2-1-2017

Creating a strong statistical machine translation system by combining different decoders

Ayah ElMaghraby

The American University in Cairo

School of Sciences and Engineering

Creating a Strong Statistical Machine Translation System by Combining Different Decoders

A Thesis Submitted to the Department of Computer Science and Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Science

By
Ayah ElMaghraby

Under supervision of
Dr. Ahmed Rafea

DEDICATIONS

To my parents, thanks for insisting on me pursuing my graduate studies and teaching me that you stop learning when you die.

To my loving husband who worked around my schedule and did his best to offer me the environment I need to work on my thesis.

To my daughter who always shared my lap with a laptop, you are my inspiration to always do my best and set a role model for you.

Acknowledgements

I praise God for giving me the will to finish this thesis.

My professional gratitude goes to Professor Ahmed Rafea, who gave me the opportunity and the pleasure of working with him. Thank you for insisting I always did my best work and my best effort on this thesis.

Abstract

Machine translation is a very important field in Natural Language Processing. The need for machine translation arises due to the increasing amount of data available online. Most of our data now is digital and this is expected to increase over time. Since human manual translation takes a lot of time and effort, machine translation is needed to cover all of the languages available. A lot of research has been done to make machine translation faster and more reliable between different language pairs. Machine translation is now being coupled with deep learning and neural networks. New topics in machine translation are being studied and tested like applying neural machine translation as a replacement to the classical statistical machine translation. In this thesis, we also study the effect of data-preprocessing and decoder type on translation output. We then demonstrate two ways to enhance translation from English to Arabic. The first approach uses a two-decoder system; the first decoder translates from English to Arabic and the second is a post-processing decoder that retranslates the first Arabic output to Arabic again to fix some of the translation errors. We then study the results of different kinds of decoders and their contributions to the test set. The results of this study lead to the second approach which combines different decoders to create a stronger one. The second approach uses a classifier to categorize the English sentences based on their structure. The output of the classifier is the decoder that is suited best to translate the English sentence. Both approaches increased the BLEU score albeit with different ranges. The classifier showed an increase of ~0.1 BLEU points while the post-processing decoder showed an increase of between ~0.3~11 BLEU points on two different test sets. Eventually we compare our results to Google translate to know how well we are doing in comparison to a well-known translator. Our best translation machine system scored 5 absolute points compared to Google translate in ISI corpus test set and we were 9 absolute points lower in the case of the UN corpus test set.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Abbreviations

NLP	Natural Language Processing
SMT	Statistical Machine Translation
MT	Machine Translation
LM	Language Model
POS	Part of Speech Tags
SVO	Subject-Verb-Object
VSO	Verb-Subject-Object
SV	Subject-Verb
VS	Verb-Subject
KNN	K-Nearest Neighbors
NN	Neural Network
PCA	Principal Component Analysis
NER	Named Entity Recognition

Chapter 1 Introduction

George Steiner [1] an American-French literary critic once said “Every language is a world. Without translation, I would inhabit parishes bordering on silence”. Translation comes from the human need to understand and be understood. Without translation, a lot of the human history and knowledge would be lost. We translate ancient drawings and writings to understand human history and evolution. Stories and Novels are translated across languages to make people from different cultures and background enjoy all kinds of literature, and share the same human experiences in reading. We even went as far as marine biologists trying to use computers to translate dolphin whistles to understand their habitats and lifestyles. Translation was and will always be a critical part in our lives as human beings.

Most of our human knowledge now is in digital form stored in the cloud or on computers, so the need for computers to do the translation became much more. We have huge amount of data and information, using humans to do translation became very hard. Many researchers try to find and enhance current translation methods and provide an almost human like translation.

In this thesis, we try to study and find ways to enhance translation between English and Arabic languages. We will describe a technique to enhance statistical machine translation between English to Arabic. The introduction is organized as follows; first we talk about the background and history of machine translation. Second, we describe the problem definition then the motivation behind this research. Then the definition of the research questions I am trying to answer.

1.1 Background

In John Hutchins’s paper [1] he traced the beginning of Machine Translation (MT) could be traced back to 1930s when two patents were submitted simultaneously in Russia and France for two electro mechanical devices that can act as translating dictionaries. These patents were submitted by French-Armenian Georges Artsrouni and a Russian Petr Troyanskii. Astrouni proposed a general-purpose machine that could

[1] French-born American literary critic, essayist, philosopher, novelist, and educator https://literature.britishcouncil.org/writer/george-steiner

function as a mechanical multilingual directory. Troyanskii not only did he propose a mechanical multilingual directory, he also outlined how synthesis and analysis could work.

In John Hutchins's book "Early years in machine translation"[2], he attributed the interest in machine translation research was a result of a memorandum written by Warren Weaver in 1949. Earlier in 1947 Warren Weaver was director of the Division of Natural Sciences at the Rockefeller Foundation, mentioned the possibility of using digital computers to translate between human languages in a letter to Norbert Wiener. Then 1949, he wrote the memorandum entitled "Translation".

In 1950s there were early systems and pioneers in MT, and there was optimism among MT researchers, then in 1966 [3] came out the ALPAC report that ended a lot of funding for research in MT field in United States of America for many years. The report was very skeptical of the research done until that point.

Not till the 1970s did the research in MT field was revived, then appeared commercial and operational systems in the 1980s. A lot of new developments in the research occurred in the 1990s [4].

In the past few years with the increase use of social media like Facebook and Twitter, Natural Language Processing methodologies and algorithms became a very hot topic. The Arab spring also tempted a lot of research to do a lot of text analysis on news, tweets and posts. Statistical machine translation has always been a very important topic and area in NLP. There is a great necessity to automate translation till it reaches a decent quality. Translation is affected by many parameters, like domain specific knowledge, training data size, language structure and grammar, SMT type ...etc.

Human evaluation for translation is expensive and would take a large amount of time to go over thousands of sentences to evaluate them. One of the automated metrics used to evaluate how good a translation is BLEU (the bilingual evaluation understudy) score[5]. This is the metric used in this thesis to evaluate experiments.

1.2 Problem Definition

Machine translation faces many problems when translating between two language pairs. The problems get more severe when one of these two languages have great difference in its structure than the other such as English and Arabic languages. In Arabic, one can say

a whole English sentence by saying one word for example in English: "**We heard her**", the translation in Arabic would be: "سَمّعناها".

One of the problems noticed after translation was that the output sentence contained all the right words, but they were not ordered correctly, words were out-of-order. This led to loss of meaning and in some cases a very different meaning than the originally intended one.

Arabic words could be very ambiguous if taken out of context. This can be caused by spelling different script words. For example, the word "على" means the preposition on if the last letter is spelled differently it would change to "علي" which is the spelling of the name Ali or means the noun up.

Another issue noticed was getting lower BLEU scores when translating sentences that were not taken from the corpus used in training. This can be caused by unseen words or difference in topic between test set and training set. This is a well-known problem in machine translation, not having a generic enough SMT that could translate data from different topics or contexts efficiently. The issue with most sentences is that they cannot be fixed easily by analyzing the structure of the sentence and trying to modify them automatically in a post processing manner as there no pattern observed. In other words, there was no way to figure out the original sentence from the output.

1.3 Motivation

In the past few years most of the research was directed towards translating from Arabic to English that was mainly triggered by the Arab spring and the increase interest of analysis on Arabic tweets and posts. Figure 1 is presenting the number of papers published in Associations of Computations Linguistics Anthology [2] talking about SMT from English to Arabic and from Arabic to English. This is the first reason why my research focused on English to Arabic SMT.

[2] http://aclweb.org/anthology/

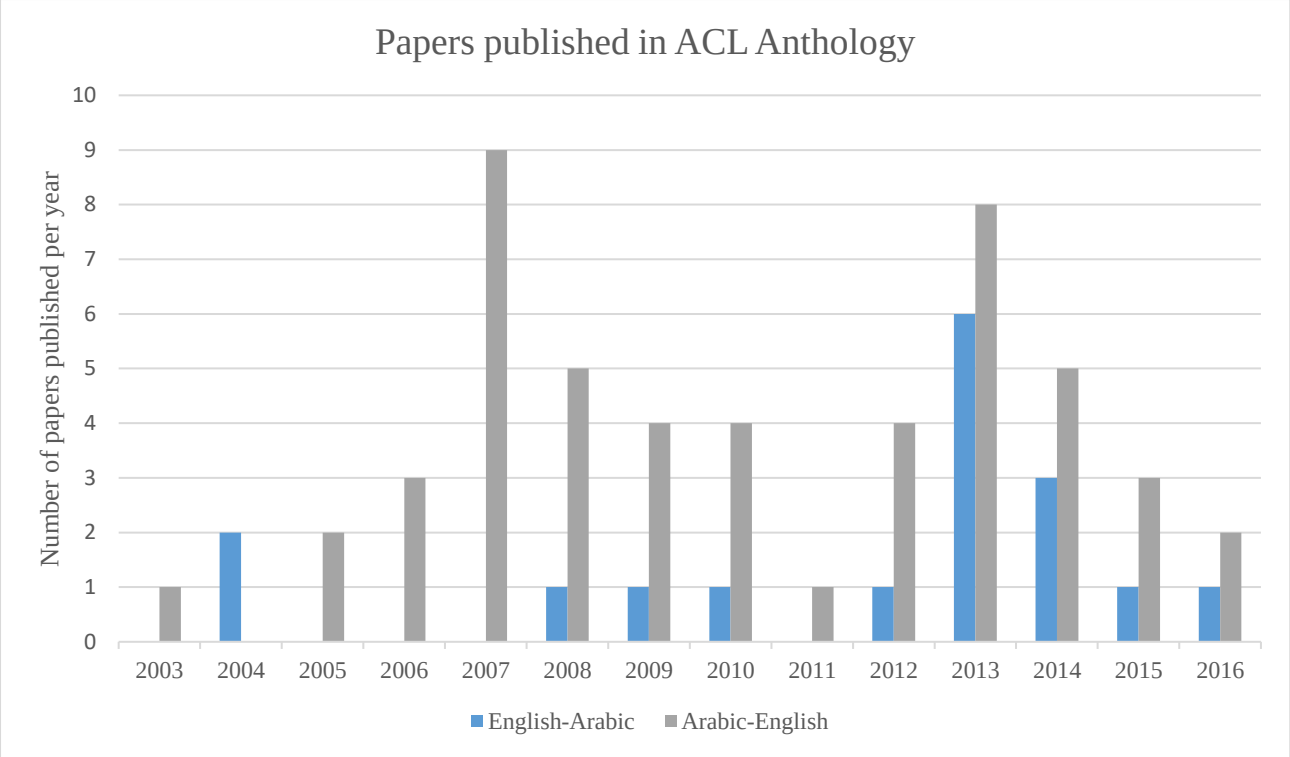

Figure 1 Number of papers published in ACL

1.4 Thesis Statement

The thesis statement is to scrutinize methods to improve translation quality from English to Arabic using statistical methods. To achieve this statement three objectives are identified:

1. Investigate different state of the art approaches to build a baseline system that produces the best results for English/Arabic translation using a small corpus. We investigate techniques like data pre-processing and choosing the best decoder type.
2. Explore the possibility of applying a post-processing machine translation system (Arabic to Arabic) for enhancing the translation quality of a test set of English sentences extracted from the same corpus or another corpus.
3. Explore the possibility of combining different decoders to create a stronger decoder which produces the best output.

1.5 Research Questions

To achieve the first objective these research question where identified:

- What is the impact of data preprocessing on translation quality?
- What is the impact of using different types of SMTs on quality of translation?

To achieve the second objective these research questions, need to be answered:

- Could post processing using another translation model built by an Arabic/Arabic corpus generated from the development data set to enhance the translation quality?
- Can the translation of sentences from different datasets not extracted from the corpus used in training be enhanced?

To achieve the third objective, the following question must be answered:

- Can the combination of these different decoders generate a stronger one that outputs the best result produced from these machines?

1.6 Thesis Layout

The thesis is organized as follows; the second chapter contains the literature review which describes machine translation challenges from English to Arabic, basics of statistical machine translation, and an overview on different techniques used to enhance machine translation from English to Arabic specifically and machine translation generally. The third chapter contains a description of our methodology, steps done to cover our research objectives and technology and datasets used in the thesis. The fourth chapter contains details of our experimental setup, experiments, their results and discussion of these results. The fifth chapter contains conclusions and future work recommendations. The fifth chapter is followed by the references used and two appendices. The first appendix contains a conversion table used in this thesis to convert POS tags to numbers. The second appendix contains a sample of translation enhancement done.

Chapter 2 Literature Review

Arabic is the native language for 300 million people in the world and is the official language for 27 countries. Arabic has 21 different dialects used across the 27 countries. Standard Arabic is a morphologically complex language. Arabic sentences can be in the form of SVO form and occasionally in VSO form especially if it is a passive sentence. As a result, a lot of research was done trying to translate Arabic whether in the standard form or a dialect, to and from English language and other languages.

The first machine translation system developed was rule based machine system it was developed in 1970s [6]. Till 1988, rule-based machine translation dominated the field of machine translation. Rule-based machine translation has lexical rules, rules for lexical transformation and rules of syntactic generation. It resulted in the evolution of some important systems like; systran [3]. In 1988 this paper [7] was published describing the basics of statistical machine translation in mathematical terms. They proposed the first statistical based approach developed in the candide project in IBM. Statistical machine translation has been a very famous approach to machine translation ever since. A lot of research was to enhance statistical approaches for translation between different language pairs. Most recently, a lot of research has been directed towards machine translation using neural networks. This approach is called neural machine translation. This approach was first introduced in [8]. They describe a model that has a conditioning and generation aspect. Their approach uses a recurrent language model that is based on a recurrent neural network to do the generation and the conditioning is done using a convolutional language model.

This chapter is organized as follows: the first part describes challenges of machine translation between English and Arabic. The second part describes basics of statistical machine translation. The third part describes some pre-processing techniques to enhance machine translation. The fourth part describes post-processing techniques to enhance machine translation. The fifth part describes using semantic and syntactic features to enhance machine translation. The sixth part describes some deep learning techniques to enhance translation. Finally, the seventh part describes a trial to combine multiple weak decoders to get the best translation out of one strong combined system.

[3] http://www.systran.de/

2.1 Challenges of Machine Translation between English and Arabic

Arabic is complex language, and the divergence between Arabic and English adds a great complexity when translating between the two languages. One word in Arabic can correspond to a complete sentence in English, for example the question “Do you know him?” is translated to “أتعرفونه؟”. As a result, when morphological analyzers aim to segment the Arabic language they are always faced with the issue of combining segmented Arabic words because some clitics can be both suffixes and prefixes. For example, a phrase like “Before your exposure” is translated to “قبل تَعرضك”, where letter “تَ” is a prefix added to “تَعرضك” to indicate that whoever is being spoken to is exposed to something, but if we make the letter “تَ” a suffix to the word “قبل” it will become “قبلتَ” which means I accepted, hence the whole sentence will become “قبلت عَرضك”, which means I accepted your offer. This therefore adds a lot of complexity to the translation process.

Arabic words could be ambiguous if taken out of context, for example the word “كتب” could mean the past tense verb “written” or the noun word “books”. So, the semantics of the word depend on the context of the sentence and what comes before or after the word.

One of the challenges of the Arabic languages is posed by the inconsistency of spelling some script letters, for example many times the letter alef can be written in many forms for example; “أ“, “ا” and “آ”. The first contains the hamza “ء” which changes the sound of the letter. The third form contain mad “~” which indicates this word should be pronounced with two alefs. As a result, this leads to problems in Arabic words spelling. A word like “آية” could be written using any of the previous forms of the letter ale even though the intended meaning is the same word which means a verse in the Quran. Changing “أ“, “آ” to “ا” is called normalization referred to in the literature is Reduced Normalization (RED). Sometimes the letter “ا” can be written in some words as dot-less yaa “ى” which is pronounced in the same way as the alef “ا”. Choosing appropriate “ا” and “ى” depending on the context is called Enriched Normalization (ENR) which was introduced in [9].

