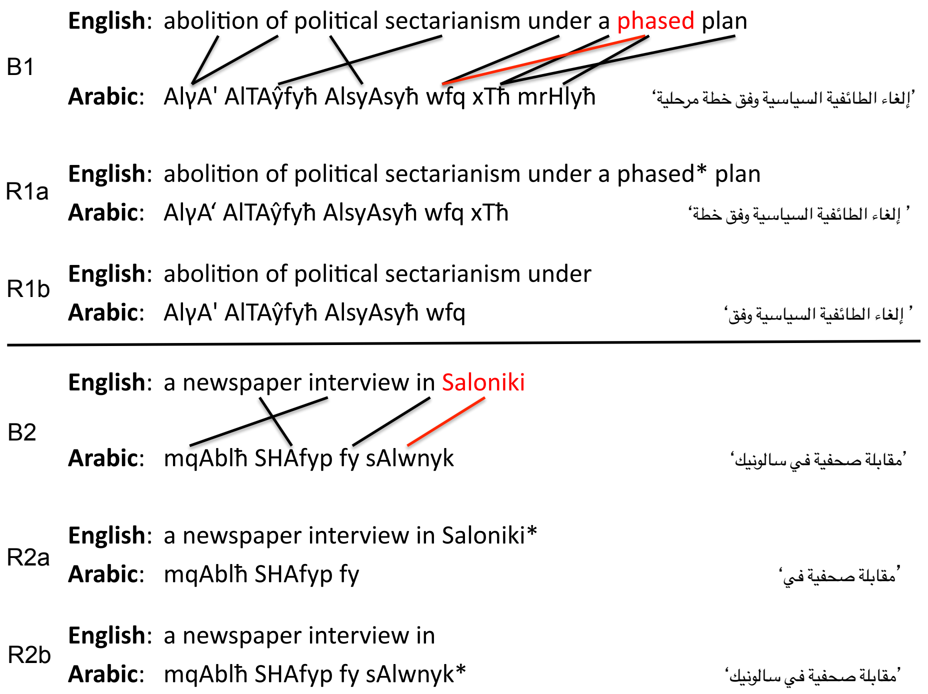

B1
English: abolition of political sectarianism under a phased plan
Arabic: AlyA' AlTAŷfyħ AlsyAsyħ wfq xTħ mrHlyħ
'إلغاء الطائفية السياسية وفق خطة مرحلية'
R1a
English: abolition of political sectarianism under a phased* plan
Arabic: AlyA' AlTAŷfyħ AlsyAsyħ wfq xTħ
'إلغاء الطائفية السياسية وفق خطة'
R1b
English: abolition of political sectarianism under
Arabic: AlyA' AlTAŷfyħ AlsyAsyħ wfq
'إلغاء الطائفية السياسية وفق'
B2
English: a newspaper interview in Saloniki
Arabic: mqAblħ SHAfyp fy sAlwnyk
'مقابلة صحفية في سالونيك'
R2a
English: a newspaper interview in Saloniki*
Arabic: mqAblħ SHAfyp fy
'مقابلة صحفية في'
R2b
English: a newspaper interview in
Arabic: mqAblħ SHAfyp fy sAlwnyk*
'مقابلة صحفية في سالونيك'

Figure 6.7: Two examples of baseline (GDFA) phrase pairs (B1 & B2) together with two pairs of phrases that are generated after symmetrization relaxation (R1a, R1b, R2a &R2b). The alignment links that are deleted as part of symmetrization relaxation are colored in red. The words marked with an asterisk do not have an equivalent in the opposite language in the phrase pair they appear in. The examples are discussed in detail in Section 6.3.5.

ation. We add a binary feature $f_{\mathbf{s},\mathbf{t}}$ to the log linear space of features in order to mark the source of the pivot phrase pairs as follows:[7]

$$f_{(\mathbf{s},\mathbf{t})} = \begin{cases} 2.718 & \text{if } (\mathbf{s},\mathbf{t}) \text{ from the baseline system} \\ 1 & \text{otherwise} \end{cases} \tag{6.3}$$

The aim from the added binary feature is to bias the translation model after tuning to favor phrase pairs from the baseline system over the complementary phrase pairs from the relaxed model.

[7] The log values of 2.718 and 1 will lead to a binary representation in the log linear space.

6.3.7 Evaluation

Next, we present a set of experiments on symmetrization relaxation for phrase-pivot SMT and on model combination.

Experimental Setup In our pivoting experiments, we build two SMT models; one model to translate from Hebrew to English and another model to translate from English to Arabic. For both models, we use the same size of parallel corpus($\approx$ 1M words) despite the fact that more English-Arabic data are available. The English-Arabic parallel corpus is a subset of available data from LDC.[8] The Hebrew-English corpus is available from sentence-aligned corpus produced by [Tsvetkov and Wintner, 2010].

Word alignment is done using GIZA++ [Och and Ney, 2003a]. For Arabic language modeling, we use 200M words from the Arabic Gigaword Corpus [Graff, 2007] together with the Arabic side of our training data. We use 5-grams for all language models (LMs) implemented using the SRILM toolkit [Stolcke, 2002].

All experiments are conducted using the Moses phrase-based SMT system [Koehn *et al.*, 2007b]. We use MERT [Och, 2003b] for decoding weight optimization. Weights are optimized using a set of 517 sentences (single reference) developed by [Shilon *et al.*, 2010].

We use a maximum phrase length of size 8 across all models. We report results on a Hebrew-Arabic evaluation set of 300 sentences with three references developed by [Shilon *et al.*, 2010]. We evaluate using BLEU-4 [Papineni *et al.*, 2002a], METEOR v1.4 [Lavie and Agarwal, 2007] and TER [Snover *et al.*, 2006].

Symmetrization Relaxation We compare the performance of symmetrization relaxation in contrast with different symmetrization methods. The results are presented in Table 6.5. In general, as expected grow-diag-final-and (GDFA) outperforms all other symmetrization methods and it is considered our baseline. Thus, the performance improves with the symmetrization relaxation for both union (U_R) and grow-diag-final-and (GDFA_R). The best performer is the relaxed grow-diag-final-and (GDFA_R). While (I) leads to comparable results to (GDFA_R), BLEU score against the

[8] LDC Catalog IDs: LDC2004T17, LDC2004E72, LDC2005E46, LDC2004T18

Symm.	**BLEU**	**METEOR**	**TER**
GDFA	20.4	33.4	62.7
GDFA_R	**20.8**	**34.0**	**62.4**
U	20.1	33.5	62.7
U_R	20.7	34.0	62.5
I	20.8	34.0	63.6

Table 6.5: Symmetrization relaxation results for different symmetrization methods for He-Ar SMT. The best performer is the relaxed grow-diag-final-and (GDFA_R). (GDFA_R) BLEU score is statistically significant over the baseline (GDFA) with p-value $= 0.12$. All other results are not statistically significant.

Symm.	**BLEU**	**METEOR**	**TER**
GDFA	20.4	33.4	62.7
GDFA_R	20.8	34.0	62.4
GDFA+GDFA_R	**21.6**	**34.4**	**61.6**

Table 6.6: Model combination experiment result. (GDFA+GDFA_R) shows a big improvement in BLEU score which is statistically significant with p-value < 0.01.

baseline (GDFA) is not statistically significant and TER is the worst across all methods.[9]

Since (GDFA_R) is the best performing model, we use (GDFA) and (GDFA_R) in our model combination experiments, next.

Model Combination We test the performance of combining the baseline (GDFA) phrase table with the relaxed (GDFA_R) phrase table as explained in Section 6.3.6.

The results in Table 6.6 show that we get a nice improvement of 1.2/1/0.8 (BLEU/METEOR/TER)

[9] Statistical significance is done using MultEval (https://github.com/jhclark/multeval) which implements statistical significance testing between systems based on multiple optimizer runs and approximate randomization [Resampling, 1989; Clark *et al.*, 2011]

points by combing the two models (GDFA) and (GDFA_R). The difference in BLEU score is statistically significant with p-value < 0.01. This result shows that our relaxation approach helps in combination with a baseline system to improve the overall translation quality. Moreover, since (GDFA_R) is a proper super-set of (GDFA) by design then the big jump in performance is due to the additional binary feature added to the log linear model. As we hoped, the binary feature biases the combined model towards the more trusted phrase pairs from (GDFA) and complement the translation model with the additional phrase pairs from symmetrization relaxation.

6.3.8 Conclusions

In this section, we discussed a symmetrization relaxation method targeting phrase-pivot SMT. The symmetrization is carried out as an optimization process to increase the matching on the pivot phrases. We show positive results (1.2 BLEU points) on Hebrew-Arabic phrase-pivot SMT. In the future, we plan to work on symmetrization based on our conclusions from this section and Chapter 7 to improve symmetrization using morpho-syntactic information.

