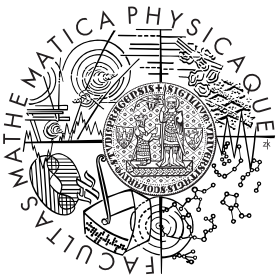
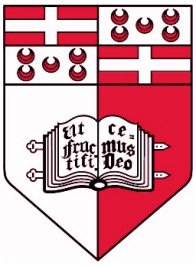
Charles University in Prague
Faculty of Mathematics and Physics
&
University of Malta
Faculty of Information and Communication Technology

Master's Thesis





Amir Kamran

Hybrid Machine Translation Approaches for Low-Resource Languages

Institute of Formal and Applied Linguistics (UFAL), Prague
Faculty of Information and Communication Technology, Malta

Supervisor:
Dr. Mike Rosner
and
Martin Popel

Study Program: LCT

Field: Computational Linguistics

October, 2011

I would like to express my gratitude to my supervisors Dr. Michael Rosner and Mgr. Martin Popel, who have been giving me knowledgeable guidance throughout this research, and thoroughly reviewed my work. I am also thankful to my friends, specially Bushra Jawaid, who has been discussing new ideas with me and always been a source of motivation.

Most of all, I want to specially mention my teacher and my mentor Dr. Tafseer Ahmed, who is the inspiration for me to join the field of Computational Linguistics. I would also like to thanks my parents for supporting my choice and always encourage me to do the best.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

Prague, August 2011. Amir Kamran

Title: Hybrid Machine Translation Approaches for Low-Resource Languages

Autor: Amir Kamran

Department: Intelligent Computer Systems
Faculty of Information and Communication Technology
University of Malta

Supervisor(s): Professor Michael Rosner & Mgr. Martin Popel

Abstract: In recent years, corpus based machine translation systems produce significant results for a number of language pairs. However, for low-resource languages like Urdu the purely statistical or purely example based methods are not performing well. On the other hand, the rule-based approaches require a huge amount of time and resources for the development of rules, which makes it difficult in most scenarios. Hybrid machine translation systems might be one of the solutions to overcome these problems, where we can combine the best of different approaches to achieve quality translation.

The goal of this dissertation is to explore different combinations of rule based and semi automatic preprocessing techniques for English-to-Urdu statistical machine translation and to evaluate their performance over the standard corpus based methods currently in use. This includes:

1. Insertion of artificial linguistic markers in English to improve the word alignment from English-to-Urdu.

2. Use of syntax-based word reordering rules to tackle the long distance reordering problem in statistical machine translation.

The novel element in the proposed work is to develop an algorithm to learn automatic reordering rules for English-to-Urdu statistical machine translation. Moreover, a comparison between hand written reordering rules with automatically learned rules will also be a part of this dissertation.

Keywords: machine translation, long-distance reordering, automatic rules extraction, language markers, low-resource languages, source-side preprocessing

Contents

List of Tables

List of Figures

1. Introduction

1.1 Motivation

Machine Translation (MT) is one of the earliest and most interesting problems of natural language processing. The aim is to bridge the language barrier by building machines capable of translating one human language into another. The past two decades focused on data-driven approaches to MT that produce significant improvement in translation quality.

From the point of view of data-driven approaches one can see MT as a statistical problem. Given a large parallel corpus, the underlying algorithms learn word level dictionaries and other phrase patterns to support the translation of unseen sentences. Typically and theoretically it has been the case that, the bigger the parallel corpora the better is the quality of translation. However, availability of large amounts of parallel text is the liberty of only few language pairs of the world and the rest of the languages, including Urdu [Baker et al., 2009, Aminzadeh and Shen, 2008, Lavie et al., 2003], are considered low-resource languages.

Availability of large parallel corpora is not the only obstacle in modern approaches to MT, there are other language dependent issues. In the beginning it was assumed that statistical methods are language independent, theoretically they might be, but the current computing power [Foster et al., 2003, Turchi et al., 2008] is not enough to build theoretical models. It has been shown in various studies [Koehn, 2007, Carpuat and Wu, 2007, Charniak et al., 2003, Ramanathan et al., 2008] that the incorporation of linguistic knowledge in Statistical Machine Translation (SMT) can improve the translation quality significantly. However, the availability of linguistic annotations for many languages is a limiting factor. For low-resource languages like Urdu the language analysis tools e.g. morphological analyzers, taggers, parsers, word-sense detectors etc. are very rare and also not easily available [Hussain, 2008].

Another major problem in SMT is dealing with significantly different word-order language pairs. It is difficult to find the correct word-order for the target language because of the computational complexity of investigating every possible target ordering. Many proposed solutions [Yamada and Knight, 2001, Chiang, 2005, Xiong et al., 2006, Zhang et al., 2007, Zollmann et al., 2008] have been effective but are resource intensive.

1.2 Thesis Objectives

The aim of this thesis is to compare different schemes of word reordering and factor mapping for English-to-Urdu statistical machine translation. The central idea is to use *only source side information* to achieve the goals as it is mentioned before that Urdu is a low-resource language. Particularly, this thesis concentrates on the following points:

- **Improving the source and target alignment using post-markers:** Urdu is a free-word order language and uses post-markers to identify cases, tenses and aspects. These post-markers are mostly separate words in the language. On the other hand, in English the identification of most of the aspects of the sentence is based on the order of words. In SMT systems word alignment plays an import role but due to the differences in marking system of Urdu and English, it is difficult to find the correct alignments of these markers. The idea proposed in this thesis is to introduce artificial post-markers in the English sentences to improve English-to-Urdu word alignment.

- **Learning automatic source-side reordering rules:** Most of the errors in English-to-Urdu statistical machine translation are due to long-distance reordering problems. An efficient way to overcome the reordering issue is to pre-process the source-side sentences to match the order of the target language. This can be achieved by writing the reordering rules manually by analyzing the source and the target languages [Jawaid and Zeman, 2011], but it is a resource intensive and language dependent task. Xia and McCord [2004] proposed a way to automatically learn rewriting patterns for French-English. More recently, Visweswariah et al. [2010] used probabilistic methods to learn and apply source-side reordering rules automatically for a number of language pairs.

  The main focus of this study will be to adopt efficient algorithms to learn the source-side reordering rules automatically for English-to-Urdu SMT to extend the work done by Jawaid [2010].

- **Experiments and Evaluation:** Any technique proposed must be deeply studied and evaluated. To make the results reliable we performed significance testing using a bootstrap resampling method [Zhang et al., 2004, Koehn, 2004]. Also, we compared the results of automatically learnt reordering rules with manually written rules.

1.3 Overview

The structure of this thesis is as follows:

Chapter 2 provides a detail theoretical background of Statistical Machine Translation, reordering techniques and MT evaluation methods, thus providing the necessary foundation to go through with the rest of the thesis.

Chapter 3 describes the experimental setup, the tools, algorithms and implementation details used in this research. It also provides a baseline for the main experiments.

Chapters 4 & 5 focus on the main goals of the thesis. Chapter 4 introduces the artificial marking scheme for English and Chapter 5 explains in detail the reordering technique and automatic acquisition of rules. The results are thoroughly compared and discussed.

Chapter 6 puts forward the conclusions of this research and suggestions for future research.

2. Background

This chapter provides the necessary theoretical background that will be needed for a proper understanding of the rest of the thesis.

2.1 History of Machine Translation

The idea of machine translation (MT) can be traced back to the seventeenth century, but it became realistically possible only in the middle of the twentieth century [Hutchins, 2007].[1] In the 1940s, with the invention of the first electronic computer, people like Warren Weaver and Alan Turing started talking about the use of computers for the task of MT.

The early systems were developed using the three basic approaches of MT: the direct approach, the rule-based transfer approach and the Interlingua approach. All these approaches require deep linguistic analysis, transformation rules written by language experts and various tools such as parsers, morphological analyzers, large bilingual dictionaries etc. to facilitate the translation process.

The Rule Based Machine Translation (RBMT) systems dominated until the appearance of corpus-based (aka data-driven) methods in the late 1980s. The new paradigm of MT basically introduces two new approaches namely Statistical Machine Translation (SMT) and Example Based Machine Translation (EBMT) [Nagao, 1984]. The most significant advantage of these new techniques over previous MT methods is their language independent nature. The early approaches require a huge amount of effort to introduce new language pairs, but with corpus-based approaches the only thing that is required is a bi-lingual corpus.

With the noteworthy success of SMT the major focus of many research groups shifted towards the corpus-based paradigm, and over the past decade SMT had seen many exciting developments. Starting from the word-based model [Brown et al., 1990] to more robust phrase-based models [Och, 2002, Koehn et al., 2003], SMT systems boosted the translation quality significantly, and have become the dominant method in MT research.

[1] For a complete reference of early history of Machine Translation see John Hutchins (1997 or revised edition 2005)

2.2 Statistical Machine Translation

The goal of the translation process in statistical machine translation is to find the most probable word sequence for the target language $e_1^I = e_1, ..., e_i, ..., e_I$ given a word sequence in source language. We can formulate the correspondence between the source and the target sentences using the Bayes decision rule [Brown et al., 1990]:

$$\begin{aligned}\hat{e}_1^I &= \arg\max_{e_1^I}\{Pr(e_1^I|f_1^J)\} \\ &= \arg\max_{e_1^I}\{Pr(e_1^I)Pr(f_1^J|e_1^I)\}\end{aligned}$$

Thus, the optimal translation is the maximization of the product of the probabilities given by the target language model $Pr(e_1^I)$ and the translation model $Pr(f_1^J|e_1^I)$.

2.3 Language Model

The famous American linguist and philosopher Chomsky said: "It must be recognized that the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term" [Chomsky, 1968, p.57]. This might be true in a certain sense that some sentences might have never been uttered before. However, malformed sentences are less likely to occur in a language than well-formed sentences and this is where the probability plays its role. The language model $Pr(e)$ reflects the fluency of the proposed target sentence. It is the probability distribution over the possible strings of a language, the higher the probability of a sentence the more it represents a natural language. Language models are usually smoothed n-gram models, typically conditioning on two (or more) previous words when predicting the probability of the current word.

$$\begin{aligned}P(w_1^n) &= P(w_1).P(w_2|w_1).P(w_3|w_1^2)...P(w_n|w_1^{n-1}) \\ &= \prod_{i=1}^{n} P(w_i|w_i^{i-1})\end{aligned}$$

Hence, the probability of word sequence is calculated by conditioning the next word on the history seen so far.

2.4 Translation Model

A translation model $Pr(f|e)$ is the set of possible translations for any target sentence. The translation model assigns probabilities to these translations, representing their relative correctness. The translation model must be learned from parallel texts where each sentence in one language is paired with a human translated sentence in another language.