Another problem in English-Arabic translation is verb-subject order in the sentence. In Arabic it’s normal to have sentences in the form of subject-verb-object or verb-subject-object and both would translate to the same sentence in English. For example, the sentence “The boy goes to school” could be translated to “يذهب الولد إلى المدرسة” or “الولد يذهب إلى المدرسة”. In the first Arabic translation the verb comes before the subject and in the second example the

subject comes before the verb. Both translations are correct, but the subject-verb order is more common than verb-subject order.

Another difference between Arabic-English syntax is the structure of the noun phrase in both languages. In Arabic the definite article "ال" is appended to the noun while in English the definite article "the" is used before. In Arabic, if we add a definite article to a noun the adjective following it should have a definite article in it too. For example, the noun phrase "The big book" has one definite article while in Arabic it's translated to "الكتاب الكبير" which contains the definite article in both words in the noun phrase.

2.2 Basics of Statistical Machine Translation

Statistical machine translation contains three main parts: the language model, the translation model and the decoder. Statistical machine translation is the process of finding the most likely translation for a sequence of words in the source language. Basics of machine translation were discussed in [7]. Machine translation in mathematical terms can be explained as; given a source language s and a target language t we try to find the target language which maximizes the following equation:

$$argmax_t\big(P(s|t) * P(t)\big) \qquad (1)$$

The language model (LM) is trained on a monolingual corpus of the target language which is used to determine the probability of a sequence of words P(t) where t is the target language. While the translation model is trained on parallel corpus finds the probability P(s|t) where t is the target language and s is the source language. Decoding as seen in equation (1) is the product of the two probabilities of LM and translation model. In other words, the translation model tries to find the most probable list of words in target language given a source language.

The decoder can be one of two types i.e. phrase-based or syntax-based. Syntax-based decoders can be tree-to-string decoder, string-to-tree decoder or tree-to-tree decoder. The difference between phrase-based and syntax-based decoders is the training data used in alignment. Phrase-based decoders align phrases which could contain continuous sequences of words, whilst the syntax-based decoders align tree to phrase or tree to tree. Phrase-based decoders are currently the most common and successful technique used in SMTs. They have the advantage of providing a many-to-many translation, so it performs well when translating non-compositional phrases i.e. phrases that have a meaning only if combined like "rock and

roll" or "the game is up". Meanwhile, syntax-based SMTs provide a better translation for determiners and prepositions. It also provides a better syntactic re-ordering for sentences as trees preserve relations between words. Syntax-based SMTs aim at incorporating explicit syntactic representation in machine translation.

Statistical machine translation was enhanced drastically over the past few years. There have been many attempts to enhance machine translation. Some attempts were pre-processing done on training data whereas some proposed post-processing techniques and others suggested enhanced grammar extraction algorithms. More recently deep learning has emerged as solution to some machine translation problems. It is being used with statistical machine translation and in some cases replacing it all together and creating a new field called neural machine translation. One interesting approach we came across was the attempt to create a strong decoder by combining weak decoders. These approaches will be discussed in the next sections.

2.3 Pre-processing Techniques to Enhance Statistical Machine Translation

Some techniques when translating from Arabic to English focus on the re-ordering done while translating as in [10] where the basic idea is to minimize the amount of re-ordering during translation by displacing Arabic words in training text. In this paper, they try to tackle the shortcomings in phrase-based translation when translating Arabic to English. The main problem they aim at solving is the difference in the order between Arabic and English verbs which is caused by the differences between the two languages' structures. The technique they apply in this paper is as follows: first, words are annotated with AMIRA [11], which is the second generation of ASVM (Active Support Vector Machine) tools that is used for Arabic pre-processing to do tokenization, POS-tagging and shallow syntactic parsing. Second, data is aligned to create production rules that helps in re-ordering the Arabic words. Production rules, for example, could be of the kind: *"move a chunk of type T along with its L left neighbors and R right neighbors by a shift of S chunks"*. The chunks here refer to a syntax tree. These production rules move the verb phrase to the right by several S chunks. These production rules are applied to the Arabic side of the training data. Then a Moses [35] phrase-based system is created using the output of the Arabic training data. They observe that

the re-ordered system performs better than the baseline system by 0.8 BLEU score points. They tested their work on two parallel corpora provided by NISTMT-09[4].

Some papers focused on the VS order present in the Arabic language as this form is mainly present when used in passive tense. In this paper [12] the VSO construction is detected using a syntactic parser from training data (LDC2009E82) that has been aligned manually. They were able to detect the VS structure with F1 score equal to 63%. Then they are compared to the English translation to check if they are in the same order or inverted. Arabic sentences which have the VS constructions are then changed to SV constructions. This is done to make Arabic words closer to English words by reordering. They noticed that forcing re-ordering on all detected VS structures during training and testing have inconsistent effect on translation quality TER metric enhanced from 44.34 to 44.03 while BLEU score decreased from 49.21 to 49.09. When re-orderings were limited only to alignment either by re-ordering correctly detected or incorrectly detected VS constructions, they recorded improvements in the BLEU score in the range of 0.2-0.3 using this technique because English is mainly a SVO language which helps the alignment and hence enhances the whole translation quality. The decoder was created using Moses toolkit and aligned using GIZA++ system.

Another approach in the same area was done as shown in [13]. However, unlike the papers before, this paper focuses on the translation from English to Arabic. They investigate using re-ordering rules to enhance translation and examine the effect of morphological segmentation of Arabic combined with re-ordering rules. Their approach was based on two things; first, they segment Arabic using a morphological analyzer MADA[14] (Morphological Analysis and Disambiguation of Arabic), which is used to segment the Arabic side of the data before aligning it. Figure 2 shows an example of two Arabic words' segmentation mentioned in the same paper "وسيقابلهم" and "وبيده".

a. w+ s+ yqAbl +hm
and will meet-3SM them
and he will meet them

b. w+ b+ yd +h
and with hand his
and with his hand

Figure 2 Arabic segmentation example from [9]

[4] Newswire sections of LDC2006E93 and LDC2009E08

Since Arabic is segmented before aligning, the output of the decoder is segmented. This is then recombined using the approach explained in [15]. Recombining is one of the main challenges of this approach for three reasons mentioned in the paper. These challenges occur due to the complex structure of Arabic. For the English side, it is tokenized using the Stanford tagger then parsed using Collins parser before any NER information was added using the Stanford NER system. Pre-processed data is then aligned GIZA++ through the Moses toolkit, to create a phrase-based MT. They recorded a gain of 0.74 over the baseline system. They also proved that they can scale up to very big test sets.

In [16] they consider morphological tokenization like previous papers and orthographic normalization. Their baseline system has no tokenization schemes and they compare it to five tokenization schemes. These five schemes were represented in papers [17] and [15]. They apply RED and ENR on the baseline system and combine them with all the other schemes. All the experiments were conducted using Moses toolkit to create a phrase-based system. All the systems except for their baseline require de-tokenization process to get correct Arabic sentences. The de-tokenization technique they are using is proposed in [20]. The de-tokenization process they are using is based on a lookup table mapping from tokenized forms to de-tokenized forms. This table is created form their language model by processing it using MADA tool. In the case of the baseline system the ENR increases the BLEU score. In the case of all the case of all the other systems RED proves to increase BLEU score. The best de-tokenization system is the one proposed by paper [17] called TB.

In [18] they study which part of the phrase-based machine translation pipeline benefits more from tokenization of target language in this case Arabic. Their baseline system is phrase-based system which is trained on un-segmented data. Their second system contains alignment of segmented training data combined with a segmented language model. Their third system contains a segmented phrase-table which contains the segmented form of the word in the phrase-table as an extra factor and they combine it with a segmented language model. The third system is created by training decoder on segmented data they are calling it 1-best segmented. The output of this system is segmented so it needs de-tokenization. The fourth system is called lattice de-segmentation described in [19]. The segmentation is done by MADA morphological analyzer. The decoder is created using Moses decoder and aligned using GIZA++ system. They reported increase in BLEU score in the case of last system more than all the other systems. The increase was in the range of 2 BLEU points.

2.4 Post-Processing Techniques to Enhance Statistical Machine Translation

As demonstrated above, morphological analyzers and tokenization process are used a way to enhance statistical machine translation. As a result, a way to de-tokenization is always needed to recombine tokenized output. Some techniques focused on using the help of morphological analysis tools to enhance the output of SMT, one of these techniques is focusing on solving the problem of de-tokenization after applying some of the tokenization mechanisms and morphological analysis mentioned above. De-tokenization approaches could be as simple as concatenating morphemes based on token markers. Another more complicated approach is table-based de-tokenization which handles de-tokenization through a lookup table observed by looking at large amount of data. Another approach could be rule-based tokenization which contains hand written rules or regular expressions to handle de-tokenization.

In [21] a de-tokenization approach is proposed, by training a sequence model that predicts the original form of the tokenized word. A transducer is trained on a set of tokenized-de-tokenized set pairs aligned on a character level. The transducer used is DIRECTL+[5] a discriminative and character level string transducer. This approach tries to produce the results obtained by rule-based and table-based approaches but through training rather than depending on humans for definition rules and lookup tables. They used for training EM-based M2M-ALIGNER [22]. This approach has several advantages one of them is that it is language independent and the rules are learned automatically from examples without human intervention. They created a phrase-based SMT using Moses toolkit and aligned training data with GIZA++. They recorded an increase of BLEU score with ~2 points.

In [23] they are working with English-Finnish language pairs. They propose an interesting approach that combines post-processing morphology prediction with morpheme-based translation. They are using three baseline systems the first is a word-based model, the second is baseline is a factored model. The third is segmented translation model based on a supervised segmentation model they are calling it Sup. They build a segmented model they called it Unsup L-match. It's a HMM based segmentation model. It's trained on 5000 most common words. To get more coverage they further segmented any words that contained most common suffixes. They apply morphology generation as a post-processing step. They treat

[5] https://github.com/anoopkunchukuttan/directl-p

the morphology generation problem as a sequence learning problem. There are three phases in their prediction model. The first phase trains a MT system that produces morphologically simplified word forms in the target language. While in the second phase the output of the MT decoder is then tagged with a sequence of abstract suffix tags. The final phase takes the abstract suffix tags and maps them to word forms and rank these outputs. They call this prediction model CRF-LM. They scored highest BLEU score when using the Unsup L-match tokenization methodology. They also measure WER (word error rate) & TER (translation error rate) but the CRF-LM scored highest WER & TER.

2.5 Using Semantic and Syntactic Language Features to Enhance Machine Translation

A hybrid approach can be seen in [24] they are trying to translate from English to Arabic, the main idea of their technique is using a statistical featured language model to add the knowledge to help solve ambiguity in Arabic language by adding syntactic and grammatical features to each word. To handle the complexity of the Arabic language and the morphological complexity of the language, they segment Arabic words using MORPH2 [25] as a morphological analyzer. Using MORPH2, the words are first tokenized, then text is pre-processed to extract clitics. Then for each word it is determined if it contains prefix, infix or suffix. Last step is determining morph-syntactic features of each word like; gender, part-of speech (POS) tag, number, time person …etc. Since Arabic is segmented before training the output of this decoder is segmented Arabic so they use this form to re-combine Arabic words to try and resolve ambiguities resulting from this decomposition:

Proclitic + prefix+ stem +suffix + enclitic.

They create their language model using SRILM [26] and integrate the morphological features they obtained from MORPH2 in the language model. They represent the words in the LM (language model) in the form of n-vector containing n features. Training and decoding both are done on segmented Arabic, they use Moses Toolkit to create a phrase-based baseline system. A correct re-combination of Arabic segmented output is done after decoding. They recorded an increase in their BLEU score equal to 0.5 points.

Another approach was proposed in [27], assuming there is semantic similarity between words if they belong to the same topic, English words are clustered in classes for example {football, field, pitch} would belong to the same class. These classes are disjoint classes;

each class contains words sharing the same meaning. These classes are projected on Arabic language, then each word on English and Arabic side are replaced by their respective class identifier. These classes are determined based on the alignments offered by GIZA++ [6]. First, Arabic training data are pre-processed by segmenting them to recognize their composition. Second, English words are clustered into classes. Third, these classes are projected on the Arabic data using the alignment done in GIZA++. Fourth, they add to the phrase-based table the English and Arabic words and their corresponding classes. Fifth, training to create the SMT is done as usual using Moses toolkit to create a phrase-based decoder. Hence two phrase tables are present, one containing class identifier and the other word identities. Translation is then guided using the class identifier introduced in this mechanism. The system reported an increase in the BLEU score with a percentage of 0.3% when translating from English to Arabic and 1.4% when translating from Arabic to English.

2.6 Using Neural Network in Machine Translation

Neural Networks have proven to be very useful in learning different tasks like; face recognition, eye retina and object detection. There is a close analogy between neural networks (NN) and the human brain. The neurons in the neural network resembles a neuron in the human brain. Neurons receive signals from different body parts, then fires back different signals accordingly. For example; when the eye sees a dog, the picture is sent back to the brain and associates the picture with the name and different memories accordingly. Neurons in Neural Networks work in a similar way, when they receive the input, according to an activation function the neuron produces an output that contributes to the final output of the neural network. There is a new topic in machine translation called neural machine translation. Neural machine translation is based on using an encoder-decoder system. The encoder is a bi-directional recurrent neural network which reads a source sentence and outputs a context vector. The decoder is a recurrent neural network that reads the context vector and predicts the output sentence.

Some recent approaches try to combine deep learning using neural networks in the statistical machine translation. One of these approaches [28] uses a deep neural network in translation from English to French by using a deep long-short term memory (LSTM) to map the input sequence from the English language to a vector of a fixed dimension. Then another

[6] http://www.statmt.org/moses/giza/GIZA++.html

deep long-short term memory deep (LSTM) to decode a target sequence from the vector. LSTM is a recurrent neural network (RNN) architecture which are neural networks with loops in them. LSTMs are a special kind of RNNs which are better in handling large sequences. They are designed to remember information for long time. In this approach, the goal of the LSTM is to predict the probability of input sequence given the output sequence. Both sequences have different lengths. They also found that reversing the order of the words in the input sentence significantly increased the BLEU score. Reversing the words order introduced many short-term dependencies between source and target which helped the optimization later-on. They recorded an increase of approximately 3 BLEU points over the phrase-based system.

Another approach that uses neural network [29], builds its idea upon the fact that re-ordering syntax structure of the source language to resemble the target language so they use the neural network to perform the re-ordering. They use the neural network in the language model and depend on the language model to do the re-ordering. In this paper, they translate German-to-English and Italian-to-English. This re-ordering system can be described as RNN with a single hidden layer. The training is done by training the RNN on a sequence of input and the output is their possible permutations. The most likely permutation is calculated and use this permutation in the decoder. They recorded an increase of 1.3 ~2 BLEU points.

Another usage of neural network can be seen in [30], they use it as a data selection method. Data selection is variant of domain adaption field where they select sentences from the out-of-domain corpus to be added to the in-domain corpus. These sentences are needed to translate the in-domain corpus. In this paper they treat data selection as a classification problem. They compared data selection using neural networks with data selection using cross-entropy. For training the neural network they use two corpora one out of domain corpus and one in domain corpus. In this research they propose a neural network classifier using a convolutional neural network (CNN) and bi-directional long short-term memory (BLSTM). The input sentences are changed to word embedding using word2vector. The word embedding is then used as an input of the neural network. Then there is a fully connected layer. They apply *softmax* function on the output to normalize output to model the probability $pM(I \mid x)$ to a given sentence x, depending whether x belongs to the in-domain corpus I or not. They recorded that the BLEU score in the case of neural networks is better than cross-entropy.