Chapter 7

Leveraging Parallel Data in Pivoting

In the previous chapter, we discussed how to maximize both precision and recall of the pivoting process using several language independent techniques. In this chapter, we consider the case where we have parallel source-target data. In Section 7.1, we introduce different approaches to improve pivot-based SMT and discuss methods of doing system combinations. While in Section 7.2, we introduce morphology constraint scores which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. This morphology constraint scores are based on the connectivity scores. We compare two methods of generating the morphology constraint scores. One method is based on hand-crafted rules relying on our knowledge of the source and target languages. In the other method, the morphology constraints are induced from available parallel data between the source and target languages which we also use to build a direct translation model. We then combine both the pivot and direct models to achieve better coverage and overall translation quality.

7.1 Combination of Pivot and Direct SMT

In this section, we discuss a selective combination approach to effectively combine both a pivot and a direct model built from a given parallel corpora to achieve better coverage and overall translation quality. We maximize the information gain by selecting the relevant portions of the pivot model that do not interfere with the more trusted direct model.

7.1.1 Selective Combination

Our goal is to perform a smart combination between the direct and pivot models to maximize the information gain. To achieve this, we investigate the idea of classifying the pivot phrase pairs into five different classes based on the existence of source and/or target phrases in the direct model. The first class contains the phrase pairs where the source and target phrases are in the direct system together. The second class is the same as the first class except that the source and target phrases exist but not together as a phrase pair in the direct system. The third, forth and fifth classes are for the existence of source phrase only, target phrase only and neither in the direct system. Table 7.1 shows the different classifications of the portions extracted from the pivot phrase table with their labels which are used later in our results tables. The question is how to improve the quality by doing a smart selection of only relevant portion of the pivot phrase table.

Pivot phrase-pairs classification	**Src exists in direct**	**Tgt exists in direct**	**Src & Tgt exist in direct**
SRC : TGT	✓	✓	✓
SRC , TGT	✓	✓	×
SRC ONLY	✓	×	×
TGT ONLY	×	✓	×
NEITHER	×	×	×

Table 7.1: Phrase pairs classification of the portions extracted from the pivot phrase table.

7.1.2 Evaluation

In this section, we present our results for the selective combination approach between direct and pivoting models. In our pivoting experiments, we build two SMT models. One model to translate from Persian to English and another model to translate from English to Arabic. The English-Arabic parallel corpus is about 2.8M sentences (≈60M words) available from LDC[1] and GALE[2] constrained data. We use an in-house Persian-English parallel corpus of about 170K sentences and 4M words. For the direct Persian-Arabic SMT model, we use an inhouse parallel corpus of about 165k sentences and 4 million words.[3]

Word alignment is done using GIZA++ [Och and Ney, 2003b]. For Arabic language modeling, we use 200M words from the Arabic Gigaword Corpus [Graff, 2007] together with the Arabic side of our training data. We use 5-grams for all language models (LMs) implemented using the SRILM toolkit [Stolcke, 2002]. For English language modeling, we use the English Gigaword Corpus with 5-gram LM using the KenLM toolkit [Heafield, 2011].

All experiments are conducted using Moses phrase-based SMT system [Koehn *et al.*, 2007b]. We use MERT [Och, 2003b] for decoding weights optimization. For Persian-English translation model, weights are optimized using a set 1000 sentences randomly sampled from the parallel corpus while the English-Arabic translation model weights are optimized using a set of 500 sentences from the 2004 NIST MT evaluation test set (MT04).

We use a maximum phrase length of size 8 across all models. We report results on an in-house Persian-Arabic evaluation set of 536 sentences with three references. We evaluate using BLEU-4 [Papineni *et al.*, 2002a], METEOR v1.4 [Lavie and Agarwal, 2007] and TER [Snover *et al.*, 2006].

For the combination experiments, Moses allows the use of multiple translation tables [Koehn and Schroeder, 2007]. Different combination techniques are available. We use the “Either" combination technique where the translation options are collected from one table, and additional options are collected from the other tables. If the same translation option (identical source and target phrases) is found in multiple tables, separate translation options are created for each occurrence, but with different scores.

[1] LDC Catalog IDs: LDC2005E83, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006G05, LDC2007E06, LDC2007E101, LDC2007E103, LDC2007E46, LDC2007E86, LDC2008E40, LDC2008E56, LDC2008G05, LDC2009E16, LDC2009G01.

[2] Global Autonomous Language Exploitation, or GALE, is a DARPA-funded research project.

[3] Available from SAIC http://www.saic.com/

7.1.2.1 Baseline Combination

Table 7.2 shows the results of the basic combination in comparison to the best pivot translation model and the best direct model. The results shows that combining both models leads to a gain in performance. The question is how to improve the quality by doing a smart selection of only relevant portion of the pivot phrase table which is discussed next.

Model	**BLEU**	**METEOR**	**TER**
Phrase_Pivot_F1K	20.5	38.6	70.6
Direct	23.4	40.1	67.5
Direct+Phrase_Pivot_F1K	**23.7**	**40.5**	**67.2**

Table 7.2: Baseline combination experiments between best pivot baseline and best direct model for Persian-Arabic.

7.1.2.2 Selective Combination

In this section, we explore the idea of dividing the pivot phrase pairs into five different classes based on the existence of source and/or target phrases in the direct system as discussed in Section 6.3.4. We discuss our results and show the trade off between the quality of translation and the size of the different classes extracted from the pivot phrase table.

Table 7.3 shows the results of the selective combination experiments on a learning curve of 100% (4M words), 25% (1M words) and 6.25% (250K words) of the parallel Persian-Arabic corpus.

The results show that pivoting is a robust technique when there is no or limited amount of parallel corpora. In our case study on Persian-Arabic SMT, the direct translation systems built from parallel corpora starts to be better than the pivot translation system when trained on 1M words or more.

The base combination between the direct translation models and the pivot translation model leads to a boost in the translation quality across the learning curve. As expected, the smaller the parallel corpus used in training the more gain we get from the combination.

The results also show that some of pivot the classes provides more information gain than others.

Model	**Parallel data set size**		
	4M	**1M**	**250K**
Direct	23.4	21.0	16.8
Phrase_Pivot_F1K	20.5		
Base Combination	23.7 *	**22.1** *	**21.7** *
SRC : TGT	22.9	21.2	17.3 *
SRC , TGT	23.0	21.3	18.5 *
SRC ONLY	23.5	20.1	17.5 *
TGT ONLY	**23.8***	21.4 *	18.3 *
NEITHER	23.4	21.6 *	19.9 *

Table 7.3: Selective Combination experiments results on a learning curve for Persian-Arabic models. The first row shows the results of the direct system. The second row shows the result of the best pivot system. The third row shows the results of the baseline combination experiments with the whole pivot phrase table. Then the next set of rows show the results of the selective combination experiments based on the different classifications. All scores are in BLEU. (*) marks a statistically significant result against the direct baseline.

In fact some of the classes hurt the overall quality; for example, (SRC : TGT) and (SRC , TGT) both hurt the quality of translation when combined with direct model trained on 100% of the parallel data (4M words).

An interesting observation from the results is that by building a translation system with only 6.25% of the parallel data ($\approx$ 250K words) combined with the pivot translation model, we can achieve a better performance (21.7 BLEU) than a model trained on four times the amount of data (Size: 1M words; Score: 21.0 BLEU).

It is also shown across the learning curve that the best gains are achieved when the source phrase in the pivot phrase table doesn't exist in the direct model. This is expected due to the fact that by adding unknown source phrases, we decrease the overall OOVs.

Pruning the pivot phrase table is an additional benefit from the selective combination approach.

Model	**Parallel data set size**		
	4M	**1M**	**250K**
SRC : TGT	0.2%	0.1%	0.1%
SRC , TGT	35.2%	29.0%	16.0%
SRC ONLY	59.9%	63.3%	64.1%
TGT ONLY	2.3%	3.4%	6.1%
NEITHER	2.3%	4.3%	13.7%

Table 7.4: Percentage of phrase pairs extracted from the original pivot phrase table for each pivot class across the learning curve.

Table 7.4 shows that percentage of phrase pairs extracted from of the original pivot phrase table for each pivot class across the learning curve. The bulk of the phrase pairs are extracted in the classes where the source phrases exist in the direct model which add the least and sometimes hurt the overall combination performance.

For the large parallel data (4M words), selective combination with (TGT ONLY) class gives a slightly better result in BLEU while hugely reducing the size of the pivot phrase table used (2.3% of the original pivot phrase table). For smaller parallel data, the advantage is reduced but here comes the trade off between the quality of the translation and the size of the model.