2.5 Word Alignment

The heart of the machine translation approach is word alignment. It is the key to train a translation model. "Brown et al. [1990] defines an alignment between a pair of strings as an object indicating for each word in the target string, the word in the source string from which it arose. The alignment is defined as a function $a : \{i \rightarrow j\}$" [Birch, 2011]. See Figure 2.1, for an example of a parallel sentence and its word alignment.

English Do you understand English and Urdu ?

Urdu ؟ ہیں سمجھتے اردو اور انگریزی آپ کیا

Gloss[2] heñ səməjhəte ərədū or ənəgərezī ə p kyā

	ہیں	آپ	انگریزی	اور	اردو	سمجھتے	کیا	؟
Do	•							
you		•						
understand						•	•	
English			•					
and				•				
Urdu					•			
?								•

Figure 2.1: English-Urdu Parallel sentence word alignment

[2]All the gloss for Urdu text in this thesis is generated using a perl script written by Daniel Zeman. More details about the transliteration system can be found on the webpage (https://wiki.ufal.ms.mff.cuni.cz/user:zeman:transliteration-of-urdu-to-latin-script).

2.6 Reordering Techniques

One of the fundamental challenges in Statistical Machine Translation is to deal with different word-order languages. Especially for languages that have major word order differences such as SOV vs. SVO, word-order problems become the root cause of error maximization. Many reordering methods have been proposed in recent years to address this problem in different aspects.

One of the earliest approaches that is still used in current phrase-based systems is based on the distance modeling. Distance based distortion models [Koehn et al., 2003, Och and Ney, 2003] are a simple way of modeling phrase level reordering. It simply penalizes longer jumps more than shorter jumps independent of the source or target phrases in question. For example, if N words are skipped, a penalty of N will be paid regardless of which words are reordered. However, for very different word order languages longer jumps are required. In theory the distortion limit can be assigned a very large value so that all possible reordering can be allowed, yet in practice it is observed that too high distortion limit not only harms efficiency but also translation performance [Koehn et al., 2005]. Some later models generalize the distance based distortion model to include lexical dependencies on the source [Tillmann, 2004, Koehn et al., 2005, Al-Onaizan and Papineni, 2006] by applying different weights to different phrases.

Another approach puts syntactic analysis of the target language into both modeling and decoding. It has been shown that direct modeling of target language constituents' movement in either constituency trees [Yamada and Knight, 2001, Zollmann et al., 2008] or dependency trees [Quirk et al., 2005] can result in significant improvements in translation quality for various language pairs. A similar hierarchical phrase-based decoding [Chiang, 2005] has also shown promising results for translating Chinese to English. It allows for long range reordering without explicitly modeling syntax. Although shown improvements in translation quality, but these approaches typically combine machine translation decoding with chart parsing, therefore significantly increasing the decoding complexity.

A third approach is to reorder each source side sentence using a set of rules applied to a parse tree of the source sentence. The goal of using these rules is to make the word order of the source sentence more similar to the expected target sentence word order. In this approach the reordering is applied as a preprocessing step with an SMT system. The efficiency of these methods has been shown on various language pairs including: French to English [Xia and McCord, 2004], German to English [Collins et al., 2005], English to Chinese [Wang et al., 2007], English to Hindi [Ramanathan et al., 2008] and English to Urdu [Jawaid, 2010].

In recent work, Visweswariah et al. [2010] extends the above source side reordering approach by automatically learning the reordering rules, which makes this approach easily adoptable for new language pairs, but it shows some negative results for English to German. Similar work is done using dependency parsers [Xu et al., 2009] that has shown improved results especially for SOV languages.

2.7 Evaluation Techniques

As any other task in natural language processing, MT research is highly dependent on the robust evaluation. It is important to use standard system-independent evaluation techniques so that the effect of different models can be seen and compared. However, the flexibility of natural language is an obvious difficulty in setting a standard of evaluation.

The important question here is "What exactly is a good translation?" This is a very difficult question to answer because for an input sentence there can be several translations and all of them can be correct. Knight and Marcu [2005] have shown 11 distinct English translations by human translators, given the same Chinese sentence.

The general evaluation process is summarized by Leusch [2005], "Generally, evaluation and comparison of MT systems takes place by sending a fixed test set of source language sentences to the systems. Then, the MT systems translate these source sentences into the target language. The generated sentences, called candidate translations, are then assessed. Evaluation scores can be calculated on the level of whole test sets, as well as on the level of single test sentences. The former is the method of choice to compare different MT approaches, or to automatically adjust parameters. The latter is useful when the actual effects of a certain change in MT system parameters have to be analyzed" [Leusch, 2005, p. 2].

2.7.1 Human Evaluation

The most accurate way to do MT evaluation is to use human evaluators. However, this method is far more time consuming than automatic methods. It is difficult for human evaluators to evaluate a large sample of translated sentences.

The Workshop on Statistical Machine Translation (WMT) is one of the recent platforms where output of same test set from different machine translation systems is manually evaluated by the non-expert annotators through Amazon's

Mechanical Turk[3]. Each source sentence is presented with the reference translation and five candidate translations retrieved from five different MT systems. Annotators then rank those output translations starting from best to worst. Bojar et al. [2011] has discussed in detail the manual scoring scheme and the final MT systems grading criteria.

Research has shown that certain automatic evaluation methods have reasonable correlation with human evaluators, and thus they are usually used for the evaluation of large test sets [Zhang, 2006]. Automatic evaluation methods are usually preferred over human evaluation techniques due to extensive human labour needed for manual evaluation. We use BLEU and NIST metrics to evaluate the output translations obtained from all experiments that are conducted for this work.

2.7.2 BLEU

The BLEU (Bilingual Evaluation Understudy) metric [Papineni et al., 2001] evaluates machine translation quality by comparing the candidate translation of an MT system with correct translations called references. A test corpus thus required for this method, giving at least one manual translation for each test sentence.

BLEU was one of the first metrics to report high correlation with human judgments of quality and quickly it became the de facto standard for machine translation evaluation. It measures how well a machine translation overlaps with multiple human translations using n-gram co-occurrence statistics [Birch, 2011]. The BLEU score is evaluated by two factors, concerning the precision and the length of candidates, respectively. Precision refers to the percentage of correct n-grams in the candidate. In the simplest case, unigram ($n = 1$) precision equals to the number of words from the candidate that appear in the references divided by the total number of words in the candidate.

$$Precision = \frac{Number\ of\ words\ from\ the\ candidate\ that\ are\ found\ in\ the\ reference}{Total\ Number\ of\ words\ in\ the\ candidate}$$

The standard n-gram precision is sometimes inaccurate in measuring translation accuracy. Take the following candidate translation for example:

[3]The Amazon Mechanical Turk (MTurk) is a crowd-sourcing Internet marketplace that enables computer programmers (known as Requesters) to co-ordinate the use of human intelligence to perform tasks that computers are unable to do yet. See https://www.mturk.com for more details.

Candidate: a a a.
Reference: a good example.

In the above case, the standard unigram precision is $3/3 = 1$, but the candidate translation is inaccurate with duplicated words. Because of this problem, BLEU uses a modified n-gram precision measure, which consumes a word in the references when it is matched to a candidate word. The modified unigram precision of the above example is 1/3, for the word 'a' in the reference is consumed by the first 'a' in the candidate. Similar to unigrams, modified n-gram precision applies to bigrams, trigrams and so on so forth.

Apart from modified n-gram precision, a factor of candidate length known as Brevity Penalty is also included in the BLEU score. The main aim of brevity penalty is to penalize short candidates, because long candidates will be penalized by low modified n-gram precisions. The brevity penalty (BP) is computed by,

$$BP = \begin{cases} 1 & if c > r \\ e^{1-r/c} & if c \leq r \end{cases}$$

"where c is the length of the corpus of hypothesis translations, and r is the reference corpus length. In the case of multiple references, the reference corpus length is most commonly set to the length of the reference corpus which is closest to the hypothesis corpus. However, some researchers use the length of the shortest reference corpus and a further alternative is to use the average length of the reference sentences." [Birch, 2011]

The final BLEU score is computed by multiplying the brevity penalty with the geometric average of the modified n-gram precision, p_n calculated over n-grams up to length N, using positive weights w_n that sum up to one.

$$BLEU = BP. \exp(\sum_{n=1}^{N} w_n \log p_n)$$

It is better to use several independent reference translations (usually 4 if available), in our experiments we use only 1 reference translation per sentence. Experiments have shown that the BLEU metrics are generally consistent with human evaluators, and thus are useful indicators for the accuracy of machine translation.

2.7.3 NIST

The NIST (National Institute of Standards and Technology) metric [Doddington, 2002] was developed on the basis of the BLEU metrics. It focuses mainly on improving two problems of the BLEU score. First, the BLEU metric uses the geometric average of modified n-gram precisions. However, because current MT systems have not reached considerable fluency, the modified n-gram precision scores may become very small for long phrases (i.e. big n). Such small scores have a potential negative effect on the overall score, which is not desired. To solve this problem, the NIST score uses the arithmetic average instead of geometric average. In this way, all modified n-gram precisions make zero or positive contribution to the overall score [Zhang, 2006].

Second and most important, the BLEU metric weighs all n-grams equally in the modified n-gram precision score. The NIST metric gives all n-grams an information weight with respect to the reference sentences, so that rarer and more informative sequences present in the translation will contribute more to the final score than sequences that are more common, and thus less informative. For example, the bigram "machine translation" is considered more useful for the evaluation than the bigram "of the". The NIST metric information weight is computed by:

$$Info(w_1...w_2) = \log_2 \left(\frac{\textit{the number of occurrences of } w_1...w_{n-1}}{\textit{the number of occurrences of } w_1...w_n} \right)$$

Besides the above two differences, the NIST score also uses a special brevity penalty score. In equation form, it can be written as:

$$BP = \exp\left(\beta \log^2(min(\frac{L_{sys}}{\bar{L}_{ref}}, 1))\right)$$

where $\bar{L}_{ref}$ is the average number of words in the references, L_{sys} is the number of words in the candidate, and β is chosen to make $BP = 0.5$ when the number of words in the candidate is 2/3 of the average number of words in the references. Finally, the NIST score for MT evaluation can be written as:

$$Score = BP. \sum_{n=1}^{N} \left(\frac{\sum_{w_1...w_n \in Matched} Info(w_1...w_n)}{\sum_{w_1...w_n \in Candidate}(1)} \right)$$

The score is calculated over n-grams up to length N.