Unlike the research before this was done on English-Arabic pair. In [31] they record the first results on English <-> Arabic translation. They do pre-processing on Arabic. They do simple tokenization using Moses script and compare it to Arabic tokenization using approaches in [15] &[18] . They also apply orthographic normalization described in [16]. They compare between neural machine translation and the conventional phrase-based machine translation while applying the pre-processing techniques mentioned above. They apply simple tokenization and true-casing on English side of the data. They are using a free implementation of the neural network[7]. The recorded increase of BLEU score is as much as +4.46 and +4.98 over the baseline in the case of phrase-based and neural network when applying orthographic normalization and morphological ware tokenization. Both phrase-based and neural networks perform comparably according to their results.

2.7 Creating a Strong Decoder from Multiple Weak Decoders

One way of enhancing machine translation was proposed in [32] by trying to create a diverse translation system from an existing single engine using bagging and boosting. Using ensemble learning, weak translation systems are created then they are used to learn a strong translation system. An overview of the approach is; first a combination system which contains several input systems is created. Second, a weak SMT is then trained on a distribution of the training set. Third, the training set is re-weighted and updated to generate a new training set using bagging and boosting weight updating methods, which is used to train another weak SMT. Finally, from the combination of the weak SMT, a strong SMT is generated. The last step is a bit tricky how to combine the system and join them into a strong system. So, in this paper they propose different methods to generate the strong system aka (combine the weak ones). They study six approaches to combine a strong system:

1- Sentence Level Combination

   This is the default method of the system. They propose a scoring function to find the final translation. The function consists of a weight representing the importance of this machine in the list and a variable representing the computation of n-gram agreement and disagreement features.

2- Minimum Bayes Risk Decoding (MBR)

   It selects the most probable translation based on probabilistic model.

3- Confusion Network and Indirect HMM

[7] https://github.com/nyu-dl/dl4mt-tutorial

They build their confusion network using indirect HMM model that learns their parameters from different resources like word surface/semantic similarity and distance-based distortion penalty.

4- Confusion Network and METEOR Alignment
   They tried a combination of confusion networks based on METEOR alignment.
5- Boosted Re-ranking
   They use the technique mentioned in [33] to do boosted re-ranking.
6- Discriminative Re-ranking
   They choose several features that indicates whether a candidate should be given a higher rank than the others.

Many other issues arise during training like adjusting training parameters to create a different SMT each time and re-weighting training sets to generate new ones. All these issues were considered and covered in depth in paper. They record an increase of 1 point in BLEU score in the best model.

2.8 Summary

Reference #	Title	Year	Results
[10]	Chunk-based Verb Reordering in VSO Sentences for Arabic-English Statistical Machine Translation	2010	- eval08nw: 43.1 to 44.04 - reo08nw: 46.9 to 47.51 - eval09nw: 48.13 to 48.96
[12]	Improving Arabic-to-English statistical machine translation by reordering post-verbal subjects for alignment	2010	- MT03: 45.95 to 46.33 - MT04: 44.94 to 45.03 - MT05: 48.05 to 48.69 - MT08nw: 44.86 to 45.06 - MT08wb: 32.05 to 31.96
[13]	Syntactic Phrase Reordering for English-to-Arabic Statistical Machine Translation	2009	- RandT Segmented 21.6 to 22.2 - RandT Un-Segmented 21.3 to 21.8 - MT05 Segmented 23.88 to 23.98

			- MT05 Un-Segmented 23.44 to 23.68 - NIST Segmented 22.57 to 22.95 - NIST Un-Segmented 22.4 to 22.95
[16]	Orthographic and morphological processing for English–Arabic statistical machine translation	2012	- Combined Arabic News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18) 24.63 to 26.51
[18]	What matters most in morphologically segmented SMT models	2015	- MT05 32.8 to 34.3 - MT08 15 to 16.4 - MT09 19 to 20.5
[21]	Reversing Morphological Tokenization in English-to-Arabic SMT	2013	- MT05 26.3 to 28.55
[23]	Combining morpheme-based machine translation with post-processing morpheme prediction	2011	- Europarl o SUP: 14.68 to 14.9 o UNSUP 14.68 to 15.09
[24]	Integrating morpho-syntactic features in English-Arabic statistical machine translation	2013	- IWSLT2010: 12.58 to 13.16
[27]	Arabic-English Semantic Class Alignment to Improve Statistical Machine Translation	2015	- IWSLT 2008: 12.86 to 14.07 - IWSLT2010: 9.3 to 10.91
[28]	Sequence to sequence learning with neural networks	2014	- ntst14: 33.3 to 36.5

[29]	Non-projective dependency-based pre-reordering with recurrent neural network for machine translation	2015	- EuroParl: 33 to 34.15 - News2013 18.8 to 19.28 - News2009 18.09 to 18.58
[30]	Neural Networks Classifier for Data Selection in Statistical Machine Translation	2016	- EuroParl 28.6 to 29.9
[31]	First Result on Arabic Neural Machine Translation	2016	- MT05 31.52 to 35.98 using phrase based decoder - MT05 28.64 to 33.62 using NMT
[32]	Bagging and Boosting statistical machine translation systems	2013	- MT05 29.23 to 30.05 - MT03 30.02 to 31.34 - MT04 30.56 to 31.77

Table 1 Summary of Literature Review

Chapter 3 Methodology

In this chapter, the proposed methodology is guided by answering the research questions posed. To answer the first and the second research questions of the first objective namely "What is the impact of data preprocessing on translation quality?", and "What is the impact of using different types of SMTs on quality of translation?" a baseline system was built, and the following experiments are to be conducted to:

- Test the effect of data pre-processing on quality of translation
- Test the effect of using different decoder types to enhance the quality of translation.

To answer the third and fourth research questions of the second research objectives namely: "Could post processing using another translation model built by an Arabic/Arabic corpus generated from the development data set, enhance the translation quality?" and "Can the translation of sentences from different datasets not extracted from the corpus used in training be enhanced?, a secondary decoder that translates Arabic text in one form to Arabic text in another form, was created and applied after translating English text to Arabic to enhance translation and the following experiments are to be conducted:

- Conduct experiments to measure the impact of another translation model built by an Arabic/Arabic corpus generated from the development data set to enhance the translation quality

- Conduct experiments to measure the enhancement of the translation of sentences not extracted from the training corpus using the approach mentioned in the above step.

To answer the research question of the fifth research objectives namely: "Can the combination of different decoders generate a stronger one that outputs better result than each decoder separately?" the following experiments are to be conducted to:

- Generate different translations for each sentence for a data set using different decoders and get the best Bleu scores for each sentence and the corresponding decoder

- Build a classifier that chooses the best decoder to translate a certain sentence and test the multi-decoder system

In the end we compare our results to a well-known translator like Google translate to find how well we are doing.

The following subsections will explain how the baseline system was built, the data preprocessing tools used, the translation models developed, how the translation results are to be enhanced using an Arabic to Arabic decoder, how to choose the best decoders to be included in the multi-decoder system, and how to generate a data set to train a classifier to choose the best decoder for each sentence.

3.1 Building Baseline System

This section explains how the corpus was divided into training, development, and testing datasets, how the language model was built, and how the decoder was chosen, and the corresponding translation model was developed.

Step 1 is to create the corpus used to build the baseline system. The corpus used is ISI Arabic-English parallel text[8], it was divided into three parts that were created randomly:

1. Training Data
   It is approximately 90% of corpus around ~1,000,000 sentences. This data was used to train the SMT that translates from English to Arabic.
2. Test Data
   It consists of exactly 2000 sentences. This data is used in testing the SMT system using the BLEU-4 score metric.
3. Development Data
   It consists of <10% of the corpus around ~90,000 sentences. It is used as in the first approach to train the Arabic/Arabic decoder. It is also used as a training data for the classifier in the second approach. This development was used to tune the results of the first decoder by creating a second decoder using them as explained in section 3.4. It is also used to create a classifier to choose the best decoder to translate the English sentence as explained in section 3.5.

The division proposed above, is done by creating a small java program that takes the corpus as an input and in every 110 sentences a sentence is picked and added to the development data set. Then every 250 sentences one sentence is picked and added to the test data set.

[8] LDC catalog Number: LDC2007T08. The data was extracted from news articles published by Xinhua News Agency and Agence France Presse.

Step 2 creating the language model that will be used by the decoder. Using the pre-processed Arabic part of the full ISI parallel corpora, we build the language model using built using IRSTLM [34] which is a tool used to create language model. Language model is created on the Arabic language as it is the target Language.

Step 3 building the translation model. The decoder uses the translation model consists of the language model to perform the translation processing the decoder. We build the decoder provided by using Moses [35] toolkit. Moses helps to build the translation model train the decoder on any pair of languages. It was used to create the translation model using the ISI corpus. Moses toolkit provides phrase based or syntax based and the decoder in both the baseline system and the post-processing technique. The decoders could be phrase based or syntax based. The syntax based decoders could be string-to-tree or tree-to-string or tree-to-tree. The best decoder or two will be chosen based on their BLEU score calculated for the test set. For the baseline system, the phrase based decoder is chosen for the baseline system. The syntax based decoder could be string-to-tree or tree-to-string or tree-to-tree. The Phrase based decoder is chosen.

Step 4, is measuring the performance of the SMT system, we used the BLEU calculator implemented in Stanford Phrasal[9] [36] because it is suitable for Arabic more than that of Moses. The Stanford Phrasal is a toolkit written Java provided by Stanford NLP group to create and train SMT. But I only use their BLEU calculator. We use BLEU-4 to score our translation systems.

3.2 Data Pre-Processing Techniques

All corpuses available must be pre-processed. Data must be cleaned, like removing empty sentences, tokenize sentences add spaces to separate words. Replace some characters like %, $, *, etc. There are two tools to pre-process the dataset that have been used to find out which is better in enhancing the translation quality: Moses toolkit tokenizer and Stanford tokenizer

Moses has a tokenizer and a cleaner. The tokenizer replaces some special characters like %, $, * ... etc. It also provides a text cleaner that removes all empty sentences and very long sentences. In the alignment of sentences to create a decoder we use Multithreaded GIZA++[10] (MGIZA++) which is an extension of the GIZA++ system aligner. Both systems GIZA++

[9] Stanford Phrasal is an open source SMT implementation.

[10] See footnote 6.

and MGIZA++ has limitation on sentence length that's why it is recommended to use the Moses cleaner script to limit sentence length to 81 words. At first, I used Moses to pre-process the dataset. But the BLEU score was very low, so I switched to Stanford Tokenizer. But I kept using the Moses cleaner.

Stanford CoreNLP [37] is a set of natural language processing tools provided by Stanford NLP group. I only used their tokenizer and parser. They provide tokenizers and parsers for several languages like English and Arabic. I used the tokenizer for English and Arabic side both. Stanford tokenizer has been available for years and they have a tokenizer dedicated for each language unlike Moses tokenizer which is very generic and didn't perform well on Arabic. Then after tokenizing test data with Stanford CoreNLP, we used the Moses cleaner to remove empty sentences and sentences that are longer than 81 words.

I used Stanford CoreNLP parser, to parse English and Arabic sides of language. The output of the parser, is then changed to the Moses tree input format. This is used when creating a syntax based tree and one or both sides of the input is tree.

3.3 Choose the Best Translation Model

Using Moses toolkit, four kinds of decoders are created:

1- Phrase Based decoder already create in the above steps.
2- Syntax Based string-to-tree decoder.
3- Syntax Based tree-to-string decoder.
4- Syntax Based tree-to-tree decoder.

To create Syntax Based decoders; one side of the training data or both sides should be parsed using any parser. I used Stanford shift/reduce parser [11] for parsing. It provides separate parsers for Arabic and English.

[11] https://nlp.stanford.edu/software/lex-parser.shtml

```
Tree t = tr.readTree();
int i = 0;
while (t != null) {
    bw.append("<tree label=\"TOP\"> " + "<tree label=\""
            + t.value() + "\"> ");
    result = traverseTree(t);
    bw.append(result);
    bw.append("</tree> ");
    bw.append("</tree>");
    bw.newLine();
    bw.flush();
    t = tr.readTree();
    i++;
    System.out.println("i = " + i);
}
```

Figure 3 Main Loop in Wrapper Function

```
public static String traverseTree(Tree t) {
    String result = "";
    List<Tree> children = t.getChildrenAsList();
    for (int i = 0; i < children.size(); i++) {
        if (children.size() == 0)
            return "";
        if (children.get(i).isLeaf()) {
            return children.get(i).value();
        } else {
            result += "<tree label=\"" + children.get(i).value() + "\"> "
                    + traverseTree(children.get(i)) + " </tree> ";
        }
    }
    return result;

}
```

Figure 4 Implementation of Traverse Function

Moses toolkit take parse trees as input, but they must be in a certain format. So, the output of the parser had to be post processed to be read by Moses. Moses toolkit provide their own wrapper for some well-known parsers like Berkeley Parser and Collins Parser. So, I wrote my own wrapper using Java language that transforms Stanford Tree output to Moses tree input. Snippets of the code showing how the wrapper work is shown in Figure 3 and Figure 4.

We created a set of syntax-based decoders the algorithm used to extract grammar must be chosen. This algorithm creates the grammar rules file that helps the decoder in translation to eliminate wrong translations.

Moses provides two grammar extraction algorithms:

1- GHKM extraction algorithm [38].
2- Chiang 2005 extraction algorithm [39].

We created a combination of syntax decoders using both extraction algorithms. We created two syntax based decoders (string-to-tree and tree-to-string) that use GHKM algorithm, along with three syntax based decoders (string-to-tree, tree-to-string and tree-to-tree) that use Chiang 2005 algorithm.

Moses doesn't support the combination of syntax-based tree-to-tree algorithm with GHKM algorithm. So, we created 6 decoders, and all of them are created using the same language model explained in section 3.1. We compare them in this step and choose a translation model or two that score the highest BLEU score(s).

Up until this point the test set we are currently using to choose the best model is the test extracted from the ISI corpus. For us to answer the fourth question in our research questions, we need first to test the best translation model found so far on a test set extracted from another corpus. For this purpose, we will be using the test set extracted from the UN corpus[12][40].

3.4 Enhancing the Translation Quality using Post-Processing Decoder

This approach is based on the idea of treating the output of the English/Arabic decoder as a different language that need to be translated to the expected Arabic side of the data. So, we created a post processing or we simple we call it a secondary decoder to enhance translation between them.

In Figure 5, there is a picture describing the idea of post-processing decoder. At the beginning we run the test set on our baseline decoder followed by the post-processing decoder.

[12] This is a corpus provided by the UN website https://conferences.unite.un.org/uncorpus . It contains the session logs of the United Nations.