7.1.3 Conclusions

We discussed a selective combination approach between pivot and direct models to improve the translation quality. We showed that the selective combination can lead to a large reduction of the pivot model without affecting the performance if not improving it. The results show that some of pivot the classes provides more information gain than others. In fact some of the classes hurt the overall quality; for example, (SRC : TGT) and (SRC , TGT) both hurt the quality of translation when combined with direct model trained on 100% of the parallel data (4M words).

We also showed that across the learning curve that the best gains are achieved when the source phrase in the pivot phrase table doesn't exist in the direct model. This is expected due to the fact that by adding unknown source phrases, we decrease the overall OOVs. However, these results only hold for medium size pivot models. When we increase the data (60M words) used for building English-Arabic model used in phrase pivoting. These benefits disappear and the basic combination lead to the best performance. We also tried on a different language pair (Hebrew-Arabic) using the big English-Arabic model and we reached the same conclusion.

7.2 Synchronous Morpho-syntactic Constraints

In this section, we leverage the idea of extracting useful information between any language pair to help in the pivoting process. We introduce morphology constraints which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. We compare two methods of generating these constraints. One method is based on hand-crafted rules relying on our knowledge of the source and target languages; while in the other method, the morphology constraints are induced from available parallel data between the source and target languages which we also use to build a direct translation model. We then combine both the pivot and direct models to achieve better coverage and overall translation quality. We show positive results on Hebrew-Arabic SMT. We get 1.5 BLEU points over phrase pivot baseline and 0.8 BLEU points over system combination baseline with direct model built from given parallel data.

We showed before how ambiguity and richness of source and target languages increase the number of combinations between source and target phrases. A basic solution to the combinatorial expansion is to filter the phrase pairs used in pivoting based on log-linear scores as discussed in Section 6.1.2. However, this doesn't solve the low quality problem.

Similar to factored translation models [Koehn and Hoang, 2007] where linguistic (morphology) features are augmented to the translation model to improve the translation quality, our approach to address the quality problem is based on constructing a list of synchronous morphology constraints between the source and target languages. These constraints are used to generate scores to determine the quality of pivot phrase pairs. However, unlike factored models, we do not use the morphology in generation and the morphology information comes completely from external resources. In addition, since we work in the pivoting space, we only apply the morphology constraints to the connected words between the source and target languages through the pivot language. This guarantees a fundamental level of semantic equivalence before applying the morphology constraints especially if there is distortion between source and target phrases.

We build on our approach in Chapter 6.2 where we introduced connectivity strength features between the source and target phrase pairs in the pivot phrase table. We again utilize the alignment links that are generated by projecting the alignments of the source-pivot phrase pairs and the pivot-target phrase pairs used in pivoting. This time instead of using the lexical mapping between source and target words, we compute quality scores based on the morphological compatibility between the connected source and target words.

To choose which morphological features to work with, we performed an automatic error analysis on the output of the phrase-pivot baseline system. We did the analysis using AMEANA [El Kholy and Habash, 2011], an open-source error analysis tool for natural language processing tasks targeting morphologically rich languages (explained in details in Chapter 4). Again, we found that the most problematic morphological features in the Arabic output are gender (GEN), number (NUM) and determiner (DET). We focus on those features in our experiments. Next, we present our approach to generating the morphology constraint features using hand-crafted rules and compare this approach with inducing these constraints from Hebrew-Arabic parallel data. We didn't show results for Persian-Arabic due to the lack of a morphological analyzer for Persian which is required to build the morphology constraints.

7.2.1 Rule-based Morphology Constraints

Our rule-based morphology constraint features are basically a list of hand-crafted mappings of the different morphological features between Hebrew and Arabic. Since both languages are morphologically rich, it is straightforward to produce these mappings for GEN, NUM and DET. Note, however, that we also account for ambiguous cases; e.g., feminine gender in Arabic can map to words with ambiguous gender in Hebrew. We additionally use different POS tag sets for Arabic (47 tags) and Hebrew (25 tags) and in many cases one Hebrew tag can map to more than one Arabic tag; for example, three Arabic noun tags *abbrev, noun* and *noun_prop* map to two Hebrew tags *feminine, masculine* noun.[4] Table 7.5 shows a sample of the morphological mappings between Arabic and Hebrew.

[4] Please refer to [Habash *et al.*, 2009] for a complete set of Arabic POS tag set and [Adler, 2007] for Hebrew POS tag set.

Features	**Arabic**	**Hebrew**
GEN	Feminine	Feminine / Both
	Masculine	Masculine / Both
NUM	Singluar	Singluar / Singluar-Plural
	Dual	Dual / Dual-Plural
	Plural	Plural / Dual-Plural / Singular-Plural
DET	No Determiner	No Determiner
	Determiner	Determiner

Table 7.5: Rule-based mapping between Arabic and Hebrew morphological features. Each feature value in Arabic can map to more than one feature value in Hebrew.

After building the morphological features mappings, we use them to judge the quality of a given phrase pair in the phrase pivot model. We add two scores W_s and W_t to the log linear space. Given a *source-target* phrase pair $\bar{s}, \bar{t}$ and a word projected alignment a between the source word positions $i = 1, ..., n$ and the target word positions $j = 1, ..., m$, W_s and W_t are defined in equations 7.1 and 7.2. F is the set of morphological features (we focus on GEN, NUM, DET and POS). M_f is the hand-crafted rules mapping between Arabic and Hebrew feature values of feature $f \in F$. In case of ambiguity for a given feature; for example, a word's gender being masculine or feminine, we use the maximum likelihood value of this feature given the word. $MLE_f(i)$ is the maximum likelihood feature value of feature f for the source word at position i, and $MLE_f(j)$ is the maximum likelihood feature value of feature f for the target word at position j. The maximum likelihood feature values for Hebrew were computed from the Hebrew side of the training data. As for Arabic, the maximum likelihood feature values were computed from the Arabic side of the training data in addition to Arabic Gigaword corpus, which was used in creating the language model (more details in Section 7.2.4.1).

$$W_s = \frac{1}{|F|} \sum_{\forall f \in F} \sum_{\forall (i,j) \in a} \frac{1}{n} [(MLE_f(i), MLE_f(j)) \in M_f] \qquad (7.1)$$

$$W_t = \frac{1}{|F|} \sum_{\forall f \in F} \sum_{\forall (i,j) \in a} \frac{1}{m} [(MLE_f(i), MLE_f(j)) \in M_f] \tag{7.2}$$

7.2.2 Induced Morphology Constraints

In this section, we explain our approach in generating morphology constraint features from a given parallel data between source and target languages. Unlike the rule-based approach we build a translation model between the source and target morphological features and we use the morphology translation probabilities as metric to judge a given phrase pair in the pivot phrase table. For the automatically induced constraints, we jointly model mapping between conjunctions of features attached to aligned words rather than tallying each feature match independently. Writing good manual rules for such feature conjunction mappings would be more difficult. Table 7.6 shows some examples of mapping (GEN), number (NUM) and determiner (DET) in Hebrew to their equivalent in Arabic and their respective bi-directional scores.

Hebrew (H)	**Arabic (A)**	$P_{FC}(A\|H)$	$P_{FC}(H\|A)$
[Fem+Dual+Det]	[Fem+Dual]	0.0006	0.0833
[Fem+Dual+Det]	[Fem+Dual+Det]	0.0148	0.3333
[Fem+Dual+Det]	[Fem+Singular] [Fem+Dual]	0.0052	0.0833
[Fem+Dual+Det]	[Masc+Dual+Det]	0.0047	0.5000

Table 7.6: Examples of induced morphology constraints for (GEN), number (NUM) and determiner (DET) and their respective scores.

As in rule-based approach, we add two scores W_s and W_t to the log linear space which are defined in equations 7.3 and 7.4. P_{FC} is the conditional morphology probability of a given feature combination (FC) value. Similar to rule-based morphology constraints, we resort to the maximum likelihood value of a feature combination when the values are ambiguous. $MLE_{FC}(i)$ is the maximum likelihood feature combination (FC) value for the source word at position i while $MLE_{FC}(j)$ is the maximum likelihood feature combination (FC) value for the target word at position j.

$$W_s = \frac{1}{n} \sum_{\forall (i,j) \in a} P_{FC}(MLE_{FC}(i)|MLE_{FC}(j)) \tag{7.3}$$

$$W_t = \frac{1}{m} \sum_{\forall (i,j) \in a} P_{FC}(MLE_{FC}(j)|MLE_{FC}(i)) \tag{7.4}$$

7.2.3 Model Combinations

Since we use parallel data to induce the morphology constraints, it would make sense to measure the effect of combining (a) the pivot model with added morphology constraints, and (b) the direct model trained on the parallel data used to induce the morphology constraints. We perform the combination using Moses' phrase table combination techniques. Translation options are collected from one table, and additional options are collected from the other tables. If the same translation option (in terms of identical input phrase and output phrase) is found in multiple tables, separate translation options are created for each occurrence, but with different scores [Koehn and Schroeder, 2007]. We show results over a learning curve in Section 7.2.4.5.