2.7.4 Significance Testing

Both BLEU/NIST metrics require a test suite to evaluate the MT systems. However, as mentioned in Section 2.7.2, several independent reference translations are required for correct evaluation thus building a test suite is not cheap. In fact, since the introduction of BLEU, the MT community has had only a few test suites with multiple human references. The BLEU/NIST scores are usually based on one test suite. Thus, when we have a BLEU/NIST for one MT system, we have to ask ourselves a question: "Is this score precise?" [Zhang et al., 2004]

"In statistical tests, we often use confidence interval to measure the precision of an estimated value. The interval represents the range of values, consistent with the data, which is believed to encompass the "true" value with high probability (usually 95%). The confidence interval is expressed in the same units as the estimate. Wider intervals indicate lower precision; narrow intervals, greater precision. The estimated range is calculated from a given set of sample data." [Zhang et al., 2004]

Since building test suites is expensive, it is not practical to create a set of testing suites to generate a set of sample BLEU/NIST scores. Instead, we use the well-known bootstrapping technique [Koehn, 2004, Zhang et al., 2004] to measure the confidence interval for BLEU/NIST.

Koehn [2004] describes a non-analytical method to compute confidence intervals for BLEU/NIST metrics, called bootstrap resampling. The intuition behind bootstrap resampling goes as follows: Assume that we have a large set of potential sentences but we can only test translation performance on a small test set of say n sentences. Given a test set, we can compute a BLEU score. Then, we draw a second test set of n sentences, and compute its BLEU score. If we repeat this process m number of times, we will get m test sets and m corresponding BLEU scores. After sorting all the BLEU scores, if we drop the top 2.5% and bottom 2.5% of scores, we have the remaining 95% of scores in an interval [a, b]. The law of large numbers dictates, that with an increasingly large m, the interval approaches the 95% confidence interval for scores of test sets of size n.

"Of course, having to translate and score sets of n sentences repeatedly, does not save anything in terms of computational translation cost and the need for a large set of potential sentences. We therefore, take the following leap: Instead of selecting the n sentences in each test set from an infinite set of test sentences, we draw them from the same set of n sentences with replacement." [Koehn, 2004]

The above described method gives us a confidence interval to estimate bounds of the true performance level of a system. However, to compare two different

systems we use a different method called *paired bootstrap resampling*. In this method, we use bootstrap resampling method on both systems. Given a small collection of translated sentences, we repeatedly (say, 1000 times) create new virtual test sets by drawing sentences with replacement from the collection. For each set, we compute the evaluation metric score for both systems. We note, which system performs better. If, say, one system outperforms the other system 95% of the time, we draw the conclusion that it is better with 95% statistical significance.

3. Setup of Experiments

This chapter describes the details of the experiments, including the software tools, the training and testing corpora, and the typical statistical machine translation process that is used by all the experiments of this thesis. In the end of this chapter, we create a baseline translation model that gives us the standard evaluation score for the rest of our experiments.

The task of statistical machine translation can be divided into several subtasks. Each task is performed by a different software component. The common tasks are collection of corpus, pre-processing, training, tuning, testing and post-processing. The next few sections of this chapter discuss the main tasks we perform in all of the experiments.

3.1 Corpora

We perform our experiments on the English-Urdu language pair. We used two different sets of parallel corpora: Emille and a small set of Penn Treebank sentences, translated in Urdu by CRULP[1]. Both of the corpora were modified and cleaned by Jawaid [2010] and we are using the modified versions. For the details about the preprocessing, normalization and issues in the corpora see Jawaid [2010, p. 18:33].

3.1.1 EMILLE Corpus

The EMILLE (Enabling Minority Language Engineering) corpus [Baker et al., 2002] is a 63 million words corpus of Indic languages, which is distributed by the European Language Resources Association (ELRA). It contains data from six different categories: consumer, education, housing, health, legal and social. This data is based on the information leaflets provided by the UK government and the various local authorities. Table 3.1 shows the statistics of the EMILLE corpus. Figure 3.1 shows a density plot of the lengths of the English and Urdu sentences, which highlights that the Urdu language uses more words. Section 4.3 discusses more differences in sentence lengths.

[1]Center for Research in Urdu Language Processing (www.crulp.org)

	English	Urdu
Sentence Pairs	8736	
Unique Sentences	8626	
Average Sentence Length	17.6	22.9
Words (number of tokens)	153519	200184
Vocabulary (classes)	9088	10381
Unigrams (tokens appear only once)	3627	4754

Table 3.1: Statistics of EMILLE English-Urdu parallel corpus

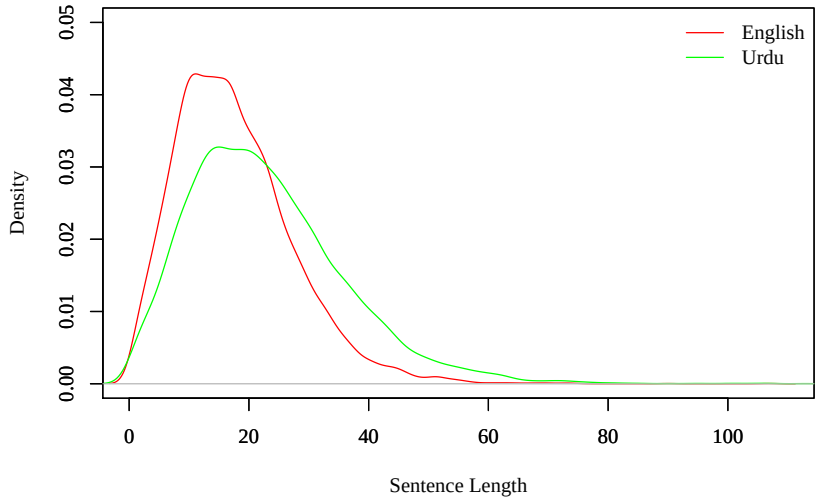


Figure 3.1: EMILLE Corpus sentence length analysis

3.1.2 Penn Treebank Corpus

Penn Treebank corpus is released through the Linguistic Data Consortium (LDC). The Parallel Penn-Urdu Treebank data is released by CRULP under the Creative Common License. The corpus is freely available online for research purposes. The Penn Treebank is a collection of the Wall Street Journal (WSJ), Brown corpus, Switchboard and ATIS. Unfortunately, the Urdu translation is only available for a small set of WSJ section that we used in our research. Table 3.2 shows the statistics of the Penn corpus and Figure 3.2 shows the comparison of sentence lengths for Penn corpus.

	English	Urdu
Sentence Pairs	6215	
Unique Sentences	6203	
Average Sentence Length	25.9	29.8
Words (number of tokens)	161154	185357
Vocabulary (classes)	15294	13029
Unigrams (tokens appear only once)	7446	6155

Table 3.2: Statistics of Penn Treebank English-Urdu parallel corpus

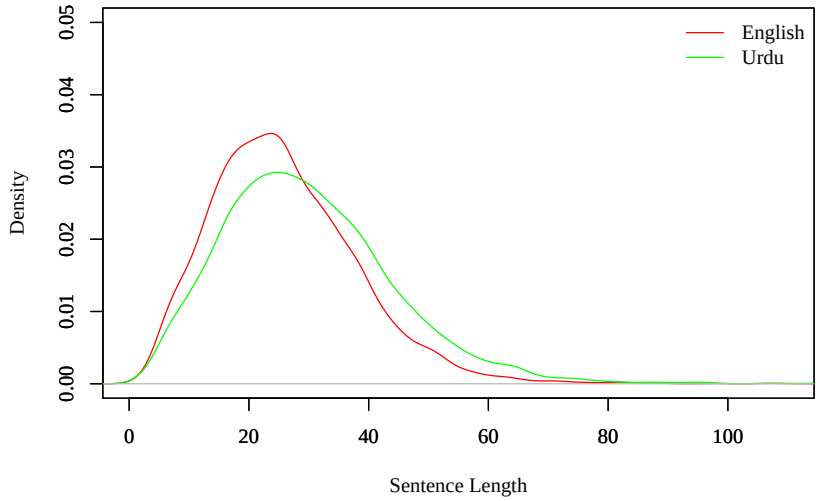


Figure 3.2: PENN Corpus sentence length analysis

3.1.3 EMILLE+Penn Corpus

We have also created a third corpus that is in fact a merge of both the above described corpora. The idea is to test whether more sentences from different domains increase or decrease the efficiency of a translation model. We named this new corpus EMILLE+Penn. Table 3.3 shows the statistics of the EMILLE+Penn corpus. The counts suggest that by combining both the corpora we have better vocabulary coverage.

3.2 Data Distribution

For creating a statistical machine translation system, we need three distinct datasets; training set (seen data) to build the model, test set (unseen data) to

	English	Urdu
Sentence Pairs	14951	
Unique Sentences	14829	
Average Sentence Length	21.0	25.8
Words (number of tokens)	314673	385541
Vocabulary (classes)	19403	19309
Unigrams (tokens appear only once)	8605	9114

Table 3.3: Statistics of EMILLE+PENN English-Urdu parallel corpus

measure the performance of the model and a development set to tune the parameters of the model. It is important to select these datasets in such a manner that each set should be truly representative of the corpus. As we do not have very large corpora, it is assumed that chunks of data following uniform distribution in terms of vocabulary coverage. We split the data in the following manner: the training set is taken from the beginning of the corpora, followed by taking the sentences for the development set and the rest of the sentences are allocated for testing. The summary of the data distribution is given in the Table 3.4 and Table 3.5.

Set		**English**	**Urdu**
Train	Sentence Pair	8000	
	Sentence Length	17.6	22.9
	Words (number of tokens)	141136	183021
	Vocabulary	8799	10003
	Unigrams	3559	4599
Development	Sentence Pair	376	
	Words (number of tokens)	6071	8322
	Vocabulary	1129	1291
	Unigrams	552	638
	Vocabulary Coverage	89%	83%
Test	Sentence Pair	360	
	Words (number of tokens)	6312	8841
	Vocabulary	1322	1388
	Unigrams	664	624
	Vocabulary Coverage	86%	86%

Table 3.4: Data distribution of EMILLE English-Urdu parallel corpus

Set		English	Urdu
Train	Sentence Pair	5700	
	Sentence Length	26.0	29.7
	Words (number of tokens)	147994	169207
	Vocabulary	14592	12509
	Unigrams	7126	5956
Development	Sentence Pair	315	
	Words (number of tokens)	8154	9934
	Vocabulary	2272	2096
	Unigrams	1465	1169
	Vocabulary Coverage	79%	82%
Test	Sentence Pair	200	
	Words (number of tokens)	5006	6216
	Vocabulary	1590	1425
	Unigrams	1069	771
	Vocabulary Coverage	84%	88%

Table 3.5: Data distribution of Penn Treebank English-Urdu parallel corpus

The PENN+EMILLE corpus as described in the previous section is a concatenation of both Penn and EMILLE corpora so we cannot use exactly the same strategy to split the data. The PENN+EMILLE corpus splitting is done in the following manner: the training set is a concatenation of the training sets of Penn and EMILLE corpora, similarly the development and test sets are the concatenation of the respective Penn and EMILLE sets. The summary of the data distribution of the PENN+EMILLE corpus is given in Table 3.6

3.3 Language Model

There are various software packages available to build Statistical Language Model. For example, the SRI Language Modeling toolkit (SRILM) [Stolcke, 2002], or IRST Language Modeling toolkit (IRSTLM) [Federico et al., 2008].