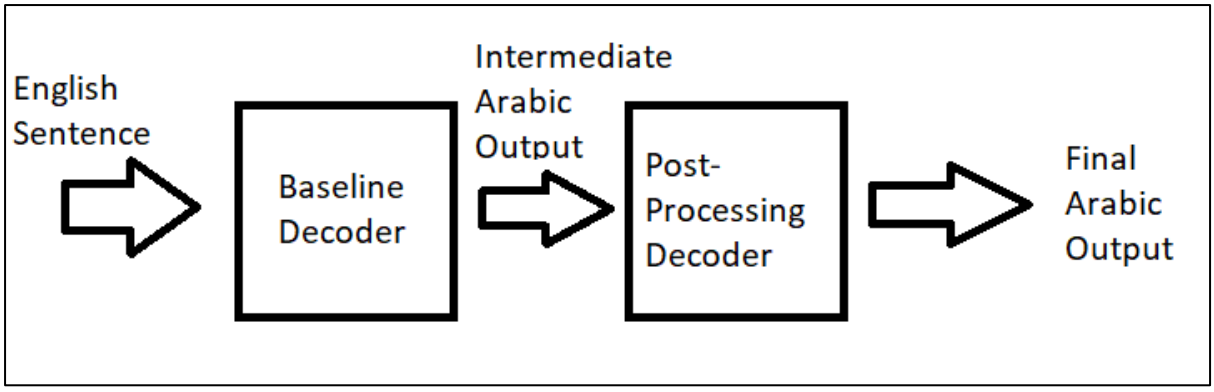

Figure 5 Post-Processing Decoding

The following steps are proposed to be followed to enhance the Arabic sentences translated and produced by the SMT system developed in step 3.3:

1- Create a post-processing decoder using the following steps:
   - Translate the English sentences in the development bilingual dataset using the syntax-based string-to-tree GHKM decoder.
   - Develop another translation system Arabic/Arabic taking the produced translation in the previous step as its source language and the Arabic sentences in the bilingual development dataset as its target language, and then measure the impact.

2- Test the effectiveness of post-processing decoder by:
   - Run the English/Arabic translation system on the test data set
   - Run the Arabic/Arabic translation system on the output produced by the English/Arabic translation system
   - Measure the impact.
   - Use the UN corpus [40] was used to examine whether the Arabic/Arabic translation system will also enhance the translation quality for this test data set extracted from a different corpus.
   - Extract from the UN corpus:
     - Around 2000 sentences randomly to create another test dataset to measure the performance of the developed system in 3.1
     - Around ~ 200,000 as another development set.

   The corpus is divided using the same program mentioned in section 3.1.
   - Repeat the experiment on a test data set extracted from another corpus.

3.5 Build Multi-Decoders System

In this step, I tried to propose a way to create a multi-decoder SMT that could be used to translate datasets from different contexts without the need to re-train a new system. The main idea that we noticed that some sentences are better translated using certain decoder while other sentences are better translated using another decoder. We wanted to choose the best decoders to be included in the multi-decoders system. Therefore, we decided to run the test data sets extracted from both the ISI, and the UN parallel corpus dataset and report the Bleu score for each sentence and for each decoder. We computed the average Bleu score for the best translation score for each sentence. These results were compared with the results of applying the Arabic/Arabic translation on the two datasets. The improvement noticed in the results encouraged us build a classifier to choose the best decoder for each sentence. The reported results were also used to choose the best decoders to be included in the multi-decoders system.

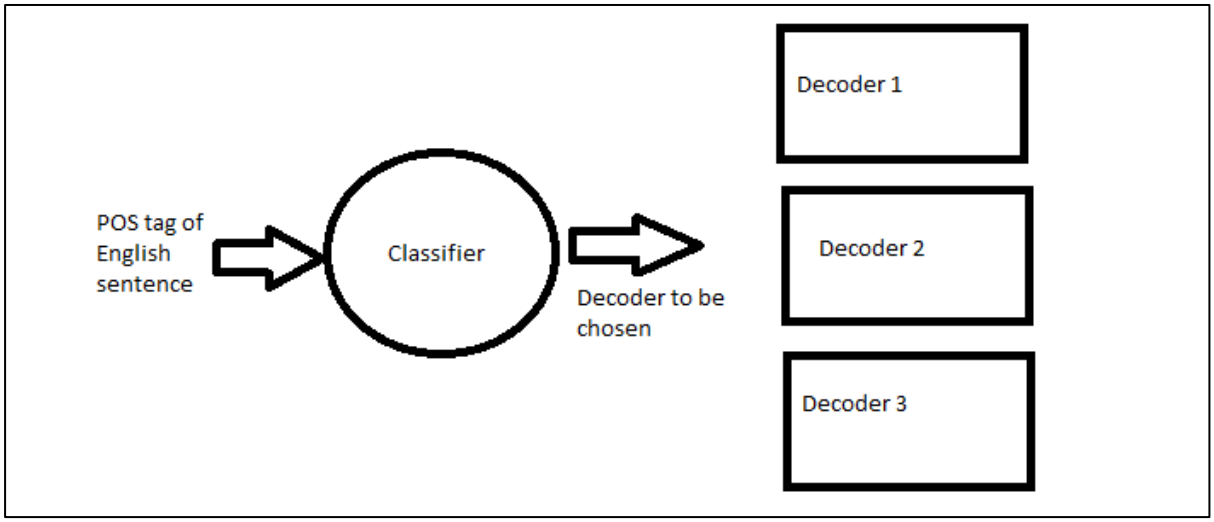

Figure 6 Proposed Multi-Decoder System

3.6 Generate a Data Set to Build a Classifier

This section describes how the training data and testing data were generated in the first two subsections then how two classifiers were built was given in the third and fourth subsections.

3.6.1 Create the Training Set

First, I created the training data by translating the development data set from the ISI corpus using three decoders (phrase-based, syntax-based string-to-tree using GHKM

extraction, and syntax-based string-to-tree using Chiang 2005 extraction). These decoders are chosen based on the applying the method described in section 3.5.

Second, I picked the best translation for each sentence and identified the decoder that produced this translation. Two files were created, the POS of the words of the English sentence being translated were considered as the input features of the classifier and the decoder that provided the best translation was considered as the class to be identified. The Stanford Parser was used to tag each word with its part of speech.

Third I convert the POS tags of each sentence to a numeric vector using Table 19 in Appendix I. These vectors are the classifier input. While the classifier expected output is the decoder number that produced the best translation for the English sentence. The size of the input vector was 81, and the size of the input data was approximately 74000.

3.6.2 Create the Test Set

Like what we did in the training set we create the test set. First, using the test set we extracted before from the ISI corpus, we run this test set using three decoders (phrase-based, syntax-based string-to-tree GHKM extraction and syntax-based string-to-tree Chiang 2005 extraction). Second, we find the highest BLEU score recorded for each sentence and record the decoder that got this score. Third, we find the POS tags for all the English sentences. Finally, our test set will contain the POS tags of 2000 English sentences in one file, and the decoder that got the highest BLEU score for each sentence in another file.

3.6.3 Train a K-Nearest Neighbors (KNN) Classifier

We used Weka platform [41], which was an open source platform that provides different classifier implementations. We used Weka to edit the training data. I created two versions of training data. The first one didn't contain the following POS tags like: ".","" -RBR-, -LRB-, because they denote square brackets, dots and commas. The second contains all these POS tags. I applied principal component analysis (PCA) using Weka which decreased the dimensionality of the vector input from 81 to 69. These two files are used to train a KNN classifier using Weka that choses the best decoder to translate an English sentence given its POS tags. I test the KNN classifier, by setting K =1, 3, 5, 10, 50, 100 and compare their results to each other and picked the best value for K that gives the highest BLEU score.

I trained the classifier to classify between 3 decoders (phrase-based, syntax based string-to-tree with GHKM extraction algorithm and syntax based string-to-tree with Chiang 2005 extraction algorithms) or 2 decoders (phrase based, and syntax based string-to-tree with GHKM extraction algorithm).

3.6.4 Train a Neural Network (NN)

The two files created in section 3.5.1 are the input and the expected output of the Neural Network (NN). The Neural Network is created using DeepLearning4j [42] platform. DeepLearning4j is a platform that helps in creating and running neural network.

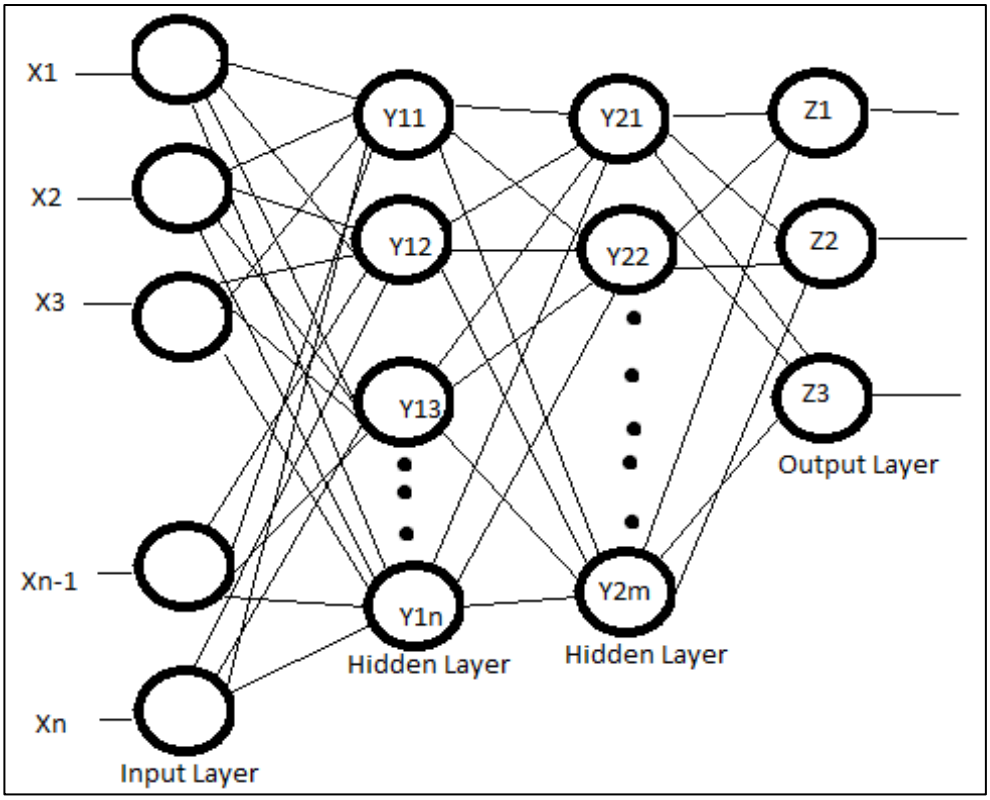

Figure 7 2 Hidden Layer Network

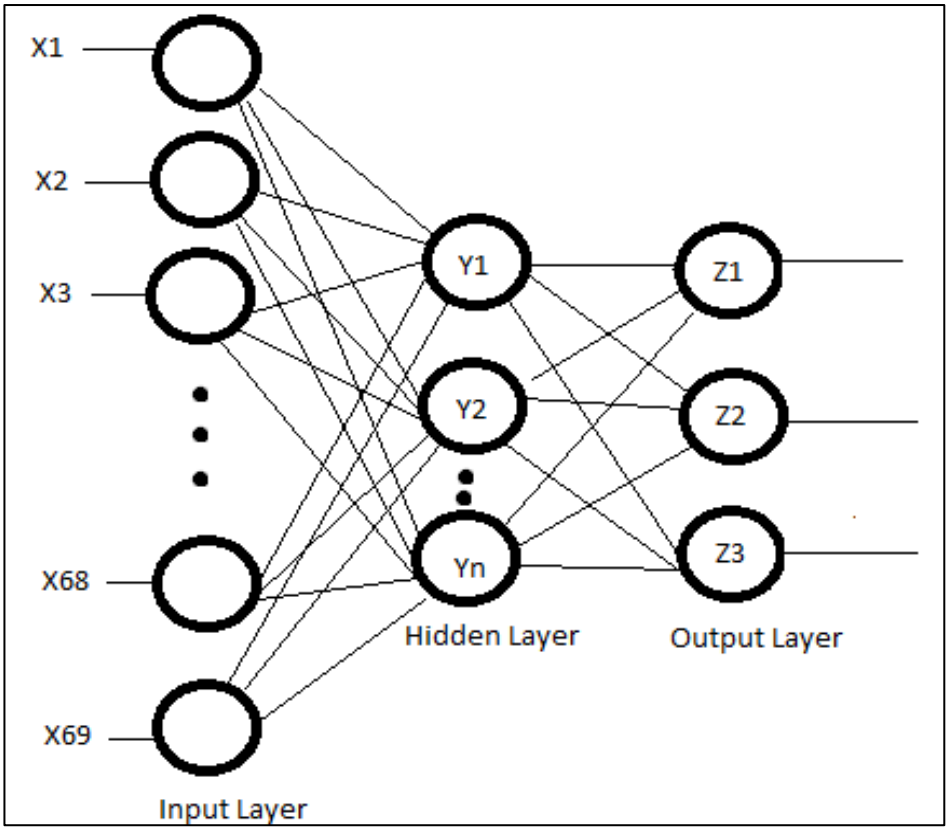

Figure 8 1 Hidden Layer Network

We first try training a two-hidden layer network like the one shown in Figure 7. Then a one hidden layer neural network as shown in Figure 8. We try different variables to tune the neural network. We set learning rate, change the number of hidden nodes and hidden layer. We tried to train neural network to classify between two decoders or 3 decoders. Activation function for output nodes is *softmax* functions and the rest of the nodes have an activation function of *tanh*. Weight initialization is *xavier* function. Both neural networks have back propagation.

Chapter 4 Experiments

In this chapter, we describe the experiments done to implement the methodology and the results acquired from those experiments. The first four experiments described in section 4.1 answer the research questions related to the first research objective namely "Investigate different state of the art approaches to build a baseline system that produces the best results for English/Arabic translation using a small corpus". In section 4.2, we answer the research questions related to the second research objective namely "Explore the possibility of applying a secondary machine translation system (Arabic to Arabic) for enhancing the translation quality of a test set of English sentences extracted from the same corpus or another corpus". In section 4.3, we answer the research question related to the last objective namely "Explore the possibility of combining different decoders to create a stronger decoder which produces the best output ".

4.1 Experiments to Choose Baseline System

In this section, the experiments done to develop a baseline system using the state of the art in statistical machine translation, are described. An elementary system was first developed, the effect of data pre-processing on quality of translation is measured, the effect of using different decoder types to enhance the quality of translation is also measured, and then the best two SMT systems were selected for identifying the SMT decoder that produces better results for a test data set extracted from a different corpus.

4.1.1 Experiment to Measure the Quality of Baseline System

Objective: Create an elementary baseline system and measure its quality

Training Set Used to Create the Decoder: Training set created from the ISI dataset.

Test Set Used: Test set created from the ISI dataset.

Method:

1- Clean corpus using Moses text cleaner.
2- Create a phrase based SMT using the above corpus to compare tokenizers.
3- Translate test set and calculate BLEU score using Stanford Phrasal [43].

Results:

	SMT with dataset not pre-processed
BLEU Score	6.56

Table 2 Baseline System

4.1.2 Experiment to Measure the Impact of Preprocessing the Bilingual Corpus

Objective: Study the impact of preprocessing on translation quality

Training Set Used to Create the Decoder: Training set created from the ISI dataset.

Test Set Used: Test set created from the ISI dataset.

Method:

1. Tokenize Arabic and English text once using Moses tokenizer and once using Stanford tokenizer.
2. Clean corpus using Moses text cleaner.
3. Create a phrase based SMT using the above corpus to compare tokenizers.
4. Translate the test dataset and calculate BLEU score using Stanford Phrasal.

Results:

	SMT with dataset tokenized by Moses toolkit	SMT with dataset tokenized by Stanford CoreNLP
BLEU Score	6.79	23.07

Table 3 Comparsion between Moses tokenizer and Stanford CoreNLP tokenizer

Conclusion: Stanford tokenizer is more suitable for Arabic language and it provides better results. Moses only separates punctuation marks they also don't support tokenization for Arabic official so they fall back to English tokenizer. It only replaces quotes with "&qout;". Stanford has a specific tokenizer for each language the Arabic tokenizer is in Utf-8 encoding.

4.1.3 Experiment to Measure the Performance of Different Translation Decoders

Objective: Study the impact of different translation models, and different grammar extraction methods on translation quality.

Training Set Used to Create the Decoder: Training set created from the ISI dataset.