In this section, we present a set of experiments comparing the use of rule-based versus induced morphology constraint features in phrase-pivot SMT as well as model combination to improve Hebrew-Arabic pivot translation quality.

7.2.4 Evaluation

7.2.4.1 Experimental Setup

In our pivoting experiments, we build two SMT models; one model to translate from Hebrew to English, and another model to translate from English to Arabic. The English-Arabic parallel corpus is about ($\approx$ 60M words) and is available from LDC[5] and GALE[6] constrained data. The Hebrew-English corpus is about ($\approx$ 1M words) and is available from sentence-aligned corpus produced by

[5] LDC Catalog IDs: LDC2005E83, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006G05, LDC2007E06, LDC2007E101, LDC2007E103, LDC2007E46, LDC2007E86, LDC2008E40, LDC2008E56, LDC2008G05, LDC2009E16, LDC2009G01.

[6] Global Autonomous Language Exploitation, or GALE, was a DARPA-funded research project.

[Tsvetkov and Wintner, 2010]. For the direct Hebrew-Arabic SMT model, we use a TED parallel corpus of about ($\approx$ 2M words) [Cettolo *et al.*, 2012].

Word alignment is done using GIZA++ [Och and Ney, 2003b]. For Arabic language modeling, we use 200M words from the Arabic Gigaword Corpus [Graff, 2007] together with the Arabic side of our training data. We use 5-grams for all language models (LMs) implemented using the SRILM toolkit [Stolcke, 2002].

All experiments are conducted using the Moses phrase-based SMT system [Koehn *et al.*, 2007b]. We use MERT [Och, 2003b] for decoding weight optimization. Weights are optimized using a tuning set of 517 sentences developed by [Shilon *et al.*, 2010].

We use a maximum phrase length of size 8 across all models. We report results on a Hebrew-Arabic development set (Dev) of 500 sentence with a single reference and an evaluation set (Test) of 300 sentences with three references developed by [Shilon *et al.*, 2010]. We evaluate using BLEU-4 [Papineni *et al.*, 2002a], METEOR v1.4 [Lavie and Agarwal, 2007] and TER [Snover *et al.*, 2006].

7.2.4.2 Baseline

We compare the performance of adding the connectivity strength features (+*Conn*) to the phrase pivoting SMT model (*Phrase_Pivot*) and building a direct SMT model using all parallel He-Ar corpus available. The results are presented in Table 7.7. Consistently with our previous results in Chapter 6.2, the performance of the phrase-pivot model improves with the connectivity strength features. While the direct system is worse than the phrase pivot model in general, the combination of both models leads to a high performance gain of 1.5/5.3 BLEU points in Dev/Test over the best performers of both the direct and phrase-pivot models.

7.2.4.3 Rule-based Morphology Constraints

In this experiment, we show the performance of adding hand-crafted morphology constraints (+*Morph_Rules*) to determine the quality of a given phrase pair in the phrase-pivot translation model. The forth row in Table 7.8 shows that although the rules are based on a one-to-one mapping between the different morphological features, the translation quality is improved over the baseline phrase-pivot model by 0.2/0.5 BLEU points in Dev/Test sets.

As expected, the system combination of the pivot model with the direct model improves the

Model	**Dev**			**Test**		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Direct	9.7	23.1	79.3	20.3	33.9	63.5
Phrase_Pivot	9.9	23.6	79.5	20.8	33.4	64.2
Phrase_Pivot+Conn	10.2	23.7	79.0	21.6	34.2	62.3
Direct+Phrase_Pivot+Conn	**11.7**	**27.0**	**76.0**	**26.9**	**39.4**	**59.7**

Table 7.7: Comparing phrase pivoting SMT with connectivity strength features, direct SMT and the model combination. The results show that the best performer is the model combination in Dev and Test sets.

overall performance. By adding the hand-crafted morphology constraints, we get a nice gain of 0.7/0.1 BLEU points in Dev/Test sets.

7.2.4.4 Induced Morphology Constraints

In this experiment, we measure the effect of using induced morphology constraints (+*Morph_Auto*) on MT quality. The fifth row in Table 7.8 shows that the induced morphology constraints improve the results over the baseline phrase-pivot model by 0.2/1.7 BLEU points in Dev/Test sets and over the Rule-based morphology constraints by 1.2 BLEU points in the Test set.

The system combination of the pivot model with the direct model improves the overall performance. The model using induced morphological features is the best performer with an increase in the performance gain by 1.5/0.6 BLEU points in Dev/Test sets. This shows that the benefit we get from the induced morphology constraints were not diluted when we do the model combination given the fact that the constraints were induced from the parallel data to start with.

It is important to note here that the induced morphology constraints outperformed the rule-based constraints across all settings. This shows that the complex morphology constraints extracted from the parallel data provide knowledge that can not be covered by simple linguistic rules. However, the simple rule-based approach comes in handy when there is no data between the source and target languages.

Model	**Dev**			**Test**		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Direct	9.7	23.1	79.3	20.3	33.9	63.5
Phrase_Pivot	9.9	23.6	79.5	20.8	33.4	64.2
Phrase_Pivot+Conn	10.2	23.7	79.0	21.6	34.2	62.3
Phrase_Pivot+Conn+Morph_Rules	10.4	23.7	78.7	22.1	34.7	62.1
Phrase_Pivot+Conn+Morph_Auto	**10.4**	**23.8**	**78.6**	**23.3**	**35.2**	**61.9**
Direct+Phrase_Pivot+Conn	11.7	27.0	76.0	26.9	39.4	59.7
Direct+Phrase_Pivot+Conn+Morph_Rules	12.4	27.4	75.0	27.0	39.6	59.6
Direct+Phrase_Pivot+Conn+Morph_Auto	**13.2***	**28.7***	**73.6***	**27.5***	**39.9***	**58.1***

Table 7.8: Morphology constraints results. The first row is the direct model. From second to fourth rows are the pivot SMT models with additional morphological constraints features. The last three rows are the results of system combination between the direct model and the different phrase pivoting models. (*) marks a statistically significant result against both the direct and phrase-pivot baseline.

7.2.4.5 Learning Curve

In this experiment, we examine the effect of using less data in inducing morphology constraints rules and the overall performance when we combine systems. Table 7.9 shows the results on a learning curve of 100% (2M words), 25% (500K words) and 6.25% (125K words) of the parallel Hebrew-Arabic corpus.

As expected, The system combination between the direct translation models and the phrase-pivot translation model leads to an improvement in the translation quality across the learning curve even when there is small amount of parallel corpora. Despite the weak performance (2.7 BLEU) of the direct system built on 6.25% of the parallel Hebrew-Arabic corpus, the system combination leads to 1.4 BLEU points gain.

An interesting observation from the results is that we always get a performance gain from the induced morphology constrains across all settings. This shows that the system combination helps in

Parallel Data Size	Model	Dev		Test	
		Single	Combined	Single	Combined
125K	Direct	2.7	n/a	8.4	n/a
	Phrase_Pivot+Conn	9.1	10.4	20.1	20.9
	Phrase_Pivot+Conn+Morph_Auto	9.2	10.6	20.6	21.3
500K	Direct	5.9	n/a	15.1	n/a
	Phrase_Pivot+Conn	9.1	10.7	20.1	22.5
	Phrase_Pivot+Conn+Morph_Auto	9.7	11.2	20.8	22.8
2M	Direct	9.7	n/a	20.3	n/a
	Phrase_Pivot+Conn	10.2	11.7	21.6	26.9
	Phrase_Pivot+Conn+Morph_Auto	**10.4**	**13.2**	**23.3**	**27.5**

Table 7.9: Learning curve results of 100% (2M words), 25% (500K words) and 6.25% (125K words) of the parallel Hebrew-Arabic corpus.

adding more lexical translation choices while the constraints help in a different dimension, which is selecting the best phrase pairs from the pivot system.

7.2.5 Case Study

In this section we consider an example from our Dev set that captures many of the patterns and themes in the evaluation. Table 7.10 shows a Hebrew source sentence and its Arabic reference. This is followed by the output from the pivot system, the direct system, the Phrase_Pivot+Conn+Morph_Auto system and the combined system.