In this thesis, we use SRILM toolkit. SRILM is a freely available collection of C++ libraries, executable programs, and helper scripts. The main purpose of the toolkit is to support language model estimation and evaluation. Estimation means the creation of a model from training data; evaluation means computing the probability of a test corpus [Stolcke, 2002].

To estimate the language model, we use the ngram-count utility provided

Set		**English**	**Urdu**
Train	Sentence Pair	13700	
	Sentence Length	21.1	25.7
	Words (number of tokens)	289130	352228
	Vocabulary	18658	18571
	Unigrams	8320	8811
Development	Sentence Pair	691	
	Words (number of tokens)	14225	18256
	Vocabulary	2962	2909
	Unigrams	1691	1523
	Vocabulary Coverage	84%	83%
Test	Sentence Pair	560	
	Words (number of tokens)	11318	15057
	Vocabulary	2534	2401
	Unigrams	1437	1157
	Vocabulary Coverage	87%	88%

Table 3.6: Data distribution of PENN+EMILLE English-Urdu parallel corpus

with the SRILM toolkit. Also, SRILM implements various smoothing algorithms such as Good-Turing, Absolute Discounting, Witten-Bell and modified Kneser-Ney (KN). We built our model using KN-discount smoothing. By default SRILM removes the unknown words in calculating the ngram-counts; we build the open vocabulary LM i.e. one that contains the unknown-word tokens as a regular word. SRILM can induce a language model of any order; in this study we have chosen to use the trigram language model unless stated otherwise.

3.4 MOSES

MOSES [Koehn et al., 2007b] is a state-of-the-art open-source toolkit for statistical machine translation. According to [Koehn et al., 2007b], "The toolkit is a complete out-of-the-box translation system for academic research. It consists of all the components needed to preprocess data, train the language models and the translation models. It also contains tools for tuning these models using minimum error rate training (MERT) [Och, 2003]".

Islam [2009] describes MOSES as, "MOSES is an extended phrase-based MT system with factors [Koehn, 2007] and confusion network decoding built in. We can integrate any language feature as a factor during training. In factored model

the surface form may be augmented with a number of factors such as POS tag or lemma. The confusion network allows the translation of ambiguous sentences. This enables tighter integration of speech recognition and machine translation instead of passing along the one best output of the recognizer. MOSES has an efficient data structure that allows memory-intensive translation model and language model by exploiting larger data resources with limited hardware. It implements an efficient representation of phrase translation table using the prefix tree structure, which allows to load only the fraction of phrase table into the memory that is needed to translate the test sentence. MOSES uses the beam-search algorithm that quickly finds the highest probability translation among the exponential number of choices."

3.5 Translation Steps of MOSES

The training process takes place in nine steps, all of them executed by the script "train-model.perl". The nine steps are as follows:

1. Data preparation
2. Run Giza++
3. Align Words
4. Get lexical translation table
5. Phrase extraction
6. Score phrases
7. Build lexicalized reordering model
8. Build generation models
9. Create configuration file

In the following sections, we briefly discuss the MOSES training steps and the external tools used by MOSES. The text in following sub-sections is mainly taken from the MOSES tutorial[2]. The settings used in our experiments is highlighted.

[2]http://www.statmt.org/moses/?n=FactoredTraining.HomePage

3.5.1 Data Preparation

In this step, the MOSES toolkit converts the provided parallel corpus into a format that is suitable for the GIZA++ toolkit. The parallel corpus is converted into a numbered format and two vocabulary files are generated. The vocabulary files contains words, integer word identifier and word count information. A sentence-aligned corpus file is also generated, which contains three lines for each sentence pair. First line contains the frequency of the sentence, the other two lines contains word ids of the source and the target sentences. GIZA++ also requires words to be placed into word classes. This is done automatically by calling the mkcls[3] program.

3.5.2 Giza++

GIZA++[4] is a statistical machine translation toolkit that is used to train IBM Models 1-5 [Brown et al., 1993] and an HMM word alignment model [Vogel et al., 1996]. In the MOSES training, GIZA++ is an initial step to establish word alignments. The word alignments are taken from the intersection of bidirectional runs of GIZA++, plus some additional alignment points from the union of the two runs.

3.5.3 Align Words

To establish word alignments based on the two GIZA++ alignments, a number of heuristics may be applied. The default heuristic grow-diag-final starts with the intersection of the two alignments and then adds additional alignment points. There are other possible alignment methods available such as: intersection, grow (only add block-neighboring points), grow-diag (without final step), union etc. In our experiments we used grow-diag-final-and heuristic. The grow-diag-final-and heuristic works via expanding the alignment by adding directly neighboring alignment points and alignment points in the diagonal neighbourhood. This step generates alignment files, the most important file is aligned.grow-diag-final-and

[3]mkcls is a tool to train word classes by using a maximum-likelihood-criterion. The resulting word classes are especially suited for language models or statistical translation models. The program mkcls was written by Franz Josef Och.

[4]GIZA++ is an extension of the program GIZA (part of the SMT toolkit EGYPT), which was developed by the Statistical Machine Translation team during the summer workshop in 1999 at the Center for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU). GIZA++ includes a lot of additional features. The extensions of GIZA++ were designed and written by Franz Josef Och.

that contains alignment information in the form of position of the source and target words.

3.5.4 Lexical translation table

Given the alignments, it is quite straight-forward to estimates a maximum likelihood lexical translation table. MOSES estimate the w(e|f) as well as the inverse w(f|e) word translation table. For example, here are the top translations for the word 'urdu' into Urdu in the file lex.f2e:

urdu اردو 0.5000000
urdu ردو 0.2500000
urdu اُردو 0.2500000

3.5.5 Phrase Extraction

In this step, all phrases are dumped into one big file "extract". The content of this file is in the following format: each line consist of source phrase, target phrase, and alignment points. An inverted alignment file "extract.inv" is also generated. By default, a lexicalized reordering model is also trained and stored in the file "extract.o".

3.5.6 Score Phrases

A translation table is created from the stored phrase translation pairs. To estimate the phrase translation probability $p(e|f)$ MOSES proceeds as follows: First, the extract file is sorted, then it collects counts and computes $p(e|f)$ for that source phrase f. To estimate $p(f|e)$, the inverted file is sorted, and then $p(f|e)$ is estimated for target phrase. Other phrase translation scores are also computed that are inverse lexical weighting, direct lexical weighting and phrase penalty.

3.5.7 Reordering model

By default, MOSES only includes a distance-based reordering model in the final configuration. Distance-based reordering model is discussed in Section 2.6. Additional conditional reordering models may be build. These are conditioned on specified factors, and learn different reordering probabilities for each phrase pair. Possible configurations are:

- msd vs. monotonicity. MSD models consider three different orientation types: monotone, swap, and discontinuous. Monotonicity models consider only monotone or non-monotone, in other words swap and discontinuous are lumped together.
- f vs. fe. The model may be conditioned on the foreign phrase (f), or on both the foreign phrase and English phrase (fe).
- unidirectional vs. bidirectional. For each phrase, the ordering of itself in respect to the previous is considered. For bidirectional models, also the ordering of the next phrase in respect to the current phrase is modeled.

The number of features that are created with a lexical reordering model depends on the type of the model. A msd model has three features, one each for the probability that the phrase is translated monotone, swapped, or discontinuous. A bidirectional model doubles the number of features - one for each direction. In all our experiments we uses the configuration msd-bidirectional-fe. The distance model is always included by MOSES.

These lexicalized reordering models are used only for local reordering and are unable to handle long-distance reordering. The long-distance reordering techniques are discussed in Section 2.6 and also some approaches are presented in this thesis in Chapter 5.

3.5.8 Generation Model

The generation model is built from the target side of the parallel corpus. By default, forward and backward probabilities are computed. For our experiments we use the default configuration of MOSES.

3.5.9 Configuration File

As a final step, a configuration file for the decoder is generated with all the correct paths for the generated model and a number of default parameter settings. This file is called moses.ini and it is stored in the folder “model”. This file is then used by the decoder for producing the translations. However, the configuration file generated by MOSES contains default weights for all parameters, which are of questionable quality. To obtain better weights, optimization is required on a development set. The optimization is discussed in the next section.

3.6 Optimization: MERT

MERT is a tool for minimum error rate training [Och, 2003], which is included in MOSES. It is an effective means to estimate the feature function weights of a linear model such that an automated evaluation criterion (such as BLEU) for measuring system performance can directly be optimized in training. The training procedure determines for each feature function its exact error surface on a given set of candidate translations. The feature function weights are then adjusted by traversing the error surface combined over all sentences and picking those values for which the resulting error count reaches a minimum.

MERT is a stochastic optimization algorithm that typically finds a different weight vector each time it runs. Foster and Kuhn [2009] have shown that, while the variance on the development set objective may be narrow, the held-out test set variance is typically much greater, but a secondary development set can be used to select a system that will have better generalization. However, we do not have large corpora and we cannot take out multiple held-out test sets for tuning. Therefore, we run MERT once only for each experiment.

3.7 Stanford Parser

Stanford Parser [Klein and Manning, 2009] is a Java implementation of probabilistic natural language parsers, both highly optimized PCFG (Probabilistic Context-Free Grammar) and lexicalized dependency parsers, and a lexicalized PCFG parser. The original version of this parser was mainly written by Dan Klein, with supporting code and linguistic grammar development by Christopher Manning. The lexicalized probabilistic parser implements a factored product model, with separate PCFG phrase structure and lexical dependency experts, whose preferences are combined by efficient exact inference, using an A* algorithm.

We use the Stanford parser in our experiments to generate syntactic trees for English in the reordering experiments, described in Chapter 5. The parser is also utilized to include dependency features in marker experiments, described in Chapter 4.

3.8 Significance Testing Tool

For significance testing, we use a bootstrapping tool[5] written in Perl. This tool implements the bootstrapping technique described by Koehn [2004] and Zhang et al. [2004] for BLEU metric. It also compares multiple machine translations systems using paired bootstrap resampling method [Koehn, 2004]. We discussed the significance testing in detail in Section 2.7.4.