Test Set Used: Test set created from the ISI dataset.

Method:

1. Create a phrase-based SMT
2. Create a string-to-tree SMT that extracts grammar using Chiang 2005 algorithm
3. Create a string-to-tree SMT that extracts grammar using GHKM algorithm
4. Create a tree-to-string SMT that extracts grammar using GHKM algorithm
5. Create a tree-to-string SMT that extracts grammar using Chiang 2005 HKM algorithm
6. Create a tree-to-tree SMT that extracts grammar using Chiang 2005 algorithm
7. Translate the test set 6 times using the above 6 SMTs
8. Calculate BLEU score for the 6 SMTs

Note: I cannot create tree-to-tree that extracts grammar using GHKM algorithm because it is not supported in Moses yet.

Results:

	Phrase-Based SMT from English to Arabic	String-to-Tree SMT from English to Arabic (Chiang 2005)	String-to-Tree SMT from English to Arabic (GHKM)	Tree-to-String SMT from English to Arabic (GHKM)	Tree-to-String SMT from English to Arabic (Chiang 2005)	Tree-to-Tree SMT from English to Arabic (Chiang 2005)
BLEU Score	23.07	19.68	21.05	16.94	17.9	16.91

Table 4 Comparison between SMTs with different extraction algorithms

Conclusion: From the results in both the above tables, I chose to complete my research with the best two configurations for the SMTs, phrase based SMT and string-to-tree SMT with GHKM extraction algorithm. According to the BLEU score the phrase-based decoder

seems to have a closer output to the expected data than all the other decoders. In Table 5 there is a sample of translations for each decoder.

The BLEU score only cares about the available words and their order only. So if a translation is semantically better but doesn't contain the expected words it could score less than a sentence that has some of the expected words but has grammatical issues or sematic issues. In Table 5, we demonstrate samples for translations by these 6 decoders. In the first sentence the phrase-based decoder has the most accurate translation compared to all the other decoders. But all the decoders has one issue in common is that they all have a problem with the second half of the sentence "او اننا تخلينا عن ارضنا للعدو". The phrase-based decoder is missing a pronoun and a preposition. While all the syntax decoder translations are using a wrong verb or a wrong verb tense. Among all the syntax decoders the string-to-tree decoder that uses the GHKM extraction algorithm is the second best. In the second sentence the phrase-based has the closer translation to the expected data. The second half of the sentence "انها تسيطر على اكثر من نصف البلاد" was translated wrongly also in this case for all decoders. This occurred also in the first sentence. The common thing between these two halves is that they are both noun phrases starting with the noun "ان". All the decoders seem to have an issue translating this kind of sentence but the phrase-based decoder seem to be the closest to the expected. Among all the syntax decoders the string-to-tree decoder that uses the GHKM extraction algorithm is the second best. The third sentence is a little longer than the previous two. For this particular sentence we have to mention the original English sentence "*He also said the clashes have stopped or seriously disrupted the UN aid programs, including water supplies to villages, food distribution and medical service*". All the decoders failed to translate the word "*stopped or seriously disrupted"*. This could have been caused by the language model because using the word seriously followed by a past tense verb isn't very common. Unlike the past two sentences the string-to-tree decoder that uses the GHKM extraction algorithm seems to provide the best translation in this case. This decoder is followed by the string-to-tree decoder that uses the Chaing 2005 extraction algorithm. Both these decoders have in common the Arabic side of the training data as parse trees. The phrase-based translation in this case has semantic issues that makes reading the sentence harder than that of the translation of syntax decoder.

Phrase-Based SMT from English to Arabic	String-to-Tree SMT from English to Arabic (Chiang 2005)	String-to-Tree SMT from English to Arabic (GHKM)	Tree-to-String SMT from English to Arabic (GHKM)	Tree-to-String SMT from English to Arabic (Chiang 2005)	Tree-to-Tree SMT from English to Arabic (Chiang 2005)	Expected
واضاف " هذا يعني اننا في حداد او اننا تخلوا عن ارض العدو ".	علينا ان " هذا يعني اننا في حداد او التي تصل الى ارض العدو ".	وهذا يعنى " اننا في حداد او اننا لم يتخل الى ارض العدو ."	واضاف " هذا يعنى ان اننا في حداد ، او ان نكون قد قدمت من الارض الى . " ان " العدو	هذا يعني " ذلك اننا في حداد ، او ان نكون قد منحت من الارض الى " .ان العدو	هذا يعنى ان " اننا في حداد ، او ان علينا اعطاء من الارض الى " . ان العدو	قال " ان هذا يعني اننا في حداد او اننا تخلينا عن ارضنا للعدو ".
وقد اعلنت الجبهة الوطنية الرواندية يوم الجمعة انه حصل على . نصف البلاد	وقد تبنى الجبهة الوطنية الرواندية اليوم الجمعة انه حصل على . نصف البلاد	وقد اعلنت الجبهة الوطنية الرواندية اليوم الجمعة انها قد فاز على . نصف البلاد	يذكر ان الجبهة الوطنية الرواندية اعلن امس الجمعة انها قد حصل على السيطرة على نصف . البلاد	يذكر ان الجبهة الوطنية الرواندية اعلن امس الجمعة انها قد حصل على السيطرة على نصف . البلاد	وكانت الجبهة الوطنية الرواندية اعلن امس الجمعة انها قد فاز السيطرة على . نصف البلاد	وقد اكدت الجبهة الوطنية الرواندية امس الجمعة انها تسيطر على اكثر من نصف . البلاد
وقال ايضا ان الاشتباكات اوقفت خطيرة الامم المتحدة ، بما فيها برامج المساعدات امدادات المياه الى القرى توزيع المواد	وقال أيضا أن المواجهات قد اوقفت او تعطيل خطيرة فى برامج مساعدات الامم المتحدة ، بما فى ذلك امدادات المياه الى القرى ،	وقال ايضا ان المواجهات قد اوقفت امدادات المياه الى قراهم او اعطال خطيرة ، بما فيها برامج مساعدات الامم المتحدة	وقال ايضا ان هذه الاشتباكات قد اوقفت او اعطال خطيرة فى الامم المتحدة برامج المساعدات ، بما فيها امدادات المياه الى القرى ، توزيع الاغذية	وقال ايضا ان هذه الاشتباكات قد اوقفت او اعطال خطيرة فى برامج مساعدات الامم المتحدة ، بما فيها امدادات المياه	وقال ايضا ان هذه الاشتباكات قد اوقفت او اعطال خطيرة فى الامم المتحدة برامج المساعدات ، بما فيها امدادات المياه	وقال ايضا ان الاشتباكات اوقفت برامج مساعدات الامم المتحدة بما فيها امدادات المياه الى القرى ،

برامج توزيع المساعدات الغذائية وتضررت الخدمات الطبية.	الى القرى ، توزيع المواد الغذائية والخدمات . الطبية	الى القرى ، توزيع المواد الغذائية والخدمات . الطبية	و الخدمات . الطبية	، توزيع المواد الغذائية والخدمات . الطبية	توزيع المواد الغذائية والخدمات . الطبية	الغذائية والخدمات . الطبية

Table 5 Sample of Translations

4.1.4 Experiment to Measure the Performance of Different Translation Decoders for Test Set Taken from Different Corpus

Objective: Test the performance of the best two decoders created in experiment on 4.1.3 on a test set from a different corpus**.**

Test Set Used: Test set created from the UN dataset.

Method:

1. Run phrase-based SMT decoder created in experiment 4.1.3 on UN test set.
2. Run Syntax-based (GHKM) decoder created in experiment 4.1.3 on UN test set.
3. Calculate BLEU score for the 2 SMTs

Results:

	Phrase-Based SMT from English to Arabic	String-to-Tree SMT from English to Arabic (GHKM)
BLEU Score	16.05	15.89

Table 6 Comparison between SMTs on UN test set

Conclusion: The Phrase-based SMT produced better results than the string-to-tree SMT with GHKM extraction algorithm. As shown in 4, the BLEU score of the translations scores of the phrase-based and syntax-based decoder are much lower than their corresponding translation scores in table 3. It is normal when translating a different test case from a different context than the training data, the BLEU score becomes lower than the test case initially used in testing.

4.1.5 Discussion

The basic idea behind the experiments described in the above subsections was to build the best baseline system using a small corpus and state of the art SMT tools. The baseline developed used Phrase-based SMT model, and Stanford tokenizer for data pre-processing. The Stanford tokenizer was far better than the Moses tokenizer for Arabic language. The other decoders examined were: string-to-tree SMT using Chiang 2005 extraction algorithm, string-to-tree SMT using GHKM extraction algorithm, string-to-tree SMT using GHKM extraction algorithm, tree-to-string SMT using GHKM extraction algorithm, tree-to-string SMT using Chiang 2005 extraction algorithm, tree-to-tree SMT using Chiang 2005 extraction algorithm. There was no tree-to-tree SMT from English to Arabic with GHKM extraction algorithm, so it is not in the list. The decoder that follows the phrase-based decoder was the string-to-tree SMT from English to Arabic using the GHKM extraction algorithm. So, all the comparisons in the following experiments were compared to the phrase-based decoder and string-to-tree SMT from English to Arabic using the GHKM extraction algorithm. We chose the string-to-tree SMT GHKM later to be used in our experiments to translate the development set to create the post-processing decoder. In Table 5, there is a sample of translations done by different decoders to show difference in decoders' behaviors when translating from English to Arabic. We tried to analyze the translations of these sentence in section 4.1.3. But it should be mentioned that BLEU score only cares about the sentences having the exact expected words and their order in the sentence. And the phrase-based translations seem to have the closer output to the expected data in most cases.

4.2 Experiments to Measure the Quality Translation Using a Post-Processing Decoder

In this section, I describe steps of two experiments done to measure the effect of applying a post processing decoder on two test sets extracted from the ISI parallel corpus and the UN parallel corpus.

4.2.1 Experiment to Measure the Impact of Post-Processing Decoder on Translation Quality

Objective: Study the impact of post processing technique on the quality of translation.

Training Set Used to Create Post-processing decoder: Development set created from the ISI dataset. Then we translate it using string-to-tree SMT created in experiment 4.1.3.

Test Set Used: Test set created from the ISI dataset.

Method:

1. Using the string-to-tree SMT with GHKM extraction algorithm I translate the development set of ISI parallel corpus. We used this decoder because the phrase-based decoder yielded worse results when used to translate the development set.
2. Train a post-processing SMT using the output from the first step with the target Arabic language. I also use the language model described in the methodology.
3. I create a 3 post-processing SMTs one that is phrase-based, one that is tree-to-string syntax SMT with GHKM extraction algorithm and the last one is tree-to-tree Syntax based SMT. Since I don't have a tree-to-tree Syntax based SMT with GHKM extraction I will use the available one with Chiang 2005.
4. Use the 3 post-processing SMT to translate test set.
5. I calculate the BLEU score for the 3 SMTs.

Results: Table 7 shows the results of re-translating the Arabic output from the test data set created from ISI Arabic-English parallel corpus. The Arabic to Arabic translation is done by using a syntax-based tree-to-string SMT.

	English to Arabic Translation	Arabic to Arabic translation
BLEU Score	Phrase-Based: 23.07	**23.1**
BLEU Score	Syntax-Based: 21.05	**21.09**

Table 7 Results of Arabic to Arabic Tree-to-String SMT applied on two different SMTs

Table 8 shows the results of phrase-based Arabic-to-Arabic SMT.

	English to Arabic Translation	Arabic to Arabic translation
BLEU Score	Phrase-Based: 23.07	22.77
BLEU Score	Syntax-Based: 21.05	20.09

Table 8 Results of Arabic to Arabic Phrase-Based SMT applied on two different SMTs

Table 9 shows the results of tree-to-tree Arabic-to-Arabic SMT.

	English to Arabic Translation	Arabic to Arabic translation
BLEU Score	Phrase-Based: 23.07	22.91
BLEU Score	Syntax-Based: 21.05	20.94

Table 9 Results of Arabic-to-Arabic Tree-to-Tree SMT applied on two different SMTs

Conclusion: Post-processing decoder has a very little effect on ISI test set because there wasn't any more missing info to be added by the post-processing decoder. In other words post-processing decoder is good when used a decoder adding extra information to be able to translate out of context copra and this could be seen in the next experiment.

4.2.2 Experiment to Measure the Impact of Post-Processing Decoder on Translation Quality for Test Set Taken from Different Corpus

Objective: Study the impact of post-processing technique on translation quality of a test set from different data set that is out of context.

Training Set Used to Create Post-processing decoder: Development set created from the UN corpus. Then we translate it using string-to-tree SMT created in experiment 4.1.3.

Test Set Used: Test set created from the UN corpus.

Method:

1. We create a test set from UN parallel corpus.
2. I create a development set from UN parallel corpus.
3. I translate the development set using string-to-tree SMT I created in experiment 4.1.3. Using other decoders to translate the development gives worse results when repeating this experiment.
4. I use the development translation and the target of the translation to create post-processing decoder.
5. I translate test set created from UN corpus and calculate BLEU score for it.

Results: Table 10 is the same as the above table but the Arabic to Arabic SMT is syntax based from tree to string.

	English to Arabic Translation	Arabic to Arabic Translation
BLEU Score	Phrase Based: 16.05	**23.51**
BLEU Score	Syntax-Based: 15.89	**19.29**

Table 10 Results of Arabic to Arabic Tree-to-String SMT applied on two different SMTs

Table 11 shows the experiment results of re-translating the Arabic output to fix the issues in it. This experiment was repeated on phrase-based and syntax-based machine. The Arabic to Arabic SMT in this experiment is phrase-based.

	English to Arabic Translation	Arabic to Arabic Translation
BLEU Score	Phrase Based: 16.05	**20.77**
BLEU Score	Syntax-Based: 15.89	**26.59**

Table 11 Results of Arabic to Arabic Phrase-Based SMT applied on two different SMTs

The experiment above is replicated but the Arabic-to-Arabic translation is done by a tree-to-tree SMT.

	English to Arabic Translation	Arabic to Arabic Translation
BLEU Score	Phrase Based: 16.05	**23.51**
BLEU Score	Syntax-Based: 15.89	**19.29**

Table 12 Results of Arabic to Arabic tree-to-tree based SMT applied on two different SMTs

Conclusion: The post-processing decoder has a better effect on the UN test set than the ISI test set. The post-processing decoder added the missing information needed by the baseline system to translate the UN test set.

4.2.3 Discussion

The impact of using post processing SMT system from Arabic to Arabic to enhance the translation quality an experiment was conducted that uses three post-processing decoders:

1- Syntax-based tree-to-string decoder.
2- Phrase-based decoder.
3- Syntax-based tree-to-tree decoder.

The post-processing syntax-based tree-to-string decoder enhanced the quality of BLEU score when translating the output of a phrase based decoder and the output of a syntax-based tree-to-string English to Arabic decoder by ~0.03-0.04 BLEU points over the baseline system.

Examples of problems that were fixed when using this approach:

- Some words and sentences weren't translated the first time and were output as English words:

Before	After
Human Rights Questions	مسائل حقوق الانسان
Provision ان لتغطية نفقات الخدمات المختلفة دولار شهريا 4500 حوالي	تغطية نفقات الخدمات المختلفة حوالي 4500 دولار للفرد شهريا

- The right gender of words like

Before	After
فريق الاتصال العسكرية	فريق الاتصال العسكري

- Fixes to translations like:

Before	After
قال ان النص النهائي، ان قرار المجلس	وللاطلاع علي النص النهائي انظر

- Fixes prepositions:

Before	After
مشاركة من السلطات المختصة من الولايات المتحدة	مشاركة السلطات المختصة في الولايات المتحدة

- Short term re-ordering like:

Before	After
المشروع شركة	شركة المشروع
مختلفة مجموعات	مختلف المجموعات

- Fixed some issue that caused semantic problems:

Before	After
ينص على هوية من مختلف المجموعات	فيما يتعلق بتحديد هوية مجموعات مختلفة

- Fixed issues with plural in Arabic like:

Before	After
الذين يأتي الى المكسيك المهاجرين	المهاجرين القادمين إلى المكسيك

The phrase-based decoder, when applied on the output of phrase-based and the syntax-based tree-to-string decoder, the BLEU score was lowered.