Two particular aspects should be noted. First is the complementary lexical coverage of the direct and pivot systems. This is seen in how one of each covers half of the phrase *middlemen and traders*. The combined system captures both. Second, the gender, number and tense of the main verb prove challenging in many ways (and this is an issue for a majority of the sentences in the Dev set). The Hebrew verb in the present tense is masculine and plural; and naturally follows the subject. The Arabic reference verb appears at the beginning of the sentence, in which location it only agrees with the subject in gender (while number is singular). Arabic Verbs in

Hebrew Source	המתווכים והסוחרים מסרבים לדבר בפומבי על המחירים. *the+middlemen and+the+traders refuse[m.p.] to+speak publicly about the+prices*
Arabic Reference	يرفض الوسطاء والتجار الحديث علنا عن الاسعار *refuse[m.s.] the+middlemen and+the+traders the+speaking publicly about the+prices*
Phrase_Pivot+Conn	وسطاء **והסוחרים** يرفض التحدث علنا عن الاسعار *middlemen* **והסוחרים** *refuse[m.s.] the+speaking publicly about the+prices*
Direct	**המתווכים** والتجار رفضوا الحديث على الملأ على الاسعار **המתווכים** *and+the+traders refused[m.p.] the+speaking upon the+public about the+prices*
Phrase_Pivot+Conn+ Morph_Auto	الوسطاء **והסוחרים** يرفضون التحدث علنا عن الاسعار *the+middlemen* **והסוחרים** *refuse[m.p.] the+speaking publicly about the+prices*
Direct+Phrase_Pivot+ Conn+Morph_Auto	وسطاء والتجار رفضوا التحدث علنا عن الاسعار *middlemen and+the+traders refused[m.p.] the+speaking publicly about the+prices*

Table 7.10: Translation examples.

SVO order agree in gender and number. All the MT systems we compare leave the verb after the subject. The direct, Phrase_Pivot+Conn+Morph_Auto, and combination systems get the number and gender correctly; however, the direct and combined system make the verb tense past. The Phrase_Pivot+Conn+Morph_Auto example highlights the value of morphology constraints; but the example points out that they sometimes are hard to evaluate automatically, since there are morphosyntactically allowable forms that do not match the translation references.

7.2.6 Phrase Pivoting Best Setup

To have a complete picture of the best setup for phrase pivoting, we combined the approaches discussed in Chapter 6 and Chapter 7 to get the maximum gain in translation quality. Approaches in both chapters complement each other. In Chapter 6, all approaches are language independent while in Chapter 7, we use linguistic information to improve the quality of the model. We use the best alignment symmetrization technique in Section 6.3 “GDFA+GDFA_R" in our experiment. Table 7.11 shows the results of combining all the approaches on Hebrew-Arabic phrase-pivot translation model. The first row is the direct system output built from parallel data between Hebrew and Arabic. The second row is the phrase pivoting model output with connectivity scores added to the model. The last two rows are the results of combining direct model with the phrase-pivot model and showing the effect of the induced morphology constraints in the third row and the finally using the best alignment symmetrization technique in the last row. Unfortunately, we didn't get a lot of improvement using alignment symmetrization relaxation and that could be because of the saturation of the phrase pivot model given the Dev and Test sets.

Model	**Dev**			**Test**		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Direct	9.7	23.1	79.3	20.3	33.9	63.5
Phrase_Pivot+Conn	10.2	23.7	79.0	21.6	34.2	62.3
Direct+Phrase_Pivot+Conn+Morph_Auto	13.2	28.7	73.6	27.5	39.9	58.1
Direct+Phrase_Pivot+Conn+Morph_Auto +Align_Symm	13.2	**28.8**	73.4	**27.7**	**40.1**	58.1 *

Table 7.11: Phrase Pivoting Best Setup on Hebrew-Arabic phrase-pivot translation model.

7.2.7 Conclusions

We presented the use of synchronous morphology constraint features based on hand-crafted rules compared to rules induced from parallel data to improve the quality of phrase-pivot based SMT. We show that the two approaches lead to an improvement in the translation quality. The induced morphology constraints approach is a better performer, however, it relies on the fact there is a parallel corpus between source and target languages. We show positive results on Hebrew-Arabic SMT. We get 1.5 BLEU points over phrase-pivot baseline and 0.8 BLEU points over system combination baseline with direct model built from given parallel data.

Chapter 8

Conclusions

In our research, we worked on improving Pivot-based Statistical Machine Translation for Morphologically Rich Languages with limited resources. We developed a pivoting framework which is based on constructing two separated SMT systems, Source-Pivot SMT and Pivot-Target; and then perform phrase-pivoting. We developed several methods to improve each component separately, and also developed methods to improve the system as the whole targeting of the final pivot SMT system. Following is a summary of contributions followed by an overall discussion of the thesis.

8.1 Summary of Contributions

The first challenge we targeted in our approaches is data sparsity. The following is a list of our contributions to solve this challenge.

- **Morphological Processing:** We explored a space of tokenization schemes and normalization options. We also examined a set of six detokenization techniques to evaluate the detokenized and orthographically corrected (enriched) output. Our best setup lead to a significant increase of BLEU score by 1.3 points for medium models and 1 point for large models.

- **Separation between Translation and Morphology Generation:** We developed three methods of modeling morphological Features that can be modeled as part of the core translation process generated or predicted. Our results suggested that depending on the language, some morphological features better be part of the core translation process and then predict the most problematic feature and then finally do a generation step. Our results outperform the state of the art factored models and we showed an improvement of 0.9 BLEU point compared to the baseline.

- **Automatic Error Analysis for Morphologically Rich Languages:** We developed AMEANA an open-source tool for error analysis of natural language processing tasks targeting MRLs. AMEANA produces detailed statistics on morphological errors in the output. It also generates an oracularly modified version of the output that can be used to measure the effect of these errors using any evaluation metric. AMEANA is a language independent tool except that a morphological analyzer must be provided for a given language.

The second challenge we were concerned with is the pivoting process itself. We try to maximize both precision and recall of the pivoting process through the following approaches.

- **Pivoting Recall Maximization:** We implemented a language independent technique to improve the recall of the pivot matching by improving the alignment symmetrization method. Symmetrization is carried out as an optimization process driven by the effectiveness of each alignment pair with respect to pivoting, and add or remove the word links that can maximize the pivoting process.

- **Pivoting Quality Maximization:** We presented two language independent features, source connectivity score and target connectivity score, to improve the quality of pivot-based SMT. We showed that these features help improving the overall translation quality and we got an nice improvements over the baselines for different language pairs.

The third challenge we are concerned with is how to make use of any parallel data between the source and target languages. We implemented different approaches to improve the pivot SMT system and methods of combination between the pivot system and the direct system built from the parallel data.

- **Combination of Pivot and Direct Models:** We developed a smart technique to combine pivot and direct models. We maximize the information gain by selecting the relevant portions of the pivot model that do not interfere with the direct model which is in principal trusted more. We showed that the selective combination can lead to a large reduction of the pivot model without affecting the performance if not improving it.

- **Morpho-syntactic Constraints** We applied morpho-syntactic constraints between the source and target languages to improve the translation quality. We compared two methods of generating the morpho-syntactic constraints. One method is based on hand-crafted rules relying on our knowledge of the source and target languages; while in the other method, the morphology constraints are induced from available parallel data between the source and target languages. Using induced morphology constraints outperformed the handcrafted rules and improved over our best model from all previous approaches by 0.6 BLEU points (7.2/6.7 BLEU points from the direct and pivot baselines respectively).

8.2 Discussion

In this thesis, we provided a pivoting framework to translate to and from morphologically rich languages (MRL) especially in the context of having limited or no parallel corpora between the source and the target languages. We addressed three main challenges. The first challenge is the sparsity of data as a result of morphological richness. The second one is maximizing precision and recall of the pivoting process itself. The last one is making use of any seed data between the source and the target languages.

In general, our discussed solutions can be applied to any MRL since most of our approaches are language independent. The few techniques requiring linguistic knowledge or the availability of morphological analyzers can be easily adapted to any language. In most of our work, we target Arabic as it is one of the most challenging languages in the field, but we work with other languages; specifically, Persian and Hebrew.

To address the first challenge of data sparsity, we presented experiments studying a large number of variables for English-Arabic SMT systems that produce correctly tokenized and enriched Arabic text. The results show that lemma based alignment leads to a better output quality. Our best system uses the Penn Arabic Treebank (PATB) tokenization scheme and reduced Arabic word forms followed by a language-model based joint detokenization and enrichment step.