3.9 Baseline Experiments

Our baseline is a plain single-factored translation model trained using MOSES without any preprocessing on the corpora. The parallel data is aligned using the Giza++ implementation of IBM Model 4. Alignments are extracted in both directions and then symmetrized using the grow-diag-final-and heuristic. A 3-gram target language model is used that is built from the training set itself. N-Best MERT is then used to tune the parameters of the model with n=10. Translation results were evaluated using BLEU and NIST metrics. The results are listed in Table 3.7.

Test Corpus	NIST	BLEU
EMILLE	4.8636	0.1889
Penn Treebank	5.5002	0.1789
PENN+EMILLE	5.4893	0.1908

Table 3.7: Results of baseline experiments

To get the significance level of the above results we calculate the BLEU scores for 1000 samples, which gives us the following confidence intervals: Table 3.8.

Test Corpus	Average BLEU	Lower limit	Upper limit
EMILLE	0.18903	0.17004	0.20700
Penn Treebank	0.17893	0.16350	0.19608
PENN+EMILLE	0.19089	0.17815	0.20339

Table 3.8: 95% confidence intervals for baseline results

[5]The significance testing tool is written and kindly provided by Martin Popel

The main reason of creating mix PENN+EMILLE corpus is to see the effect of an increase in vocabulary size from two different domains. The above results show that there is no significant improvement in our experiments. Although, the average BLEU score is higher for PENN+EMILLE corpus but the confidence intervals are overlapping with the other two corpora, which suggests that the improvement is not significant.

Table 3.9 shows sample output translations from the baseline system, including comparison of the sentences translated using individual corpora models with the mixed corpus (PENN+EMILLE) model. The second example is interesting; the translation of 'or' in the reference urdu sentence is 'or | اور', which is actually the translation of 'and'. However, the sense in which 'or' is used in the sentence both 'or' and 'and' can be a good translation. The baseline system (both corpora) translated 'or' to 'yā | یا', which is the correct translation of 'or' but the BLEU score was penalized because the translated word does not exist in the reference translation. This is one of the major problems of using a single reference with BLEU metric.

Example 1	
Source	you do not need to get any benefits
Reference	آپ کے لئے ضروری نہیں ہے کہ آپ کو کوئی بینیفٹ ملتا ہو ə p ke lae zərūrī nahīñ he kəh ə p ko koī benīfəṭ mələtā ho Matching n-grams: 1-grams: 5, 2-grams: 3, 3-grams: 2, 4-grams: 1
EMILLE Output	آپ کو کوئی بینیفٹ حاصل کرنے کی ضرورت نہیں ə p ko koī benīfəṭ ḥāṣəl kərəne kī zərūrət nahīñ Matching n-grams: 1-grams: 13, 2-grams: 11, 3-grams: 9, 4-grams: 8
PENN+EMILLE Output	آپ کو لئے ضروری نہیں ہے کہ آپ کو کوئی بینیفٹ ملتا ہو ə p ko lae zərūrī nahīñ he kəh ə p ko koī benīfəṭ mələtā ho Matching n-grams: 1-grams: 13, 2-grams: 11, 3-grams: 9, 4-grams: 8
Example 2	
Source	to get more information or other leaflets get in touch with your social security office.
Reference	مزید معلومات اور دوسرے لیف لٹ حاصل کرنے کے لئے اپنے سوشل məzīd mə'əlomāt or dosəre līf laṭ ḥāṣəl kərəne ke lae əpəne sošəl سیکورٹی آفس سے رابطہ کیجئے ۔ sīkwrəṭī ə fəs se rābəṭəh kījae . Matching n-grams: 1-grams: 16, 2-grams: 13, 3-grams: 11, 4-grams: 10
EMILLE Output	مزید معلومات یا دوسرے لیف لٹ حاصل کرنے کے لئے اپنے سوشل məzīd mə'əlomāt yā dosəre līf laṭ ḥāṣəl kərəne ke lae əpəne sošəl سیکورٹی آفس سے رابطہ کیجئے ۔ sīkwrəṭī ə fəs se rābəṭəh kījae . Matching n-grams: 1-grams: 16, 2-grams: 13, 3-grams: 11, 4-grams: 10
PENN+EMILLE Output	مزید معلومات یا دیگر لیفلیٹس حاصل کرنے کے لئے اپنے سوشل سیکورٹی məzīd mə'əlomāt yā dīgar līfəleṭəs ḥāṣəl kərəne ke lae əpəne sošəl sīkwrəṭī آفس سے رابطہ قائم کر سکتے ہیں ə fəs se rābəṭəh qāīm kər səkəte heñ Matching n-grams: 1-grams: 12, 2-grams: 8, 3-grams: 5, 4-grams: 3
Example 3	
Source	the lawsuit alleges that an inventory of pinkerton 's memorabilia disclosed that one of the watches had n't been forked over by american brands
Reference	مقدمہ دعوی کرتا ہے کہ پنکرٹنز کے یادگاروں کے ایک məqədəməh də'wī kərətā he kəh pənəkərəṭənəz ke yādəgāroñ ke ek فرد نے انکشاف کیا کہ گھڑیوں میں سے ایک ، fərəd ne ənəkəšāf kəyā kəh ghəṛīoñ meñ se ek , امریکی برانڈز کے زریعے واپس نہیں کی گئی تھی əmərīkī bərānəḍəz ke zəry'e wāpəs nahīñ kī gəeī thī
EMILLE Output	عدالتی مقدمہ خارج کرتے ہیں کہ کی ایک کا انکشاف کیا 'ədālətī məqədəməh xārəj kərəte heñ kəh kī ek kā ənəkəšāf kəyā کہ سے ایک کے طور پر گھڑیاں نہیں کی گئی تھی kəh se ek ke ṭwr pər ghəṛəyāñ nahīñ kī gəeī thī ، امریکن برانڈز کے ذریعے ۔ , əmərīkən bərānəḍəz ke żəry'e . Matching n-grams: 1-grams: 19, 2-grams: 8, 3-grams: 4, 4-grams: 1
PENN+EMILLE Output	عدالتی مقدمہ خارج کر دیا کہ کے ایک کا انکشاف کیا 'ədālətī məqədəməh xārəj kər dəyā kəh ke ek kā ənəkəšāf kəyā کہ گھڑیوں میں سے ایک کے طور پر نہیں کی گئی kəh ghəṛyoñ meñ se ek ke ṭwr pər nahīñ kī gəeī تھی ، امریکن برانڈز کے ذریعے ۔ thī , əmərīkən bərānəḍəz ke żəry'e . Matching n-grams: 1-grams: 21, 2-grams: 12, 3-grams: 8, 4-grams: 5

Table 3.9: Sample output of baseline system

4. Language Features and SMT

As discussed in section 2.6 reordering is one of the fundamental problems when dealing with different word-order languages. Applying various pre-processing techniques can effectively solve the problem of word ordering. However, there are other issues that also badly influence the translation quality particularly when translating from a fixed-word order language to a free-word order language. These issues are mostly related to the generation of markers and morphology, as free-word order languages require extra language features to identify different grammatical relations of a sentence.

This chapter will present the differences between English and Urdu case marking system. It will highlight some of the morphological differences specially related to verbs. There will be a discussion on the factored model [Koehn, 2007] approach, which provides a framework for incorporating lemmas, suffixes, POS tags, and any other linguistics factors in a log-linear model for phrase-based SMT. After that, we will present our approach for generating markers and morphology for English-to-Urdu statistical machine translation. We will compare the suggested approach with factored based model for English-to-Urdu with emphases on the fact that Urdu is considered a low-resource language. Finally, the results of the experiments will be discussed.

4.1 Markers in Urdu and English

A marker is a free or bound morpheme that indicates the grammatical function of the marked word phrase, or sentence. For example, Case is a system of marking dependent-nouns for the type of relationship they bear with their heads. The subject or object of a verb can be marked by a case system.

There are different marking schemes in different families of languages. Many Indo-European languages use morphological inflections. In English, a fixed word order language, the subject and object are distinguished by their position. South-Asian Languages, especially New Indo-Aryan languages (1000 AD - present) including Urdu devised a new method to mark cases. Along with few morphological inflections that are remains of Old and Middle Indo Aryan languages, these languages use clitics and postpositions to mark the case [Ahmed, 2007]. There are seven cases in Urdu listed in Table 4.1. Spencer [2005] discussed case system of Urdu and difference between inflection, clitic[1] and postposition.

[1]In morphology and syntax, a *clitic* is a morpheme that is grammatically independent but

Case	Marker
Nominative	ϕ
Ergative	ne|نے
Accusative	ko|کو
Dative	ko|کو
Instrument	se|سے
Ablative	se|سے
Locative	meñ|میں, pər|پر, ...

Table 4.1: Case Markers in Urdu

The following example illustrates the difference between case marking in Urdu and English.

English	The hunter killed the lion with the gun .
Urdu	شکاری نے بندوق سے شیر مارا ۔
Gloss	. mārā šer se bənədūq ne šəkārī

In above example, the subject in Urdu sentence is marked with ergative marker "ne | نے", object is marked with nominative "ϕ" and the instrument is marked with the instrumental "se | سے". On the other hand, in the English sentence "the hunter" comes before the verb, which makes the hunter subject and "the lion" comes after the verb hence it is the object. The instrument is marked with the preposition "with".

Case markers are also important in making a language free-word order. In the above Urdu example we can easily move the subject and object anywhere in the sentence given that the markers remains the same. e.g.

SOV	بندوق سے شکاری نے شیر مارا ۔ . mārā šer ne šəkārī se bənədūq
OSV	شیر بندوق سے شکاری نے مارا ۔ . mārā ne šəkārī se bənədūq šer
VOS	مارا شیر بندوق سے شکاری نے ۔ . ne šəkārī se bənədūq šer mārā

The meaning of all the above sentences will remain exactly the same. While for the English side, if we try to juggle with words in the sentence, it will alter the whole semantics of the sentence. e.g.

phonologically dependent on another word or phrase.

* The lion killed the hunter with the gun .

Cases are not the only property of the language which requires marking, other features of the language are also marked e.g. in both English and Urdu the attributes of nouns are marked using inflections.

boy - ləṛəkā | لڑکا boys - ləṛəke | لڑکے

In Urdu nouns also have gender marked with inflections,

girl - ləṛəkī | لڑکی girls - ləṛəkyāñ | لڑکیاں

The aspect and tense of a verb in Urdu is also marked using postpositions and inflectional changes. In English the aspect and tense of verbs are marked with auxiliary verbs placed before the verb and inflections in main verb e.g. –ing, –ed, –s.