The syntax-based tree-to-tree decoder, when applied on the output of phrase-based and the syntax-based tree-to-string decoder that translate English to Arabic, the BLEU score was lowered.

The previous experiment was repeated on test data set extracted from a different corpus (UN) and the post-processing decoder was built using a small corpus extracted from UN corpus as well. The same three post-processing decoders used in the previous experiment were tested:

1- Syntax-based tree-to-string decoder.
2- Phrase-based decoder.
3- Syntax-based tree-to-tree decoder.

The BLEU score became higher using the first decoder just like what happened with the ISI parallel corpus. But this time the BLEU score increased by 4~11 BLEU points. The high increase could be attributed to having the first decoders trained on the ISI parallel corpus while the second decoder is trained on the UN parallel corpus. So, the second decoder adds some missing information and fixes translation issues caused by translation done initially using the first decoder.

In Table 20 in appendix II, we demonstrate some of the enhancements on the UN corpus with examples.

One last note is post-processing decoder has better results in the case of the UN corpus because the decoder is trained on the UN corpus and adds some missing basic information not found in the baseline decoder needed to translate the UN corpus. This is not the case when talking about the ISI corpus because there is no more missing basic info to be added by the post-processing decoder. Most of this basic information is already available in the baseline decoder.

4.3 Experiments to Describe the Steps to Create Multi-Decoder System

To create a multi-decoder system, it was necessary to run an experiment to get the Bleu score of each sentence translated by each decoder for the test data sets extracted from the ISI and UN corpora and to find whether an improvement could be obtained by translating each sentence using an appropriate decoder. Then an experiment was conducted to train a classifier using the data generated from the first experiment to test whether the classifier could identify the appropriate decoder given a sentence as input.

4.3.1 Study the Contributions of Different Decoder Types to Translation of a Data Set

Objective: find whether an improvement of translation quality could be obtained by translating each sentence in a data set using an appropriate different decoder

Data Sets Used: Test sets created from the ISI dataset and UN corpora

Method:

1. Translate the test data sets using the following decoders
    a. Syntax Based GHKM followed by tree-to-tree syntax-based Intermediate Decoder.
    b. Phrase-based followed by tree-to-tree Syntax Based intermediate decoder
    c. Syntax-based GHKM followed by tree-to-string syntax-based intermediate decoder (UN)
    d. Phrase-based followed by tree-to-string syntax-based intermediate decoder (UN)
    e. Syntax-based GHKM followed by phrase-based intermediate decoder (UN)
    f. Phrase-based followed by phrase-based intermediate decoder (UN)
    g. Syntax-based Chiang English to Arabic tree-to-tree
    h. Syntax-based Chiang English to Arabic tree-to-string
    i. Syntax-based GHKM English to Arabic tree-to-string
    j. Syntax-based Chiang English to Arabic string-to-tree
    k. Syntax-based GHKM English to Arabic string-to-tree (UN)
    l. Phrase-based English to Arabic (UN)
2. Calculate the BLEU score for each sentence in each of the 12 results created above.
3. Calculate the overall Bleu score of the data set by taking the best Bleu score of each sentence.

{Note: Only 6 are charted for data set extracted form UN corpus because all other decoders produced zeroes. Those used for UN data set are marked in front of each decoder used}

Results: The final BLEU score for the best translation of the ISI data set is 26.39. The best translation of this data set using phrase based decoder followed by a syntax based Tree to string decoder, was 23.1. The two graphs below show the share of each decoder in the results of the best translation. The first graph shows the results of the decoders stacked next to each other.

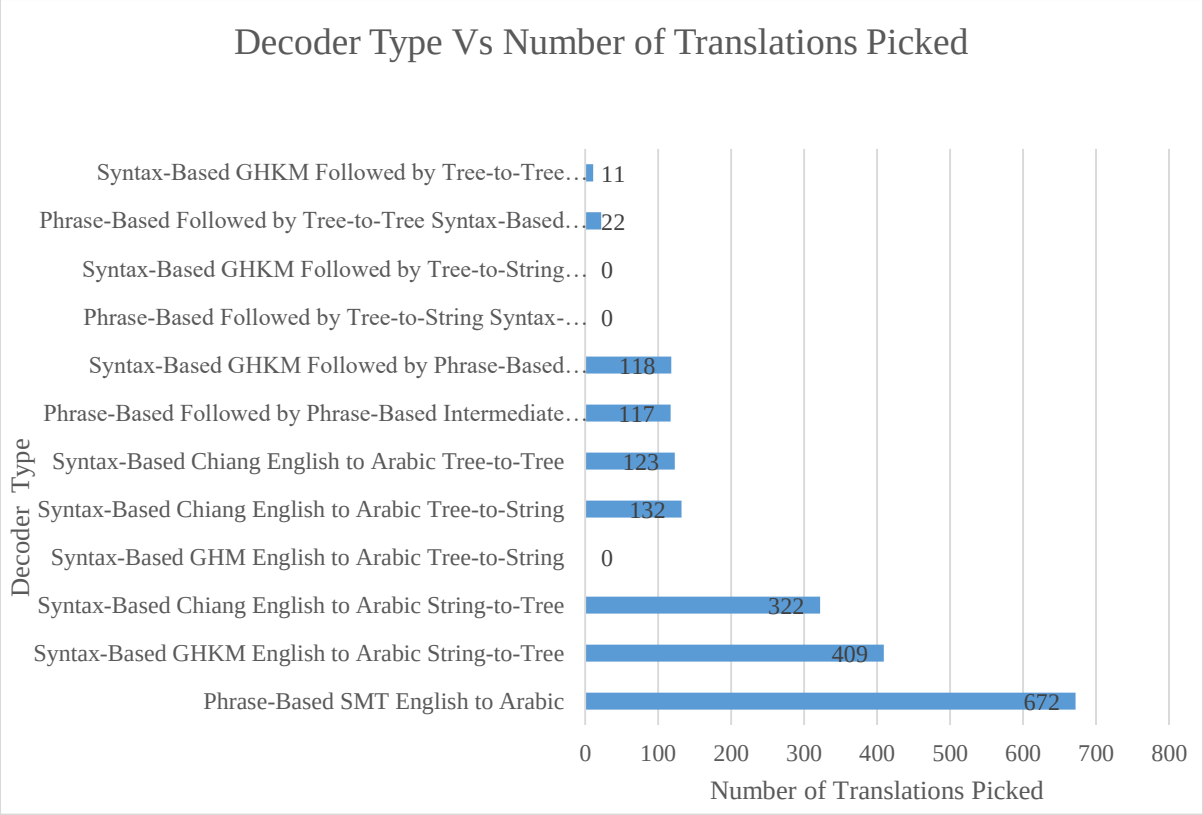

Figure 9 Bar Chart showing contribution of different decoders in ISI test set

The second chart is a pie chart displaying the results in a percentage form. From pie chart below, I can see that I only need 3 machines to cover as close as 73% of the test set. The phrase-based decoder English to Arabic, syntax-based GHKM English to Arabic string-to-

tree and syntax-based Chiang English to Arabic string-to-tree decoder.

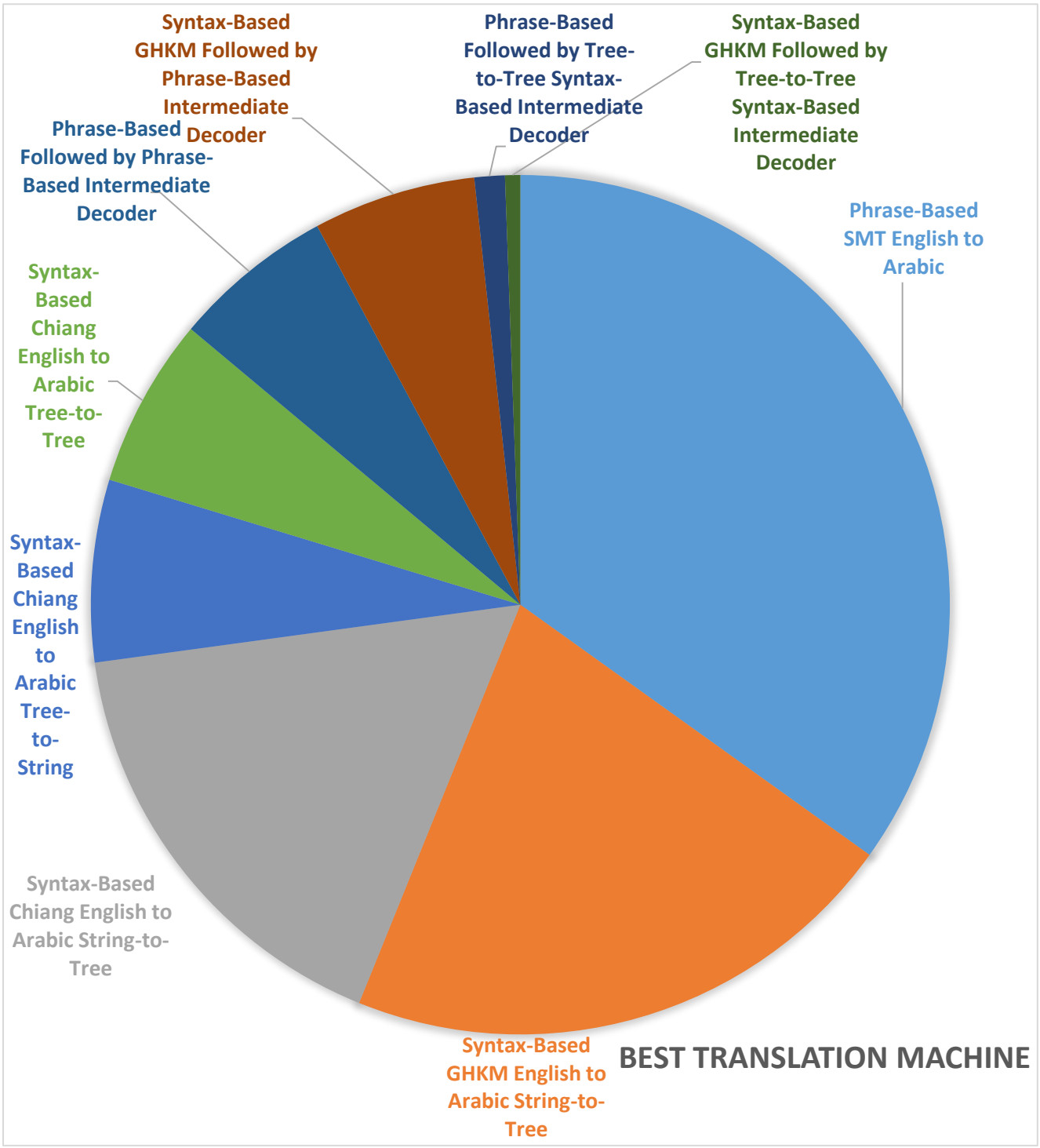

Figure 10 Pie Chart of Best translation for ISI test set

The Charts below show the contributions in the case of the UN corpus. Please note that UN corpus only 6 are charted because all other decoders produced zeroes.

Results: The final BLEU score for the best translation is 30.2. The two graphs below show the share of each decoder in the results of the best translation. The first graph shows the results of the decoders stacked next to each other. The second graph shows the percentage of the contribution of each decoder to the result.

The second chart is a pie chart displaying the results in a percentage form. From pie chart below, I can see that I only need 3 machines to cover as close as 74% of the test set. The phrase-based decoder English to Arabic, syntax-based GHKM followed by tree-to-string syntax-based intermediate decoder and syntax-based GHKM followed by phrase-based intermediate decoder.

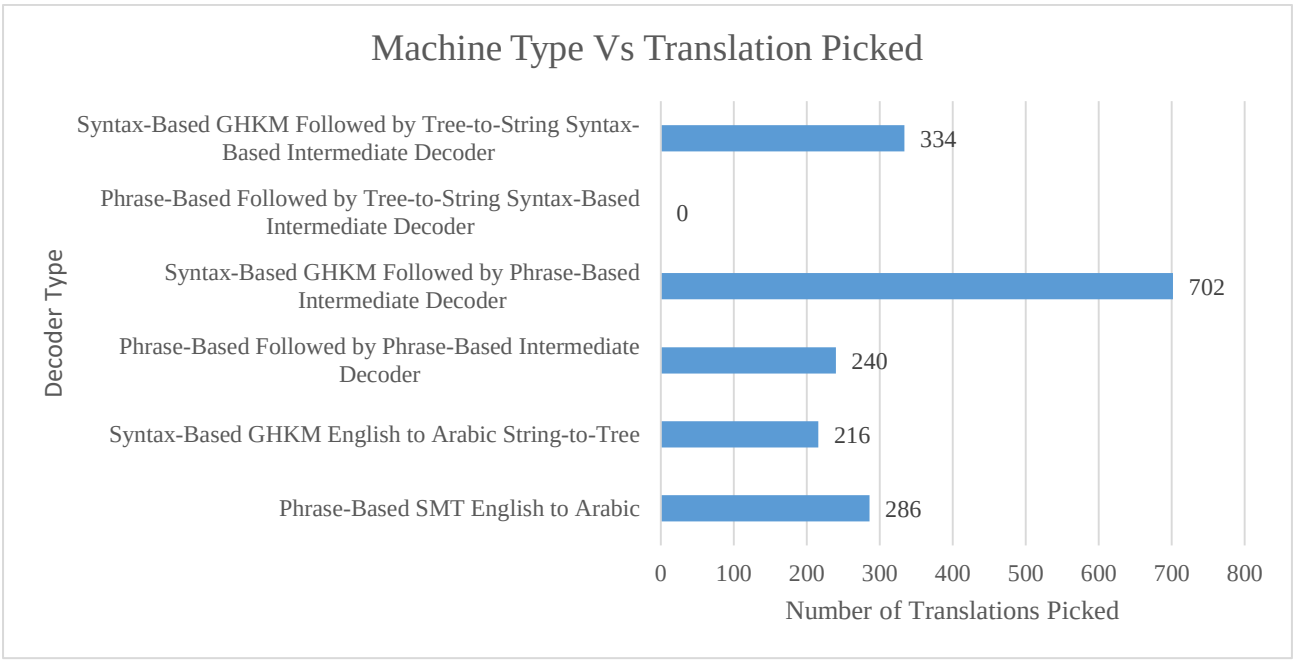

Figure 11 Bar Chart for Decoder Contribution in UN test set

In the first graph, we have 6 decoders and in the pie chart you will see only 5. The one that gave zero was eliminated in the 2nd graph