In another direction, we address these challenges through different modeling methods. In our approach, morphological features can be modeled as part of the core translation process mapping source tokens to target tokens. Alternatively, these features can be generated using target monolingual context as part of a separate generation (or post-translation inflection) step. Finally, the features can be predicted using both source and target information in a separate step before generation. Our results show that the generation of fully inflected forms from uninflected lemmas in a purely monolingual setting such as our morphological generation step is very hard – we get only 82.2% BLEU starting with gold lemmas. Adding different combinations of gold values of the three most problematic morphological features improves the score by over 12% absolute BLEU to a higher performance ceiling (94.8% BLEU).

Automatically modeling these features at a high accuracy for SMT, however, turns out to be rather hard. If we consider using them as part of the translation step together with lemmas, we find that they almost always hurt the end-to-end (translation-generation) MT system except for the DET feature which improves over an inflected tokenized baseline by about 0.6% BLEU.

Predicting the feature values using an independent supervised learning step that has access to the English word, POS and syntax features produces accuracy scores ranging in mid to high 80s%. Comparing the prediction accuracy of GEN, NUM and DET, we find NUM is the easiest to predict, followed by DET and then GEN. This makes sense given the information provided from English, which is inflected for NUM, but not GEN.

The results also show that DET, as a single feature, helps more when it is part of the translation step (30.1 BLEU) compared to being predicted (29.7∼29.8). In both cases, it fares better than simply leaving determining DET to the generation step (29.5).

Neither GEN nor NUM, as single features, help much (or at all) over the baselines when part of the translation step or when predicted. However, when both are combined with DET they consistently help only when GEN and NUM are predicted, not translated. It is possible that the lower performance we see as part of the translation is a product of how we translate: we do not factor these features in the translation – a direction we plan to consider in the future. We postulate that the prediction step helps because it has access to more information than used in our translation step, e.g., source language syntax.

In order to help decide which features to translate, to generate or to predict, we present AMEANA (Automatic Morphological Error Analysis), an automatic error analysis tool that is designed to identify morphological errors in the output of a given system against a gold reference. AMEANA produces detailed statistics on morphological errors in the output. It also generates an oracularly modified version of the output that can be used to measure the effect of these errors using any evaluation metric. AMEANA is a language independent tool except that a morphological analyzer must be provided for a given language.

The second problem that we are concerned with in this thesis is the pivoting process itself. In the standard phrase-pivoting approach, many phrase pairs between source and target languages are not generated because of bad matching of pivot phrases. On the other hand, the size of the newly created pivot phrase table is very large. Moreover, many of the produced phrase pairs are of low quality which affects the translation choices during decoding and the overall translation quality.

We try to maximize both precision and recall of the pivoting process, and we discuss several techniques to improve the recall of the pivot matching. One of the techniques works on the level of the word alignment symmetrization, like the common heuristics for symmetrization. We aim to find a balance between the intersection and union. But unlike the state of the art heuristics, symmetrization is carried out as an optimization process driven by the effectiveness of each alignment pair with respect to pivoting, and add or remove the word links that can maximize the pivoting process. We showed big jump in BLEU scores on Hebrew-Arabic and Persian-Arabic phrase-pivot SMT. In one of our best models we reach an improvement of 1.2 BLEU points.

Despite the fact that we miss a lot of matches in pivoting and that we try to improve the recall, we also need to consider the quality precision of phrase pivoting. One of the manifestations of phrase pivoting is that the size of the newly created pivot phrase table is very large. Moreover, many of the produced phrase pairs are of low quality which affects the translation choices during the decoding and the overall translation quality. We discuss different techniques to determine the quality of the pivot phrase pairs between the source and the target. In one of the language independent approaches, we generate different connectivity scores between the source and target phrase pairs based on the alignment information propagated from the source-pivot and pivot-target systems. These features were shown to be very effective and consistently improved the translation quality across all systems. The results show that we get a nice improvement of ≈0.6/0.5 BLEU points for both models (Pesrian-Arabic and Hebrew-Arabic).

The third challenge we are concerned with is how to make use of any parallel data between the source and the target languages. We discuss different approaches to improve the pivot SMT system and methods of the combination between the pivot system and the direct system built from the parallel data.

In one of the approaches, we introduce morphology constraint scores which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. This morphology constraint scores are based on the connectivity scores. We compare two methods of generating the morphology constraints. One method is based on hand-crafted rules relying on our knowledge of the source and target languages; while in the other method, the morphology constraints are induced from available parallel data between the source and target languages which we also use to build a direct translation model. We then combine both the pivot and direct models to achieve better coverage and overall translation quality. We show that the two approaches lead to an improvement in the translation quality. The induced morphology constraints approach is a better performer, however, it relies on the fact there is a parallel corpus between source and target languages. We show positive results on Hebrew-Arabic SMT. We get 1.5 BLEU points over phrase-pivot baseline and 0.8 BLEU points over system combination baseline with direct model built from given parallel data.

We also discussed applying smart techniques to combine pivot and direct models. We aim at having a better coverage and overall translation quality. The combination approach needs to be optimized in order to maximize the information gain. We maximize the information gain by selecting the relevant portions of the pivot model that do not interfere with the direct model which is in principal trusted more. The results show that the selective combination can lead to a large reduction of the pivot model without affecting the performance if not improving it.

8.3 Future Work

There are two directions where we see a space of improvements. One direction is targeting the main building block which is SMT for morphologically rich languages and the other direction is the pivoting process itself.

Regarding SMT for morphologically rich languages, we would like to work more of the idea of separation between translation and generation. We want to investigate the use of system combination techniques and language modeling approaches that target complex morphology such as factored

LMs [Bilmes and Kirchhoff, 2003]. We also would like to work on improving the morphological feature choices for generation. Moreover, we want to explore the idea of retargeting our framework to post-editing any generic MT system.

Along the same lines of post-editing, we would like to work more on AMEANA. We plan to convert the tool from just an error analysis to tool into a machine translation metric when the target language is a morphologically rich language. This may require a lot of tuning for each language separately but it could a better sense of fluency than the current harsh metrics.

The other direction that we see a big potential for improvement is our work on pivoting. We would like to explore other features to determine the quality of the produced phrase pairs between source and target languages, e.g., the number of the pivot phrases used in connecting the source and target phrase pair and the similarity between these pivot phrases. We also plan to work on reranking experiments as a post-translation step based on morphosyntactic information between source and target languages. Another idea with good potential is to work on word reordering between morphologically rich languages to maintain the relationship between the word order and the morphosyntactic agreement in the context of phrase pivoting.

In another direction to prune the pivot phrase table, we discuss training a binary classifier on any available parallel corpus between source and target languages to prune pivot phrase pairs in a way that is directly related to the translation quality, and can take advantage of several feature functions that account for different aspects of phrase pair quality.

Taking the problem one level deeper by targeting the alignment models, we aim of using the alignment framework by [Ganchev *et al.*, 2008; Graça *et al.*, 2010] to incorporate agreement constraints to EM training using Posterior Regularization (PR) that aims to incorporate linguistic information extracted from the seed data into unsupervised estimation in the form of constraints on the model's posteriors.

Part I

Bibliography

Bibliography

[A. Cuneyd Tantug and El-Kahlout, 2008] Kemal Oflazer A. Cuneyd Tantug and Ilknur Durgar El-Kahlout. Bleu+: a tool for fine-grained bleu computation. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[Adler, 2007] Menahem Meni Adler. *Hebrew morphological disambiguation: An unsupervised stochastic word-based approach*. PhD thesis, Ben-Gurion University of the Negev, 2007.

[Al-Haj and Lavie, 2010] Hassan Al-Haj and Alon Lavie. The Impact of Arabic Morphological Segmentation on Broad-coverage English-to-Arabic Statistical Machine Translation. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA-2010)*, Denver, Colorado, November 2010.

[Alkuhlani and Habash, 2011] Sarah Alkuhlani and Nizar Habash. A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, Oregon, USA, 2011.

[Badr *et al.*, 2008] Ibrahim Badr, Rabih Zbib, and James Glass. Segmentation for English-to-Arabic Statistical Machine Translation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 153–156, Columbus, Ohio, June 2008.

[Bar-Hillel, 1964] Y. Bar-Hillel. A demonstration of the nonfeasibility of fully automatic high quality machine translation. In *Language and Information: Selected essays on their theory and application*, pages 174–179, Jerusalem, Israel, 1964. The Jerusalem Academic Press Ltd.

[Berger *et al.*, 1996] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22:39–71, 3 1996.

[Bertoldi *et al.*, 2008] Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. Phrase-based statistical machine translation with pivot languages. *Proceeding of IWSLT*, pages 143–149, 2008.

[Bilmes and Kirchhoff, 2003] Jeff A. Bilmes and Katrin Kirchhoff. Factored language models and generalized parallel backoff. In *Proceedings of the Human Language Technology Conference/North American Chapter of Association for Computational Linguistics (HLT/NAACL-03)*, pages 4–6, Edmonton, Canada, 2003.