English	I am going .
Urdu	میں جا رہا ہوں ۔
Gloss	. hūñ rəhā jā meñ
English	He had finished work .
Urdu	وہ کام ختم کر چکا تھا ۔
Gloss	. thā čəkā kr xətəm kām wəh
English	She sleeps .
Urdu	وہ سوتی ہے۔
Gloss	. he sotī wəh

An interesting and notable thing here is the use of extra words (tokens) in Urdu as markers. It is really important for a statistical machine translation system to align these extra tokens correctly.

4.2 Factor-Based Models

Rich morphology often poses a challenge to statistical machine translation, since a multitude of word forms derived from the same lemma fragment the data and lead to sparse data problems. Essentially, all morphological forms of a word and its translations have to exist in the training corpus, and every word has to appear with every possible case marker, which will require an impossible amount

of training data. Therefore, it is imperative to make it possible for the system to learn general rules for morphology and case marking.

The basic idea behind factored translation models is to represent phrases not simply as sequences of fully inflected words, but instead as sequences containing multiple levels of information. A factored language model views a word as a vector of k factors:

$$w_i = f_i^1, f_i^2, ..., f_i^k$$

Factors can be anything, including morphological classes, stems, roots, and other such features in highly inflected languages (e.g., Arabic, German, Finnish, etc.), or data-driven word classes or semantic features useful for sparsely inflected languages (e.g., English). A two-factor language model is generated by standard class-based [Brown et al., 1992] language models, where one factor is the word class and the other is word itself. A factor-based model is a model over factors, i.e.

$$p(f_t^{1:k}|f_{t-1:t-n}^{1:k})$$

that can be factored as a product of probabilities of the form $p(f|f_1, f_2, ..., f_n)$. In factor-based statistical machine translation the task is twofold: One is to find an appropriate set of factors and two is to include an appropriate statistical model over those factors [Bilmes and Kirchhoff, 2003, Dep, 2008].

Factored translation models closely follow the statistical modeling methods used in phrase-based models. Each of the mapping steps is modeled by a feature function. This function is learned from the training data, resulting in translation tables and generation tables [Koehn et al., 2007a].

Koehn et al. [2007a] showed improvement for a number of language pairs using factor-based models. Ramanathan et al. [2009] applied factor-based models to English-Hindi language pair. Yeniterzi and Oflazer [2010] showed improvements with syntax-to-morphology mapping for factored translation models for English-to-Turkish.

However, the factor-based SMT approach is dependent on the quality of factors on both source and target sides. The mapping of factors from source to target also significantly affects the translation quality. Also, compared to the phrase-based models, the decomposition of the phrase translation into several mapping steps creates additional computational complexity.

4.3 Artificial Markers on Source-side

In this thesis, we are focusing on low-resource scenario for the target side. In this regard, the generation of factors for the target side is a limiting factor. To tackle this problem, we have introduced a new source-side preprocessing approach based on basic knowledge of the target (Urdu) language. *The idea is to add artificial tokens in English to mark subject, object and aspects of verbs.* Using this technique we can use a normal phrase-based approach with single-factored model for statistical machine translation.

In Section 3.1, we showed that on average Urdu is using more words than English per sentence. We have also discussed in Section 4.1 the differences between the marking system of English and Urdu. The fact that Urdu is using separate words as markers is one of the reasons why the average sentence length in Urdu is higher. The important question here is, *'What are the effects of these extra tokens on word alignment?'*. The analysis of Urdu case-marker alignments for the EMILLE corpus (training set only) gives us the following statistics:

Marker	نے|ne		کو|ko		سے|se	
Number of times aligned in corpus	364		2791		2263	
Most frequent word alignments	82	have	1113	to	428	from
	40	has	345	the	143	with
	23	the	149	should	124	by
	18	made	131	a	105	than
	6	took	96	have	99	before
	6	had	60	sure	96	of
	6	did	44	tell	78	contact
	6	decided	34	are	59	over
	4	started	33	give	50	-
	4	reported	27	get	44	most

Table 4.2: Statistics of Urdu case-marker alignments in EMILLE English-Urdu parallel corpus

It can be seen from the Table 4.2 that case-markers in Urdu are high frequency words which are wrongly aligned to various words in English, specially for نے|ne and کو|ko for which there are no corresponding words in English. On the other hand, the instrumental marker سے|se have corresponding prepositions in English,

which is why it is correctly aligned to those prepositions. This gives us the motivation to introduce artificial words in English sentences corresponding to the Urdu case markers, so that the alignments can be improved. Similarly, we can add aspect and tense markers in English.

It is mentioned already in Section 2.4 that the statistical machine translation is dependent on the translation model. The high frequency of wrong alignments of case markers will badly effect the translation model and the extraction of phrases. Figure 4.1 shows an example of wrong alignment in a sentence produced by Giza++ toolkit.

English Yeargin won widespread local support .

Urdu ۔ جیتی حمایت نجی زیادہ بہت نے یئرگن

Gloss . jītī h əmāyət nəjī zəyādəh bəhət ne yəerəgən

	یئرگن	نے	بہت	زیادہ	نجی	حمایت	جیتی	۔
Yeargin	•							
won		•					•	
widespread			•	•				
local					•			
support						•		
.								•

Figure 4.1: Example of alignment error: The Urdu word ‘نے|ne’ wrongly aligned to English word ‘won’

In the above example only one node is wrongly aligned but it can leads to two wrong phrase extractions. The phrase ‘Yeargin won’ can be translated to ‘یئرگن نے’ and the phrase ‘won widespread’ can be aligned to ‘نے بہت زیادہ’, both of which are wrong. However, this problem can be resolved by introducing one extra word for نے|ne in the English sentence. Figure 4.2 shows the same example with an artificial marker.

4.3.1 Generation of Artificial Markers

The markers we introduce in English are based on the following basic knowledge of Urdu:

	یرگن	نے	بہت	زیادہ	مقامی	حمایت	جیتی	۔
Yeargin	•							
nsubj		•						
won							•	
widespread			•	•				
local					•			
support						•		
.								•

Figure 4.2: Example of alignment with artificial marker: The Urdu word 'نے|ne' aligned to the artificial marker 'nsubj'

- The subject and object are marked with postpositions.
- The aspect and tense of the verb is also marked using postpositions.
- Urdu does not use articles and we can drop the article 'the' from English.

The first step to generate the markers in English side is to identify the grammatical relations in the English sentence. For this purpose we use the Stanford dependency parser [De Marneffe et al., 2006]. The Stanford dependency parser uses 55 relations to express the dependencies among the various words in a sentence. These relations form a hierarchical structure with the most general relation at the root. There are various argument relations like subject, object, objects of prepositions, and clausal complements, modifier relations like adjectival, adverbial, participial, and infinitival modifiers, and other relations like coordination, conjunct, expletive, and punctuation. However, we are only interested in the subject and object relations.

The Stanford parser marks the subject with 'nsubj' relation, marks the direct object with 'dobj' relation and marks the indirect object with 'iobj' relation. We use these relations and add them as postpositions in English sentence after the corresponding head noun.

For marking the aspect and tense of the verb, we use the part-of-speech tag of the verb. The Stanford parser uses the Penn Treebank tagset [Marcus et al., 1993] for part-of-speech tags and phrasal categories. We filter out the tags for verb and append them after the verb in English sentences. The part-of-speech tags used in our experiments are listed in Table 4.3.

VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present

Table 4.3: Part-of-speech tags used to mark verbs

The algorithm for the addition of artificial markers can be formalized as in Algorithm 1:

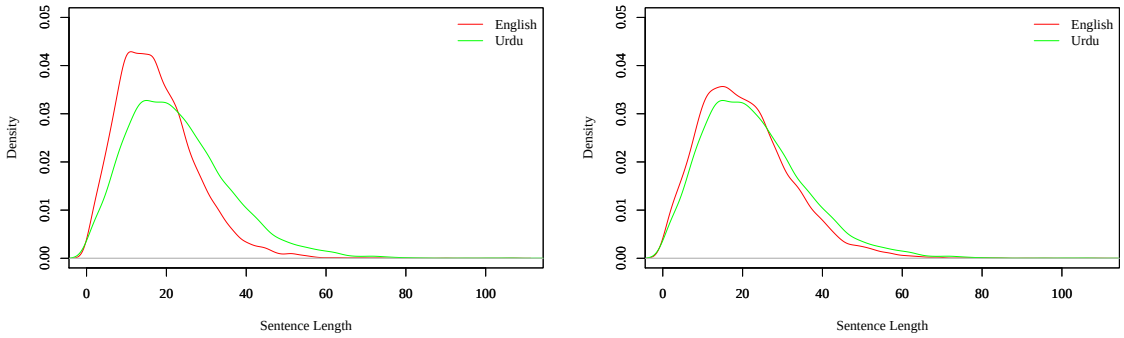
Algorithm 1 Artificial Marker Generation for English

Require: A set S consisting of English sentences.

$S' \Leftarrow \phi$ {new set for sentences with artificial markers}
for all sentence $s \in S$ **do**
  $s' \Leftarrow \phi$
  for all node $n \in Parse(s)$ **do**
    if $n.value! = ``the''$ {ignore definite article the} **then**
      $append(n.value, s')$
      if $n.relation \in$ set of required relations {nsubj, dobj, iobj} **then**
        $append(n.relation, s')$
      end if
      if $n.tag \in$ set of verb tags {vb, vbd, vbg, vbn, vbp, vbz} **then**
        $append(n.tag, s')$
      end if
    end if
  end for
  $S' \Leftarrow S' \cup s'$
end for
return S'

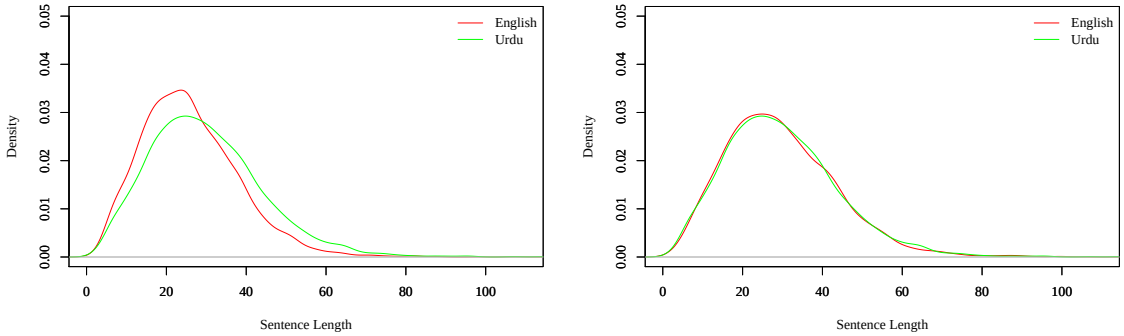
4.3.2 Improvements in Word Alignment

After adding the artificial markers in the source corpus the obvious change is the increase in sentence lengths. The noticeable thing is that more English sentences now have comparable lengths with Urdu sentences as highlighted in Figure 4.3 and 4.4.