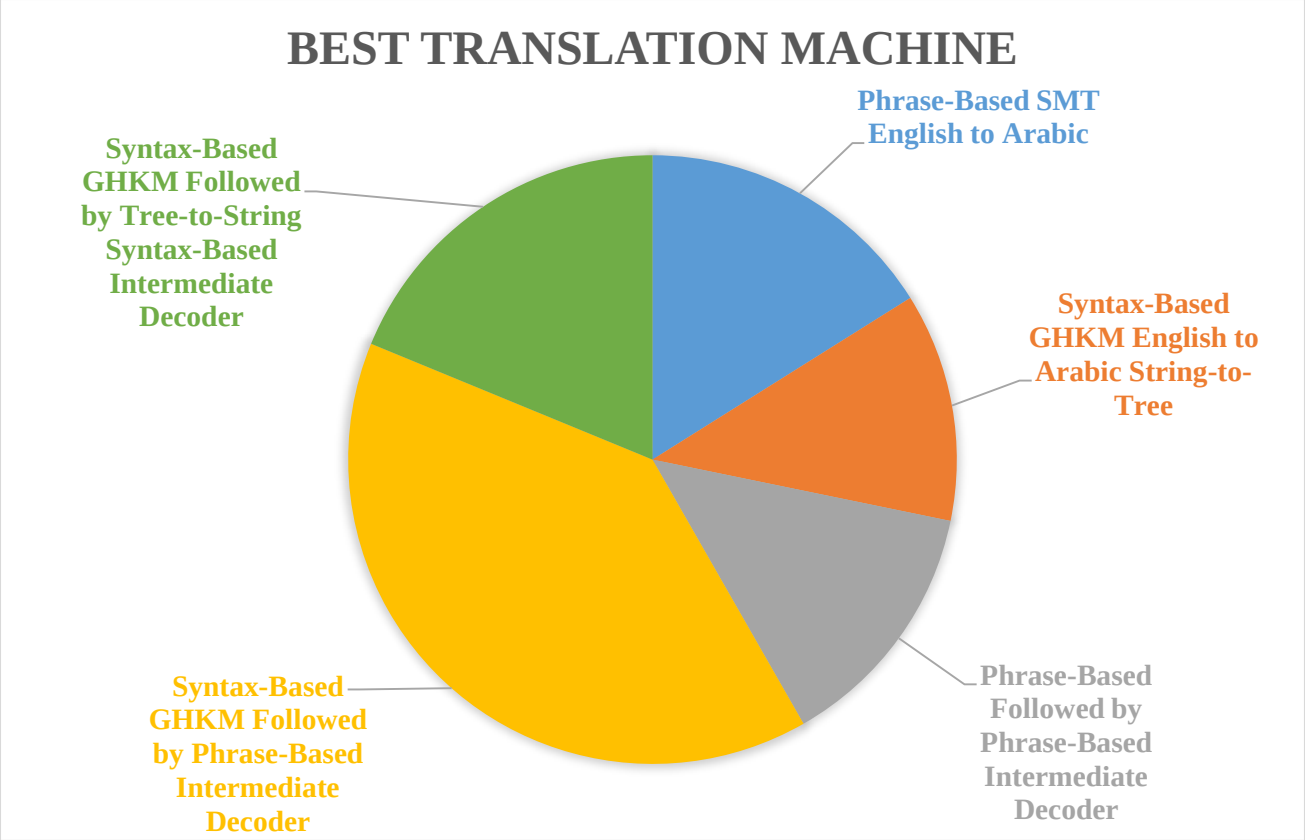

Figure 12 Pie Chart for Decode Contribution in UN test set

4.3.1.1 Discussion

Form the graphs shown above, it is obvious there is no single decoder that can significantly out-perform all others on all sentences in a certain test set.

From Figure 9 and Figure 10, we can see that the phrase-base does better than the other decoders on 35% of the sentences, and this is the highest decoder of all the others. The maximum BLEU score that could be obtained by combining all these decoders is 26.39. But using the phrase-based decoder along with the syntax based string-to-tree decoder with GHKM extraction algorithm and the syntax based string-to-tree decoder with Chiang extraction algorithm we could cover around 70% of the test set. Hence the maximum BLEU score then would be 25.4. Meaning, theoretically on these 70% we could get the best score possible if we could combine all these decoders, and translate each sentence with its perspective decoder that gives the best output. The problem is; in real life we don't know the best decoder for each sentence, so the solution for this issue could be training a classifier that chooses a decoder for each sentence to translate. The decoder will be trained on the English sentences since this is our input.

In Figure 11 and Figure 12, a different distribution for the decoders and their performance on the sentences. This time the output of the post-processing decoders is better than the single decoders. Which clearly shows the effect of using a post-processing decoder on the UN corpus. The maximum BLEU score that could be obtained by combining all these decoders would be 30.1.

4.3.2 Experiment to Build a Classifier

In this section, using the analysis of the first approach, I propose a way to automate choosing the best decoder to translate the sentences as described in section 3.5.

In this section, we try to create a multi-decoder system that will combine the 3 decoders discussed in experiments 4.3.1. We will only describe a multi-decoder system on the ISI corpus. The training sets for these classifiers are created by translating the development set 3 times using different decoders (phrase-based, syntax based string-to-tree with GHKM extraction algorithm and syntax-based string-to-tree with Chiang 2005 extraction algorithm) as described in section 3.6.1 in the methodology. The maximum BLEU score we got when only using these three decoders is 25.4. Our feature vector is created by finding the POS tagger using Stanford CoreNLP. Then transforming it into number according to the table in Appendix I.

4.3.2.1 K-Nearest Neighbors Algorithms

Objective: Classify English sentences based on their structure to be translated by 3 or 2 decoders. We try to configure K-nearest neighbor algorithm to reach the highest possible BLEU score.

Training Set Used: Development set created from ISI corpus.

Test Set Used: Test set created from the ISI corpus.

Method:

1- Use the development set created from ISI corpus as a training data for creating KNN classifier.
2- Create a KNN classifier using Weka platform on the training data. Training data is changed and tuned till we reach a high BLEU score.
3- Try K =1, 3, 5, 10, 50, 100. Stopped when BLEU stopped increasing.
4- Run classifier on test data.
5- Calculate BLEU Score for test data after running KNN.

Note: In the results section these abbreviations are used P is precision, R is recall, F1 is F1 score & MAE (Mean Absolute Error). MAE is calculated by Weka, so we just use it. The equation used to calculate P, R, and F1:

$$\mathbf{P}= \frac{true\ positive}{true\ positive+false\ positive}$$

$$\mathbf{R} = \frac{true\ positive}{true\ positive+false\ positive}$$

$$\mathbf{F1}= 2 * \frac{P*R}{P+R}$$

In case of 3 class classifier, we calculate precision and recall by creating the confusion matrix 3 times. Each time we make one class positive while the other two are negative. We then calculate the precision and the recall for each confusion matrix and then average all three values for precision and recall.

Results:

Table 13 shows the results of KNN classifier that is trained on a training set with full feature vector. Lowest error is recorded on least BLEU score, this happens because the error is inversely proportional to the accuracy of the classifier, while the BLEU score represents the quality of translating a sentence. However, the F- score makes more sense because the better the F-score the better the BLEU score. This indicates that when the classifier chooses the appropriate deodar the translation improves.

K-Value	1	3	5	10
P	0.346	0.334	0.332	0.335
R	0.345	0.339	0.326	0.332
F1	**0.346**	0.337	0.329	0.333
MAE	0.9045	0.8559	0.8375	**0.8149**
BLEU score	**21.45**	20.5	20.2	19.77

Table 13 Results of Training Data with full size feature vector

Table 14 shows the results of KNN classifier that is trained on a training set after applying best first algorithm provided by Weka tool. Using Weka, one can apply different attribute selection algorithms provided by the tool. In this table the mean absolute error occurs with the best BLEU score unlike the table before. Although this makes sense, but the best BLEU score didn't happen with the best F-score which isn't in consistent with the results in table 11.

K-Value	1	3	5	10	50	100	200
P	0.331	0.334	0.338	0.335	0.344	0.35	0.341
R	0.332	0.334	0.338	0.336	0.349	0.362	0.358
F1	0.332	0.334	0.338	0.335	0.346	**0.356**	0.349
MAE	0.774	0.545	0.502	0.458	0.407	0.398	**0.393**
BLEU score	21.78	21.85	21.86	21.94	22.2	22.44	**22.68**

Table 14 Results of KNN classifier after applying best first algorithm on training data

Table 15 shows the results of KNN classifier that is trained on a training set after applying PCA algorithm provided by Weka tool. Using Weka, one can apply different attribute selection algorithms provided by the tool. This table shows a similar relation between MAE & BLEU score as in Table 13.

K-Value	1	3	5	10
P	0.321	0.319	0.3299	0.327
R	0.32	0.333	0.346	0.309
F1	0.32	0.326	**0.338**	0.317
MAE	0.9	0.828	0.812	**0.802**
BLEU score	**22.04**	21.05	20.98	20.53

Table 15 Results of KNN classifier after applying PCA on training data

In Table 16 we repeat the above experiment shown in Table 15, but this time we classify between two decoders instead of three. This table also shows lowest MAE with lowest BLEU score. The results shown in table 14 coincide with those in table 11

K-Value	1	3
P	0.46	0.208
R	0.57	0.474
F1	**0.513**	0.289
MAE	1.55	**1.52**
BLEU score	**22.3**	21.07

Table 16 Results of KNN classifier after applying PCA on training data

4.3.2.2 Neural Networks

Objective: Classify English sentences based on their structure to be translated by 3 or 2 decoders. We apply PCA on training data with extra POS tags to minimize attributes. We try to configure the neural network to reach the highest BLEU score. This time we are using accuracy instead of the mean absolute error because it's calculated by deeplearning4j.

Training Set Used: Development set created from ISI corpus.

Test Set Used: Test set created from the ISI corpus.

Method:

1- Use the training data that produced the highest BLEU Score in KNN. The training data after applying PCA on it.
2- Create a Neural Network using DeepLearning4j platform on the training data.
3- Start Tuning Neural network by changing different variables till we reach highest possible BLEU score.
4- Run NN on test data.
5- Calculate BLEU score.

Below is the list of settings used to tune the neural network.

Setting 1:

- Number of Iterations = 50, learning Rate = 0.001
- Training data after applying PCA
- 2-Hidden Layer NN the first layer contains 20 nodes and the second contains 5 nodes.
- Output: 3 Classes

Setting 2:

- Number of Iterations = 50, learning Rate = 0.001
- Training data after applying PCA
- 1-Hidden Layer NN with 17 nodes
- Output: 3 Classes

Setting 3:

- Number of Iterations = 50, learning Rate = 0.001
- Training data with selected attributes after applying best first algorithm using Weka.

- 1-Hidden Layer NN with 17 nodes
- Output: 2 Classes

Setting 4:

- Number of Iterations = 50, learning Rate = 0.001
- Training data after applying PCA
- 2-Hidden Layer NN the first with 20 nodes and the second with 15 nodes
- Output: 2 Classes

Setting 5:

- Number of Iterations = 50, learning Rate = 0.001
- Training data after applying PCA
- 2-Hidden Layer NN the first with 20 nodes and the second with 20 nodes
- Output: 2 Classes

Setting 6:

- Number of Iterations = 50, learning Rate = 0.001
- Training data after applying PCA
- 2-Hidden Layer NN the first with 25 nodes and the second with 20 nodes
- Output: 2 Classes

In Table 17, we show the results of neural network classifier, also setting 5 has the highest BLEU score. But setting 4 has highest accuracy. The results show that the best translation results occurred with the best recall and accuracy scores of the classifier

	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6
P	0.2379	0.488	0.5073	0.4878	0.7381	0.4805
R	0.333	0.3335	0.5002	0.4991	0.5005	0.4956
F1	0.2775	0.3962	0.5037	0.4934	**0.5965**	0.488
Accuracy	0.4755	0.476	0.4765	0.476	**0.4765**	0.4745
BLEU	23.102	23.105	23.086	23.082	**23.105**	22.96

Table 17 Neural Network Results

4.3.2.3 Discussion

Even though the graphs in sections 4.2.1 and 4.2.2 suggest that we should combine decoders to create a stronger decoder that task doesn't look easy using the suggested feature vector. The increase in BLEU score is very low compared to the maximum BLEU score calculated by combining decoders which is **25.4**.

It is should be mentioned that the training set we are using ~74K is relatively small when compared to the dimensionality of the input (80). Minimizing features applied using best first algorithm decreases it to (23) and PCA decreases it to (69), enhances results a little bit. In the case of KNN combined with best first we got **22.68** and in the case of KNN with PCA we got **22.04**. In the case of NN combined with the best first we get **23.086** and in the case of NN combined with PCA we got **23.105**.

A major difference between KNN and NN when running this example is that sometimes the neural network fails to differentiate between classes and end up choosing one class for all test set sentences. It's obvious that when the accuracy of NN decrease the BLEU score decrease. In setting 1 & setting 2 NN failed to differentiate between all 3 classes it was never able to detect sentences that should be translated by syntax-based Chiang 2005 SMT. It either picks phrase-based or syntax-based with GHKM SMT but never syntax-based Chiang 2005 SMT.

In the case of the K-NN algorithm (Blue score **22.68**), it is worth noting that the classifier out-performed all the syntax based decoder created in the experiment shown in section 4.1.3 (Blue score **21.05**). But this was not enough to better than our chosen baseline system. Decreasing the number of attributes has little effect on the output of the classifier.

In the case of NN, decreasing the number of nodes, decreases the number of unknowns needed to make the NN classify between different classes this can be seen in setting 5 & setting 6. This has a little effect on BLEU score.

When we compare post-processing decoders and classifiers approach, we can see that in section 4.2.1 the highest BLEU score was **23.1** for enhancing a phrase-based translation using tree-to-string syntax-based SMT, a close result was achieved by NN in setting 5 **23.105**. When we enhanced a syntax-based translation using a tree-to-string syntax-based SMT we got **21.09**, which was outperformed by both KNN (**22.68**) and NN (**23.105**).

4.4 Comparison to Google Translate

Objective: We ran our test sets on Google translate, to check the quality of our translation in comparison to other well-known translation tools.

Test Set: ISI test set & UN test set.

Method:

1- Create a java application that uses Google translate API to translate ISI test set & UN test set from English to Arabic.
2- Calculate BLEU score using Stanford Phrasal.

Table 18 shows the results of translation of ISI test set and UN test set using Google translate. As shown in the table our baseline system outperformed Google translate on ISI test set while in the case of UN test set Google translate did better than all of our decoders and our combined decoders created in section 4.3.1.

UN corpus is a free corpus available to use by anyone, so it is probably already part of the training data in Google Neural Machine Translators, while ISI corpus is not free and might not be used as part of the training set for Google translator.

Test Set	BLEU Score	Our Highest BLEU Score
ISI Test Set	17.5	**23.1**
UN Test Set	**35.5**	26.59

Table 18 Google Translate BLEU Score

Chapter 5 Conclusion and Future Work

Machine translation is a very important topic in natural language processing. There is ongoing quest to enhance translation between different language pairs. There have been different approaches to enhance machine translation as described in chapter 2. Machine translation has many issues some depend on the language pairs we are translating, and some are caused by the lack of training data to create the decoders. Morphologically complex languages like Arabic pose a difficulty during translation. Arabic as a language is very complex and rich language. Sometimes two words in Arabic could have same spelling but different meanings. Adding clitics to Arabic words as prefixes or suffixes could change tense or change a single word to a sentence. We chose to translate from English to Arabic because most of the recent research is directed in the other direction from Arabic to English.

In this thesis we tried to enhance translation from English to Arabic by trying two different approaches; post-processing decoder and creating a multi-decoder by combining different kinds of translation models. The three research objectives addressed in the theses are:

1. Investigate different state of the art approaches to build a baseline system that produces the best results for English/Arabic translation using a small corpus.
2. Explore the possibility of applying a secondary machine translation system (Arabic to Arabic) for enhancing the translation quality of a test set of English sentences extracted from the same corpus or another corpus
3. Explore the possibility of combining different decoders to create a stronger decoder which produces the best output.

To cover these objectives, we had to answer 2 research questions related to the first objective are:

- What is the impact of data preprocessing on translation quality?
- What is the impact of using different types of SMTs on quality of translation?

The two research questions related to the second objective are:

- Could post processing using another translation model built by an Arabic/Arabic corpus generated from the development data set enhance the translation quality?
- Can the translation of sentences from different datasets not extracted from the corpus used in training be enhanced?