[Blanchon and Boitet, 2007] Hervé Blanchon and Christian Boitet. Pour l'évaluation des systèmes de TA par des méthodes externes fondées sur la tâche. *TAL*, 48(1):33–65, 2007.

[Brown *et al.*, 1993a] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[Brown *et al.*, 1993b] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311, 1993.

[Buckwalter, 2004] Tim Buckwalter. Buckwalter Arabic Morphological Analyzer Version 2.0, 2004. Linguistic Data Consortium, University of Pennsylvania. LDC Catalog No.: LDC2004L02, ISBN 1-58563-324-0.

[Callison-Burch *et al.*, 2006] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, pages 249–256, Trento, Italy, 2006.

[Callison-Burch *et al.*, 2008] Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further meta-evaluation of machine translation. In *Proceedings of* *the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 70–106, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[Callison-Burch *et al.*, 2009] Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. Findings of the 2009 workshop on statistical machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 1–28, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Carpuat and Wu, 2007] Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic, June 2007.

[Cettolo *et al.*, 2012] Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May 2012.

[Chiang, 2005] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, 6 2005. Association for Computational Linguistics.

[Clark *et al.*, 2011] Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181. Association for Computational Linguistics, 2011.

[Clifton and Sarkar, 2011] Ann Clifton and Anoop Sarkar. Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 32–42, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[Crego and Yvon, 2009] Josep Maria Crego and François Yvon. Gappy translation units under left-to-right smt decoding. In *Proceedings of the meeting of the European Association for Machine Translation (EAMT)*, pages 66–73, Barcelona, Spain, 2009.

[Denkowski and Lavie, 2010] Michael Denkowski and Alon Lavie. METEOR-NEXT and the METEOR Paraphrase Tables: Improved Evaluation Support For Five Target Languages. In *Proceedings of the ACL 2010 Joint Workshop on Statistical Machine Translation and Metrics MATR*, 2010.

[Diehl *et al.*, 2009] F. Diehl, M.J.F. Gales, M. Tomalin, and P.C. Woodland. Morphological Analysis and Decomposition for Arabic Speech-to-Text Systems. In *Proceedings of InterSpeech*, 2009.

[Doddington, 2002] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Human Language Technology*, pages 128–132, San Diego, 2002.

[Dreyer and Marcu, 2012] Markus Dreyer and Daniel Marcu. Hyter: Meaning-equivalent semantics for translation evaluation. In *Proceedings of the The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL:HLT'12, 6 2012.

[Durgar El-Kahlout and Oflazer, 2006] ilknur Durgar El-Kahlout and Kemal Oflazer. Initial explorations in English to Turkish statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 7–14, New York City, 6 2006. Association for Computational Linguistics.

[El Kholy and Habash, 2010a] Ahmed El Kholy and Nizar Habash. Orthographic and Morphological Processing for English-Arabic Statistical Machine Translation. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-10)*, 2010. Montréal, Canada.

[El Kholy and Habash, 2010b] Ahmed El Kholy and Nizar Habash. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, 2010.

[El Kholy and Habash, 2011] A. El Kholy and N. Habash. Automatic Error Analysis for Morphologically Rich Languages. In *MT Summit XIII*, September 2011.

[El Kholy and Habash, 2012] Ahmed El Kholy and Nizar Habash. Rich Morphology Generation Using Statistical Machine Translation. In *Proceedings of 7TH International Natural Language Generation Conference (INLG 2012)*, Utica IL, USA, 2012.

[Farrus *et al.*, 2010] Mireia Farrus, Marta R. Costa-jussa, Jose B. Marino, and Jose A. R. Fonollosa. Linguistic-based evaluation criteria to identify statistical machine translation errors. In *Proceedings of EAMT*, pages 52–57, Saint Raphael, France, 2010.

[Flanagan, 1994] Mary Flanagan. Error classification for mt evaluation. In *Proceedings of AMTA*, pages 65–72, Columbia, Maryland, USA, 1994.

[Ford and Fulkerson, 1956] L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[Foster *et al.*, 2006] George Foster, Roland Kuhn, and Howard Johnson. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, Sydney, Australia, 7 2006. Association for Computational Linguistics.

[Galley and Manning, 2008] Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 847–855, Honolulu, Hawaii, 10 2008. Association for Computational Linguistics.

[Galley and Manning, 2010] Michel Galley and Christopher D. Manning. Accurate non-hierarchical phrase-based translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974, Los Angeles, California, 6 2010. Association for Computational Linguistics.

[Ganchev *et al.*, 2008] Kuzman Ganchev, Joao V. Graca, and Ben Taskar. Better alignments = better translations? In *Proceedings of ACL-08: HLT*, pages 986–993, Columbus, Ohio, 6 2008. Association for Computational Linguistics.

[Goldwater and McClosky, 2005] Sharon Goldwater and David McClosky. Improving Statistical MT Through Morphological Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 676–683, Vancouver, Canada, 2005.

[Graça *et al.*, 2010] João Graça, Kuzman Ganchev, and Ben Taskar. Learning tractable word alignment models with complex constraints. *Comput. Linguist.*, 36:481–504, 2010.

[Graff, 2007] David Graff. Arabic Gigaword 3, LDC Catalog No.: LDC2003T40, 2007. Linguistic Data Consortium, University of Pennsylvania.

[Habash and Hu, 2009] Nizar Habash and Jun Hu. Improving Arabic-Chinese Statistical Machine Translation using English as Pivot Language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181, Athens, Greece, March 2009.

[Habash and Rambow, 2005] Nizar Habash and Owen Rambow. 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'05)*, pages 573–580, Ann Arbor, Michigan, 2005.

[Habash and Sadat, 2006] Nizar Habash and Fatiha Sadat. Arabic Preprocessing Schemes for Statistical Machine Translation. In *Proceedings of the 7th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL06)*, pages 49–52, New York, NY, 2006.

[Habash *et al.*, 2007] Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer, 2007.

[Habash *et al.*, 2009] Nizar Habash, Owen Rambow, and Ryan Roth. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In Khalid Choukri and Bente Maegaard, editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*. The MEDAR Consortium, April 2009.

[Habash, 2007] Nizar Habash. Arabic Morphological Representations for Machine Translation. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer, 2007.

[Habash, 2010] Nizar Habash. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers, 2010.

[Hajič *et al.*, 2000] Jan Hajič, Jan Hric, and Vladislav Kubon. Machine Translation of Very Close Languages. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP'2000)*, pages 7–12, Seattle, 2000.

[Heafield, 2011] Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, UK, 2011.

[Heintz, 2008] Ilana Heintz. Arabic language modeling with finite state transducers. In *Proceedings of the ACL-08: HLT Student Research Workshop*, pages 37–42, Columbus, Ohio, June 2008.

[Henriquez *et al.*, 2010] Carlos Henriquez, Rafael E. Banchs, and José B. Mariño. Learning reordering models for statistical machine translation with a pivot language. 2010.

[Hopcroft *et al.*, 2006] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[Itai and Wintner, 2008] Alon Itai and Shuly Wintner. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March 2008.

[Jadidinejad *et al.*, 2010] Amir Hossein Jadidinejad, Fariborz Mahmoudi, and Jon Dehdari. Evaluation of PerStem: a simple and efficient stemming algorithm for Persian. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 98–101. 2010.

[Jurafsky and Martin, 2008] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd edition)*. Prentice Hall, 2008.

[Kathol and Zheng, 2008] Andreas Kathol and Jing Zheng. Strategies for building a Farsi-English smt system from limited resources. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH'2008)*, pages 2731–2734, Brisbane, Australia, 2008.

[Khalilov *et al.*, 2008] M. Khalilov, Marta R. Costa-jussá, José A. R. Fonollosa, Rafael E. Banchs, B. Chen, M. Zhang, A. Aw, H. Li, José B. Mariño, Adolfo Hernández, and Carlos A. Henríquez Q. The talp & i2r smt systems for iwslt 2008. In *International Workshop on Spoken Language Translation. IWSLT 2008, pg. 116–123.*, 2008.

[Kirchhoff *et al.*, 2007] Katrin Kirchhoff, Owen Rambow, Nizar Habash, and Mona Diab. Semi-automatic error analysis for large-scale statistical machine translation systems. In *Proceedings of the Machine Translation Summit (MT-Summit)*, Copenhagen, Denmark, 2007.

[Klein and Manning, 2003] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *IN PROCEEDINGS OF THE 41ST ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, pages 423–430, 2003.

[Knight and Marcu, 2005] Kevin Knight and Daniel Marcu. Machine translation in the year 2004. In *In Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP*, pages 965–968. IEEE Computer Society, 2005.