(a) Sentence lengths without markers (b) Sentence lengths with markers

Figure 4.3: EMILLE Corpus sentence length analysis after adding markers



(a) Sentence lengths without markers (b) Sentence lengths with markers

Figure 4.4: Penn Treebank Corpus sentence length analysis after adding markers

Furthermore, addition of markers at source side helps in improving alignments. Table 4.4 lists the alignments of Urdu case-markers after adding the artificial markers in English. Most of the Urdu case-markers are now aligned with the added markers in English as opposed to the wrong alignments of Urdu case-markers mostly with the English content words as previously shown in Table 4.2. The change in the probabilities of all words that were previously aligned to these markers should also have positive impact on the overall alignment and phrase extraction of the system.

Marker	نے|ne		کو|ko		سے|se	
Number of times aligned in corpus	571		3866		2739	
Most frequent word alignments	254	nsubj	1819	dobj	499	dobj
	83	have	614	to	419	from
	50	vbd	600	nsubj	144	by
	34	has	241	vb	112	with
	33	vbn	76	should	106	before
	11	made	75	iobj	100	than
	7	took	39	vbn	94	to
	7	did	28	make	77	of
	7	decide	25	a	74	vbg
	3	ve	22	vbg	60	contact

Table 4.4: Statistics of Urdu case-marker alignments in EMILLE English-Urdu parallel corpus after the introduction of artificial markers in English sentences

4.4 Experiments and Results

Using the Algorithm 1 described above we preprocess our corpora to generate the training and testing instances. The details of the corpora are given in Section 3.1. We then followed the decoder training, tuning and testing steps of MOSES as described in Section 3.5. The results of the experiments are listed in Table 4.5.

Test Corpus	NIST	BLEU
EMILLE	4.8611	0.1927
Penn Treebank	5.5632	0.1786
PENN+EMILLE	5.2818	0.1554

Table 4.5: Results of marker experiments

Test Corpus	Average BLEU	Lower limit	Upper limit
EMILLE	0.19281	0.17300	0.21244
Penn Treebank	0.17864	0.16179	0.19415
PENN+EMILLE	0.15545	0.14326	0.16815

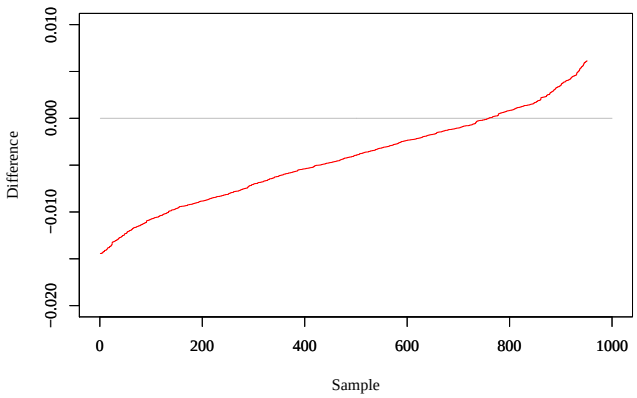
Table 4.6: 95% confidence intervals for markers results

We have developed a strong hypothesis in the previous sections that adding artificial markers in English should increase the English-to-Urdu translation quality. Apparently, the results of the experiment do not support our hypothesis. Although, there is a slight increase in the BLEU score for EMILLE corpus. However, from the significance testing results (Figure 4.5), it is visible that for 95% confidence interval the results are not significant.

	Average BLEU	Confidence Interval
Baseline	0.18876	(0.17114 to 0.20643)
Markers	0.19304	(0.17502 to 0.21236)
Difference	-0.00435	(-0.0144 to 0.00637)

	Baseline	Markers
Baseline	x	?
Markers	?	x

(a) Paired bootstrap resampling results with 95% confidence interval



(b) Difference (baseline – markers) plot of paired bootstrap resample for 95% confidence interval

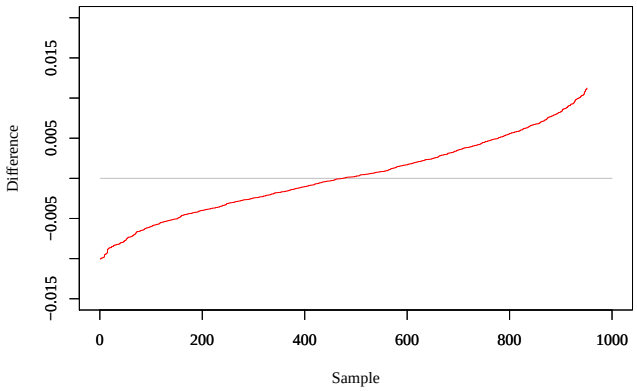
Figure 4.5: Comparison of markers results with baseline results for EMILLE corpus

The marker experiments produce similar results for Penn Treebank corpus. The results are not significant with 95% confidence interval (Figure 4.6).

	Average BLEU	Confidence Interval
Baseline	0.17883	(0.16271 to 0.19505)
Markers	0.17860	(0.16214 to 0.19555)
Difference	0.00023	(-0.00995 to 0.01125)

	Baseline	Markers
Baseline	x	?
Markers	?	x

(a) Paired bootstrap resampling results with 95% confidence interval



(b) Difference (baseline – markers) plot of paired bootstrap resample for 95% confidence interval

Figure 4.6: Comparison of markers results with baseline results for Penn Treebank corpus

Example 1	
Source	you can usually get extra money for your family .
Reference	آپ عام طور پر اپنے کنبے کے لئے اضافی رقم حاصل کر سکتے ہیں - . āp ʔām ṭwr pər apəne kənəbe ke lae əẓāfī raqəm ḥāṣəl kər səkəte heñ .
Baseline Output	کر سکتے / سکتی ہیں عام طور پر اپنے خاندان کے لئے بھی اضافی kər səkəte / səkətī heñ ʔām ṭwr pər apəne xānədān ke lie bhī əẓāfī رقم حاصل کر سکتے ہیں - raqəm ḥāṣəl kər səkəte heñ . Matching n-grams: 1-grams: 16, 2-grams: 10, 3-grams: 7, 4-grams: 5
Markers Output	کر سکتے ہیں اور آپ کے عام طور پر اپنے کنبے کے لئے اضافی kər səkəte heñ or āp ke ʔām ṭwr pər apəne kənəbe ke lae əẓāfī رقم حاصل کر سکتے ہیں - raqəm ḥāṣəl kər səkəte heñ . Matching n-grams: 1-grams: 19, 2-grams: 15, 3-grams: 13, 4-grams: 11

Example 2	
Source	help with exceptional expenses which are hard to pay out of your regular income .
Reference	غیر معمولی اخراجات جن کی اپنی باقاعدہ آمدنی سے ادائیگی کرنا آپ کے ğyr məʾəmolī əxərājāt jən kī apənī bāqāʾədəh se āmədənī ədāīgī kərənā āp ke لئے مشکل ہو ، میں مدد - lae məšəkəl ho , meñ mədəd . Matching n-grams: 1-grams: 13, 2-grams: 6, 3-grams: 4, 4-grams: 2
Baseline Output	غیر معمولی اخراجات کے لئے مدد جسکا کرنا مشکل ہے کی اپنی باقاعدہ ğyr məʾəmolī əxərājāt ke lie mədəd jəsəkā kərənā məšəkəl he kī apənī bāqāʾədəh آمدنی سے - āmədənī se . Matching n-grams: 1-grams: 13, 2-grams: 6, 3-grams: 4, 4-grams: 2
Markers Output	غیر معمولی اخراجات کے لئے مدد ہیں جو ادا کرنا مشکل ہوتا ہے ğyr məʾəmolī əxərājāt ke lie mədəd heñ jo ādā kərənā məšəkəl hotā he کہ اپنی باقاعدہ آمدنی سے - kəh apənī bāqāʾədəh āmədənī se . Matching n-grams: 1-grams: 12, 2-grams: 5, 3-grams: 3, 4-grams: 1

Example 3	
Source	the pressure for change will rise with costs .
Reference	تبدیلی کے لیے دباؤ ، لاگتوں کے ساتھ بڑھے گا - təbədīlī ke lye dəbāo , lāgətoñ ke sāth bəṛhe gā .
Baseline Output	تبدیلی کی بلندی کے لیے دباؤ کرے گا ، لاگتوں کے ساتھ - təbədīlī kī bələnədī ke lye dəbāo kəre gā , lāgətoñ ke sāth . Matching n-grams: 1-grams: 10, 2-grams: 5, 3-grams: 3, 4-grams: 1
Markers Output	دباؤ کے ساتھ تبدیلی کی لاگت کے لیے اضافہ کرے گی - dəbāo ke sāth təbədīlī kī lāgət ke lye əẓāfəh kəre gī .
	Matching n-grams: 1-grams: 7, 2-grams: 2, 3-grams: 0, 4-grams: 0

Table 4.7: Sample output of marker system

5. Reordering

Preprocessing by reordering the source side sentences in statistical machine translation has proved to be useful for a number of language pairs e.g. Visweswariah et al. [2011] shows significant improvements for Hindi-to-English, Urdu-to-English and English-to-Hindi; Jawaid and Zeman [2011] shows improvement using manually written reordering rules for English-to-Urdu; Visweswariah et al. [2010] applied reordering for various languages and showed improvements for English-to-Spanish, French and Hindi but they got negative results for English-to-German. In this chapter, we focus on the reordering issues between English-to-Urdu machine translation, which is the second research objective of our thesis. Section 2.6 provided an overview of different reordering techniques.

The first part of the chapter describes a source-side reordering technique based on hand written tree transformation rules [Jawaid, 2010]. Then, we present our approach to automatically learn the reordering rules using a small set of manually aligned sentences. The results of both manual reordering rules and automatically learnt reordering rules are compared with baseline and with each other.

5.1 Manual Reordering Rules

Urdu word order is significantly different from the word order of English; English is a fixed word order language that follows a Subject Verb Object (SVO) order. This is in contrast to Urdu, which is a free word order language, however, the typical order followed is Subject Object Verb (SOV). This can result in words that are close in English moving arbitrarily far apart in Urdu depending on the length of the noun phrase representing the object and the length of the verb phrase [Visweswariah et al., 2011]. These long distance reordering are hard to model for a phrase based system like MOSES.

Use of syntax is a potential solution for long distance reordering. The basic idea is to change the word order of the source sentence to make it more similar to word order of the target sentence in a preprocessing step. This preprocessing step is inspired by previous approaches like Xia and McCord [2004], which split translation into two steps:

$$S \rightarrow S' \rightarrow T$$

Where S is the source language sentence which is reordered in the first step according to the word order of target language, resulting in the reordered sentence

S'. In the second step the reordered sentence S' is monotonously translated into the target language sentence T.