One research question was related to our third objective:

- Can the combination of different decoders generate a stronger one that outputs the best result produced by each decoder separately?

Our first objective was achieved by creating our baseline system. We then show that phrase-based was the best translation decoder with BLEU score 23.07, followed by syntax-based string-to-tree GHKM decoder with BLEU score 21.05. Even though we showed in Table 5 that sometimes syntax decoders could provide a semantically sane translation the BLEU score only cares about the words in the expected sentence and their order. So, accordingly the phrase-based decoder has the best translation.

We then explored using a post-processing decoder to enhance machine translation. We tried six combinations based on syntax-based & phrase-based. The best two were; phrase-based SMT followed by syntax-based tree-to-string SMT with BLEU score of 23.1, and Syntax-based SMT string-to-tree GHKM followed by syntax-based tree-to-string SMT with BLEU score of 21.09.

In this experiment we also tried to tackle the problem of translating a test set created from a corpus that is not related to the original corpus used in training the baseline decoder and the three post-processing decoders. Initially, the phrase-based scored 16.05 & syntax-based scored 15.89. We tried the previous six combinations again on this test set extracted from the UN corpus. The best was syntax-based SMT string-to-tree GHKM followed by phrase-based SMT combination with BLEU score of 26.59 followed by phrase-based SMT followed by syntax-based tree-to-string SMT with BLEU score of 23.51. In this experiment we managed to increase the BLEU score by 8~10 points. The post-processing decoder is better in the case of the UN corpus because it adds the basic information missing needed to translate the corpus by the baseline decoder.

Then we compared the percentage of test set sentences each decoder translates better than all the other decoders. We did this on two test sets the ISI test set and a test set from the UN corpus. In the case of the ISI test set, we found that no certain decoder is better than all other decoders, the best was phrase based decoder and it only covered around 35%, the second decoder was the syntax-based with GHKM extraction algorithm and it covered around 21% & the third was the syntax-based with Chiang extraction algorithm and it covered around 17%. In the case of the UN test set, we found that the test set is distributed between 5 decoders. The best decoder is syntax-based GHKM followed by a post-processing phrase-based decoder and it covered 39% and the second decoder is syntax-based GHKM followed

by tree-to-string syntax-based post-processing decoder that covered 21% and the third decoder is phrase-based decoder that covered 17%. Then we calculate the BLEU score that we can get from combining these decoders. The BLEU score for ISI test set is 25.4 and the BLEU score for UN test set is 30.5.

The results of this analysis lead to the idea of combining these decoders to create one stronger decoder. The basic idea was to classify English sentences based on their structure to choose the best decoder to translate the sentence. The input of the classifier is the POS tag of the sentence. We change the POS tag of the sentences to integer vectors and use it as input for the classifier.

The training data for the classifier consisted of POS tags as input and the identifier of the best decoder that could translate this sentence. We tried two types of classifiers; one we created using K-NN algorithm and the other was a Neural Network. The highest BLEU score calculated when using K-NN algorithm was 22.86 and the highest BLEU score calculated when using Neural Network was 23.105. This method didn't yield very high BLEU score. One of the problems of this approach was the classifier's failure to produce high accuracy. The highest F1 score is 0.596 by Neural Network with accuracy of 0.476. Classifier training should be enhanced by enhancing quality of training data. Another problem is size of the feature vector in comparison with respect to the size of the training data. Algorithms to minimize the feature vector should be used to pick the most important parts of the feature vector that could help in classifying the English sentences to be translated. One point worth mentioning, we have already reach 90% of the maximum BLEU score that could be reached by the decoders for this test case.

In the end, we wanted to compare our results with a well-known translator to know how well we are doing in comparison to these translators. We chose Google translate to compare our results with. In the case of ISI test set Google translate got 17.5 BLEU score while we got 23.1 using a post-processing decoder and a Neural Network and in the case of UN test set Google translate got 35.5 BLEU score while we got 26.59 using a post-processing decoder. Our baseline decoder did better than Google translate and in the case of ISI test set and Google translate did better than when applied on the UN test data set where the best score we got for UN test data set was 26.59 BLEU score.

Our contributions in this thesis could be observed in the following:

- Devise a methodology for building the best translation system using available tools.
- Study the impact of preprocessing and translation models on the quality of translation.
- Build a baseline system to study its improvement using new approaches.
- Propose the idea of post-processing decoder to translate Arabic to better Arabic which was very successful when applied on test data set taken form a different corpus and build the post-processing decoder from the development data set of this different corpus.
- Create a multi-decoder from a set of decoders and build a classifier that could choose the decoder that produces the best result for each sentence. This approach need more investigation to enhance the classifier accuracy.

For the future work, it is recommended to try a different feature vector that could represent a sentence in the source language. One of the ideas is to use syntactic structure to be added to the feature vector to enhance the classifier accuracy. Another idea could be segmenting the sentences to create smaller sentences to be used to train the classifier. Another idea would be fine tuning feature selection algorithm from the POS tags to create a new feature vector.

Also for post-processing decoder, when training a decoder on a data set and testing it on a data set extracted from a different corpus, the results aren't satisfactory. The post-processing decoder proved to work well when used on UN test set. It helped enhancing the translation quality for the UN test set. More testing on different data sets should be done to confirm if this approach could be used to generalize translating corpora from a different context.

Using deep learning would be also a good idea to try and compare with the results produces in this thesis.

References

[1] J. Hutchins, “Two precursors of machine translation: Artsrouni and Trojanskij,” *International Journal of Translation*, vol. 16, no. 1, pp. 11–31, 2004.

[2] J. Hutchins, “Warren Weaver and the Launching of MT,” *Early Years in Machine Translation, ed. W. J0hn Hutchins. Amsterdam: John Benjamins. 20o0*, pp. 17–20, 2000.

[3] J. Hutchins, “ALPAC: the (in) famous report,” *Readings in machine translation*, vol. 14, pp. 131–135, 2003.

[4] J. Hutchins, “Machine translation: History and general principles,” *The encyclopedia of languages and linguistics*, vol. 5, pp. 2322–2332, 1994.

[5] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311–318.

[6] W. J. Hutchins, “History of research and applications,” *Routledge Encyclopedia of Translation Technology*, p. 120, 2014.

[7] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, “The Mathematics of Statistical Machine Translation: Parameter Estimation,” *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, Jun. 1993.

[8] N. Kalchbrenner and P. Blunsom, “Recurrent Continuous Translation Models.,” in *EMNLP*, 2013, vol. 3, p. 413.

[9] A. E. Kholy and N. Habash, “Techniques for Arabic Morphological Detokenization and Orthographic Denormalization,” 2011.

[10] A. Bisazza and M. Federico, “Chunk-based Verb Reordering in VSO Sentences for Arabic-English Statistical Machine Translation,” in *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, Stroudsburg, PA, USA, 2010, pp. 235–243.

[11] M. Diab, K. Hacioglu, and D. Jurafsky, “Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks,” in Proceedings of HLT-NAACL 2004: Short Papers, Stroudsburg, PA, USA, 2004, pp. 149–152.

[12] M. Carpuat, Y. Marton, and N. Habash, “Improving Arabic-to-English statistical machine translation by reordering post-verbal subjects for alignment,” in *Proceedings of the ACL 2010 Conference Short Papers*, 2010, pp. 178–183.

[13] I. Badr, R. Zbib, and J. Glass, “Syntactic Phrase Reordering for English-to-Arabic Statistical Machine Translation,” in Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 86–93.

[14] N. Habash and O. Rambow, “Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop,” in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 2005, pp. 573–580.

[15] I. Badr, R. Zbib, and J. Glass, “Segmentation for English-to-Arabic statistical machine translation,” in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, 2008, pp. 153–156.

[16] A. El Kholy and N. Habash, “Orthographic and morphological processing for English–Arabic statistical machine translation,” Machine Translation, vol. 26, no. 1–2, pp. 25–45, Mar. 2012.

[17] N. Habash and F. Sadat, “Arabic preprocessing schemes for statistical machine translation,” in Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, 2006, pp. 49–52.

[18] M. Salameh, C. Cherry, and G. Kondrak, “What matters most in morphologically segmented SMT models,” *Syntax, Semantics and Structure in Statistical Translation*, p. 65, 2015.

[19] M. Salameh, C. Cherry, and G. Kondrak, “Lattice Desegmentation for Statistical Machine Translation.,” in *ACL (1)*, 2014, pp. 100–110.

[20] A. E. Kholy and N. Habash, “Techniques for Arabic Morphological Detokenization and Orthographic Denormalization,” 2011.

[21] M. Salameh, C. Cherry, and G. Kondrak, “Reversing Morphological Tokenization in English-to-Arabic SMT.,” in *HLT-NAACL*, 2013, pp. 47–53.

[22] S. Jiampojamarn, G. Kondrak, and T. Sherif, *Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion*. 2007.

[23] A. Clifton and A. 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-Volume 1, 2011, pp. 32–42.

[24] I. T. Khemakhem and S. Jamoussi, “Integrating morpho-syntactic features in English-Arabic statistical machine translation,” *ACL 2013*, p. 74, 2013.

[25] N. C. Kammoun, L. H. Belguith, and A. B. Hamadou, “The MORPH2 new version: A robust morphological analyzer for Arabic texts,” in *JADT 2010: 10th International Conference on Statistical Analysis of Textual Data*, 2010.

[26] A. Stolcke and others, “SRILM-an extensible language modeling toolkit.” in *Interspeech*, 2002, vol. 2002, p. 2002.

[27] I. T. Khemakhem, S. Jamoussi, and A. B. Hamadou, “Arabic-English Semantic Class Alignment to Improve Statistical Machine Translation,” RECENT ADVANCES IN, 2015, p. 663.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[29] A. V. Miceli-Barone and G. Attardi, “Non-projective dependency-based pre-reordering with recurrent neural network for machine translation,” in the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing, 2015.

[30] Á. Peris, M. Chinea-Rios, and F. Casacuberta, “Neural Networks Classifier for Data Selection in Statistical Machine Translation,” *CoRR*, vol. abs/1612.05555, 2016.

[31] A. Almahairi, K. Cho, N. Habash, and A. Courville, “First Result on Arabic Neural Machine Translation,” *arXiv:1606.02680 [cs]*, Jun. 2016.

[32] T. Xiao, J. Zhu, and T. Liu, “Bagging and Boosting statistical machine translation systems,” *Artificial Intelligence*, vol. 195, pp. 496–527, Feb. 2013.

[33] K. Duh, K. Kirchhoff, Beyond log-linear models: Boosted minimum error rate training for N-best Re-ranking, in: Proceedings Annual Meeting of the Association for Computational Linguistics (ACL), Columbus, OH, 2008, pp. 37–40.

[34] M. Federico, N. Bertoldi, and M. Cettolo, “IRSTLM: an open source toolkit for handling large scale language models.” in Interspeech, 2008, pp. 1618–1621.

[35] P. Koehn et al., “Moses: Open Source Toolkit for Statistical Machine Translation,” in Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Stroudsburg, PA, USA, 2007, pp. 177–180.

[36] D. C. Spence Green and C. D. Manning, “Phrasal: A toolkit for new directions in statistical machine translation,” in Proceedings of the Ninth Workshop on Statistical Machine Translation, 2014, pp. 114–121.

[37] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit.,” in ACL (System Demonstrations), 2014, pp. 55–60.

[38] P. Williams and P. Koehn, “Ghkm rule extraction and scope-3 parsing in moses,” in Proceedings of the Seventh Workshop on Statistical Machine Translation, 2012, pp. 388–394.

[39] D. Chiang, “A Hierarchical Phrase-based Model for Statistical Machine Translation,” in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 263–270.

[40] Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B, "The United Nations Parallel Corpus, Language Resources and Evaluation (LREC’16)," Portorož, 2016.

[41] “Weka Workbench Online Appendix for ‘ Data Mining: Practical Machine Learning Tools and Techniques ’ Morgan Kaufmann, Fourth Edition, 2016 - Semantic Scholar.” [Online]. Available: /paper/Weka-Workbench-Online-Appendix-for-Data-Mining-Pra-Frank-Hall/9062aee8c5eefa5a60f4d604a77b89f7f2a1a6b2. [Accessed: 13-Jun-2017].

[42] Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org

[43] D. C. Spence Green and C. D. Manning, “Phrasal: A toolkit for new directions in statistical machine translation,” in Proceedings of the Ninth Workshop on Statistical Machine Translation, 2014, pp. 114–121.

[44] M. Galley, M. Hopkins, K. Knight, and D. Marcu, “What’s in a translation rule,” DTIC Document, 2004.

APPENDIX A

Below is the conversion table of POS tags to corresponding number. This table was used in Section 3.6.1 and 3.6.2 when creating training and data sets.

POS tag	Number
CC	1
CD	2
DT	3
EX	4
FW	5
IN	6
JJ	7
JJR	8
JJS	9
LS	10
MD	11
NN	12
NNS	13
NNP	14
NNPS	15
PDT	16
POS	17
PRP	18
PRP$	19
RB	20
RBR	21
RBS	22
RP	23
SYM	24
TO	25
UH	26

VB	27
VBD	28
VBG	29
VBN	30
VBP	31
VBZ	32
WDT	33
WP	34
WP$	35
WRB	36
“.”	37
“,”	38
-RBR-	39
-LRB-	40
Otherwise	41

Table 19 POS tag to Number Conversion Table

APPENDIX B

The table below shows examples of translation enhancement in UN corpus using post-processing decoder.

Source Sentence	Output of Syntax-Based Decoder	Output of Phrase-Based Post-Processing Decoder	Expected Output
Approves applications by the following Government observer delegations for participation in meetings of the Standing Committee from October 2003 to October 2004	طلب من Approves الحكومة بعد مراقب من الوفود المشاركة فى اجتماعات اللجنة الدائمة من تشرين الاول / اكتوبر 2003 الى اكتوبر عام 2004	تقرّ الطلبات التي قدمتها وفود الحكومات التالية التي تحضر بصفة مراقب للمشاركة في اجتماعات اللجنة الدائمة من تشرين الأول / أكتوبر 2003 إلى تشرين الأول / أكتوبر 2004	توافق على الطلبات المقدمة من وفود الحكومات التالية المتمتعة بمركز مراقب للاشتراك في اجتماعات اللجنة الدائمة في الفترة من تشرين الأول / أكتوبر 2003 إلى تشرين الأول / أكتوبر 2004
To limit the requests for documents that need to be translated	من اجل الحد من المطالب من الوثائق التى تحتاج الى ترجمة	الحد مما تطلبه من وثائق تحتاج إلى الترجمة	الحد مما تطلبه من وثائق تحتاج إلى الترجمة
Human rights situations and reports of special rapporteurs and representatives : report of the Third Committee	من اوضاع حقوق الانسان وان التقارير الخاصة المقررون : وممثلين عن اللجنة الثالثة	حالات حقوق الإنسان والتقارير المقدمة من المقررين والممثلين الخاصين اللجنة الثالثة	حالات حقوق الإنسان والتقارير المقدمة من المقررين والممثلين الخاصين : تقرير اللجنة الثالثة
Budget for the United Nations Mission in the Sudan for the period from 1 July 2009 to 30 June 2010	في الميزانية الى بعثة الامم المتحدة فى السودان فى الفترة من 1 يوليو عام 2009 الى 30 حزيران / يونيو عام 2010	ميزانية في بعثة الأمم المتحدة في السودان للفترة من 1 تموز / يوليه 2009 إلى 30 حزيران / يونيه 2010	ميزانية بعثة الأمم المتحدة في السودان للفترة من 1 تموز / يوليه 2009 إلى 30 حزيران / يونيه 2010

Table 20 Sample of UN Enhanced Results