[Koehn and Hoang, 2007] Philipp Koehn and Hieu Hoang. Factored translation models. In *EMNLP-CoNLL*, pages 868–876, 2007.

[Koehn and Schroeder, 2007] Philipp Koehn and Josh Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227. Association for Computational Linguistics, 2007.

[Koehn *et al.*, 2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proc. NAACL-HLT 2003*, pages 48–54, 2003.

[Koehn *et al.*, 2005] Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proc. of the International Workshop on Spoken Language Translation*, 10 2005.

[Koehn *et al.*, 2007a] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, 6 2007. Association for Computational Linguistics.

[Koehn *et al.*, 2007b] Philipp Koehn, Hieu Hoang, Alexandra Birch, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, 2007.

[Koehn *et al.*, 2009] Philipp Koehn, Alexandra Birch, and Ralf Steinberger. 462 machine translation systems for europe. *Proceedings of MT Summit XII*, pages 65–72, 2009.

[Koehn, 2004] Philipp Koehn. Statistical significance tests formachine translation evaluation. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP'04)*, Barcelona, Spain, 2004.

[Koehn, 2010] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[Krovetz, 1993] R. Krovetz. Viewing Morphology as an Inference Process,. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 191–203, 1993.

[Kumar and Franz, 2007] Shankar Kumar and J Franz. Och, and wolfgang macherey. 2007. improving word alignment with bridge languages. In *Proceedings of the 2007 Joint Conference* *on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 42–50, 2007.

[Kumar *et al.*, 2006] Shankar Kumar, Yonggang Deng, and William Byrne. A weighted finite state transducer translation template model for statistical machine translation. *Nat. Lang. Eng.*, 12:35–75, 3 2006.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F.C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning. Morgan Kaufmann.*, pages 282–289, 2001.

[Lavie and Agarwal, 2007] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, 2007.

[Lee, 2004] Young-Suk Lee. Morphological Analysis for Statistical Machine Translation. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, pages 57–60, Boston, MA, 2004.

[Lopez, 2008] Adam Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3), 2008.

[Maamouri *et al.*, 2004a] Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The Penn Arabic Treebank : Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt, 2004.

[Maamouri *et al.*, 2004b] Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt, 2004.

[Madsen, 2009] Mathias Winther Madsen. The limits of machine translation. Master's thesis, Departement of Scandinavian Studies and Linguistics, Faculty of Humanities, University of Copenhagen, Copenhagen, 2009.

[Manning and Schütze, 1999] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[Mariño *et al.*, 2006] José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta Ruiz Costa-jussà. N-gram-based machine translation. *Computational Linguistics*, 32(4), 2006.

[Minkov *et al.*, 2007] Einat Minkov, Kristina Toutanova, and Hisami Suzuki. Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 128–135, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[Nießen and Ney, 2004] Sonja Nießen and Hermann Ney. Statistical Machine Translation with Scarce Resources using Morpho-syntactic Information. *Computational Linguistics*, 30(2), 2004.

[Och and Ney, 2002] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics.

[Och and Ney, 2003a] Franz Josef Och and Herman Ney. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29:19–51, 2003.

[Och and Ney, 2003b] Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–52, 2003.

[Och and Ney, 2004] Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4), 2004.

[Och *et al.*, 1999] Franz Josef Och, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 20–28, 1999.

[Och, 2003a] Franz Josef Och. Minimum Error Rate Training for Statistical Machine Translation. In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, 2003.

[Och, 2003b] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[Oflazer and Durgar El-Kahlout, 2007] Kemal Oflazer and Ilknur Durgar El-Kahlout. Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[Papineni *et al.*, 2002a] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, 2002.

[Papineni *et al.*, 2002b] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting on ACL*, pages 311–318, 2002.

[Paul *et al.*, 2009] Michael Paul, Hirofumi Yamamoto, Eiichiro Sumita, and Satoshi Nakamura. On the importance of pivot language selection for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 221–224. Association for Computational Linguistics, 2009.

[Popovic and Ney, 2006] Maja Popovic and Hermann Ney. Error analysis of verb inflections in spanish translation output. In *TC-STAR Workshop on Speech-to-Speech Translation. Barcelona, Spain*, pages 99–103, 2006.

[Rasooli *et al.*, 2013] Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaeid Moloodi. Development of a Persian syntactic dependency treebank. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, Atlanta, USA, 2013.

[Resampling, 1989] Bootstrap Resampling. Computer-intensive methods for testing hypotheses: an introduction. *Computer*, 1989.

[Roth *et al.*, 2008] Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL-08: HLT, Short Papers*, pages 117–120, Columbus, Ohio, 2008.

[Russell and Norvig, 2009] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition, 2009.

[Saralegi *et al.*, 2011] Xabier Saralegi, Iker Manterola, and Iñaki San Vicente. Analyzing methods for improving precision of pivot based bilingual dictionaries. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 846–856. Association for Computational Linguistics, 2011.

[Sarikaya and Deng, 2007] Ruhi Sarikaya and Yonggang Deng. Joint morphological-lexical language modeling for machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 145–148, Rochester, New York, April 2007.

[Shilon *et al.*, 2010] Reshef Shilon, Nizar Habash, Alon Lavie, and Shuly Wintner. Machine translation between hebrew and arabic: Needs, challenges and preliminary solutions. In *Proceedings of AMTA*, 2010.

[Shilon *et al.*, 2012] Reshef Shilon, Nizar Habash, Alon Lavie, and Shuly Wintner. Machine translation between Hebrew and Arabic. *Machine Translation*, 26:177–195, 2012.

[Simard *et al.*, 2005] Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. Translating with non-contiguous phrases. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 755–762. Association for Computational Linguistics, 2005.

[Singh and Habash, 2012] Nimesh Singh and Nizar Habash. Hebrew morphological preprocessing for statistical machine translation. EAMT, 2012.

[Smith, 2011] Noah A. Smith. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, 5 2011.

[Snover *et al.*, 2006] Matthew Snover, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *5th Conference of the Association for Machine Translation in the Americas (AMTA)*, Boston, Massachusetts, 8 2006.

[Snover *et al.*, 2009] Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Fluency, adequacy, or hter? exploring different human judgments with a tunable mt metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation at the 12th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2009)*, Athens, Greece, March 2009.

[Specia, 2011] Lucia Specia. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th conference of the European Association for Machine Translation*, pages 73–80, 5 2011.

[Stolcke, 2002] Andreas Stolcke. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO, 2002.

[Stymne, 2011] Sara Stymne. Blast: A tool for error analysis of machine translation output. In *ACL 2011 demonstration session*, Portland, Oregon, 2011.

[Tillmann, 2004a] Christoph Tillmann. A unigram orientation model for statistical machine translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

[Tillmann, 2004b] Christoph Tillmann. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 101–104. Association for Computational Linguistics, 2004.

[Tofighi Zahabi *et al.*, 2013] Samira Tofighi Zahabi, Somayeh Bakhshaei, and Shahram Khadivi. Using context vectors in improving a machine translation system with bridge language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 318–322, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[Toutanova *et al.*, 2008] Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[Tsvetkov and Wintner, 2010] Y. Tsvetkov and S. Wintner. Automatic acquisition of parallel corpora from websites with dynamic content. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 3389–3392, 2010.

[Ueffing *et al.*, 2002] Nicola Ueffing, Franz Josef Och, and Hermann Ney. Generation of word graphs in statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadephia, PA, July 6-7 2002.

[Utiyama and Isahara, 2007] Masao Utiyama and Hitoshi Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491, Rochester, New York, April 2007. Association for Computational Linguistics.

[Vilar *et al.*, 2006] David Vilar, Jia Xu, Luis Fernando DHaro, and Hermann Ney. Error analysis of machine translation output. In *Proceedings of LREC*, pages 697–702, Genoa, Italy, 2006.

[Wu and Wang, 2009] Hua Wu and Haifeng Wang. Revisiting pivot language approach for machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 154–162, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[Zens *et al.*, 2002] R. Zens, F. J. Och, and H. Ney. Phrase-based statistical machine translation. In M. Jarke, J. Koehler, and G. Lakemeyer, editors, *KI - 2002: Advances in Artificial Intelligence. 25. Annual German Conference on AI*. Springer Verlag, 2002.

[Zhu *et al.*, 2013] Xiaoning Zhu, Conghui Zhu, Tiejun Zhao, Zhongjun He, Hua Wu, and Haifeng Wang. Improving pivot-based statistical machine translation using random walk. 2013.

[Zollmann *et al.*, 2006] Andreas Zollmann, Ashish Venugopal, and Stephan Vogel. Bridging the Inflection Morphology Gap for Arabic Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 201–204, New York City, USA, 2006.