5.1.1 The Preprocessing Step

Our preprocessing approach is based on the technique used by Jawaid [2010]. The English sentences are reordered using hand written rules for English-to-Urdu tree-to-tree grammar mapping [Ata et al., 2007].

We use the Rule-Based English to Urdu Machine Translation System (RBMT) [Ata et al., 2007] for the preprocessing of the source corpus. The RBMT system transformation module takes source parse tree as an input and produces the reordered source tree on the basis of linguistically rich mapping rules written by human experts. Then, the reordered source tree is passed to a translation module which performs the task of the translation. For our experiments we took out the RBMT system's transformation module and used it as a preprocessor for our corpora.

The first step of the preprocessing is to parse the source sentence. The RBMT system uses Stanford Parser to generate the parse trees. The transformations are applied in the second step recursively in a breadth-first traversal. The rules are based on the assumption that each subtree is independent of the rest of the tree and nodes inside a subtree can take any permutation within that subtree. Table 5.1 shows some example transformation rules. Further implementation details of the RBMT system is available in Ata et al. [2007] and Jawaid [2010].

Example 1	
Source side Grammar Rule	VP → VP NP
Target side Transformation Rule	VP → NP VP

Example 2	
Source side Grammar Rule	VP → VB* PP
Target side Transformation Rule	VP → PP VB*

Example 3	
Source side Grammar Rule	VP → default
Target side Transformation Rule	VP → reverse

Table 5.1: Example of manual reordering rules

In those cases where exact match of the head node and the child nodes se-

quence is not found, then that rule is retrieved whose left hand side is similar to the head node and the right hand side approximately matches with the child node in the same order. Example 2 shows such generic grammar rule in which (*) can be matched with any of VBZ, VBP, VBN etc.

If the transformation module is unable to find any match for the current subtree then the default rule of the head node is applied. Example 3 shows the default rule for the verb phrase. According to the transformation rule, all the child nodes should be placed in the reverse order. Appendix A lists all the manual rules used.

The algorithm for the manual reordering of source English sentences can be formalized as in Algorithm 2:

Algorithm 2 Manual Reordering Algorithm

Require: A set S consisting of English sentences.

$S' \Leftarrow \phi$ {set of resultant reordered sentences}
for all sentence $s \in S$ **do**
  Tree $root \Leftarrow Parse(s)$
  Queue $q \Leftarrow \phi$
  $enqueue(q, root)$
  while $q \neq \phi$ **do**
    $subtree \Leftarrow dequeue(q)$
    $rule \Leftarrow GetRule(subtree)$
    $transformation \Leftarrow SearchTransformation(rule)$
    if $transformation \neq \phi$ **then**
      apply transformation on t
    else
      apply default transformation on t
    end if
    $enqueue(q, subtree.children)$
  end while
  $S' \Leftarrow S' \cup string(root)$
end for
return S'

5.1.2 Experiments and Results

Using the Algorithm 2 described above we preprocess our corpora to generate the training and testing instances. The details of the corpora are given in Section 3.1. We then followed the decoder training, tuning and testing steps of MOSES as described in Section 3.5. The results of the experiments are listed in Table

5.2. Furthermore, significance testing is performed and we get 95% confidence intervals for BLEU metric for each corpus (Table 5.3).

Test Corpus	NIST	BLEU
EMILLE	5.1323	0.2047
Penn Treebank	5.6008	0.1969
PENN+EMILLE	5.6127	0.1974

Table 5.2: Results of manual reordering experiments

Test Corpus	Average BLEU	Lower limit	Upper limit
EMILLE	0.20469	0.18385	0.22518
Penn Treebank	0.19657	0.17915	0.21328
PENN+EMILLE	0.19738	0.18399	0.20953

Table 5.3: 95% confidence intervals for manual reordering results

Pairwise bootstrap resampling shows that the manual reordered system performs significantly better than the baseline for both EMILLE and PENN corpus with a 99% confidence interval. The results of comparison of baseline and manually reordered system is given in Figure 5.1 and 5.2.

5.2 Automatic Learning of Reordering Rules

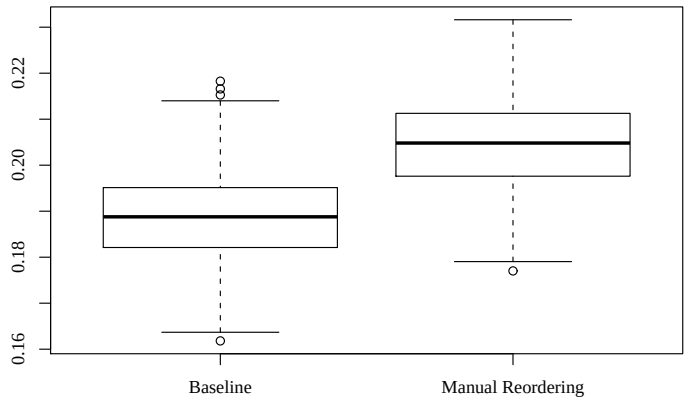
The previous section shows that the use of syntax based reordering is an efficient way to model long distance reordering. However, writing manual rules is a resource intensive task and requires a lot of human effort. Various attempts have been made to learn the reordering rules automatically using data driven methods [Visweswariah et al., 2010, Zhang et al., 2007, Lavie et al., 2003] that shows promising results.

In this research, the main goal is to show that for low resource languages we can use a small amount of data to learn reordering rules. For this purpose we develop a simple tree-to-string algorithm to learn the source side reordering rules automatically from a small set of manually aligned sentences. The technique is similar to Visweswariah et al. [2010] and Liu et al. [2006]. However, we are not using any probabilistic approach in either extracting or applying the reordering rules. The idea is to extract frequently occurring rules in a format similar to the

	Average BLEU	Confidence Interval
Baseline	0.18871	(0.16514 to 0.21175)
Manual Reordering	0.20479	(0.17908 to 0.2314)
Difference	-0.01607	(-0.03135 to -0.00083)

	Baseline	Manual Reordering
Baseline	x	<
Manual Reordering	>	x

(a) Paired bootstrap resampling results with 99% confidence interval



(b) Boxplot of pairwise bootstrap resampling of baseline and manual reordering with 99% confidence interval

Figure 5.1: Comparison of manual reordering results with baseline results for EMILLE corpus

manual reordering approach [Jawaid, 2010] and apply the same process that we followed in the previous manual reordering method.

5.2.1 Manual Alignment

To facilitate the automatic learning of reordering rules, we need a correctly aligned corpus. The alignment generated using automatic algorithms is not worthy enough because errors in alignment will make the system learn invalid reordering rules.

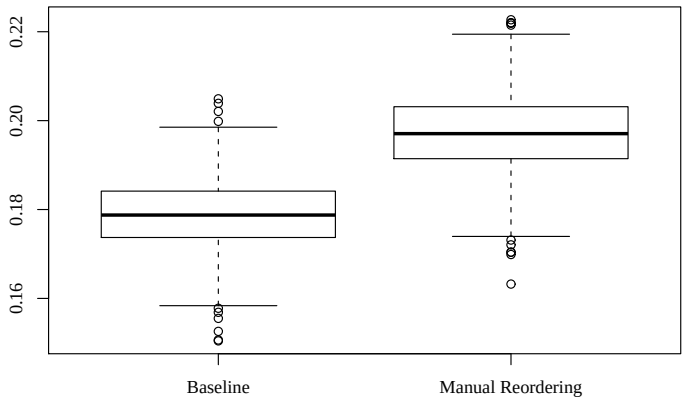
For this purpose we randomly choose 100 sentences from the Penn English-to-Urdu corpus and aligned them manually using UMIACS word alignment tool[1].

[1]UMIACS word alignment tool was originally written by Rebecca Hwa and later modi-

	Average BLEU	Confidence Interval
Baseline	0.17887	(0.15778 to 0.19834)
Manual Reordering	0.19714	(0.17312 to 0.21945)
Difference	-0.01826	(-0.02982 to -0.00698)

	Baseline	Manual Reordering
Baseline	x	<
Manual Reordering	>	x

(a) Paired bootstrap resampling results with 99% confidence interval



(b) Boxplot of pairwise bootstrap resampling of baseline and manual reordering with 99% confidence interval

Figure 5.2: Comparison of manual reordering results with baseline results for Penn Treebank corpus

Figure 5.3 shows the UI of the alignment tool with an example alignment.

The manual alignment tool generate an alignment file "alignment.i" for each sentence, where i is the sentence number. For example, for the sentence pair:

English:	After the race, Fortune 500 executives drooled like schoolboys over the cars and drivers.
Urdu:	دوڑ کے بعد ، فورچون ۵۰۰ افسروں ، کاروں اور ڈرائیوروں کے doṛ ke bəʾəd , forəčwn 500 əfəsəroñ , kāroñ or ḍarāīwroñ ke بارے میں سکول کے بچوں کی طرح بہت زیادہ خوش تھے ۔ bāre meñ səkwl ke bəčoñ kī ṭərəḥ bəhət zəyādəh xūš the .

fied by Nitin Madnani. It is a Java based standalone application available on the website [http://www.umiacs.umd.edu/ nmadnani/alignment/forclip.htm]

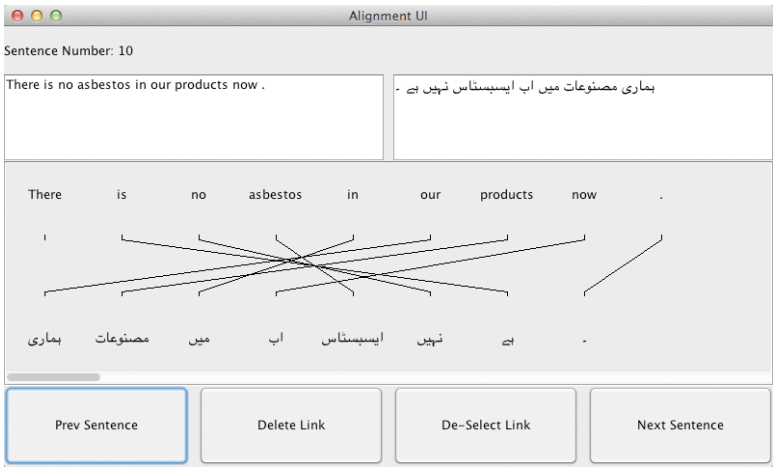


Figure 5.3: Manual Alignment tool with an example alignment of English-Urdu sentence

the generated alignment file contains the following contents:

3	1	(race, دوڑ)
1	2	(after, کے)
1	3	(after, بعد)
4	4	(„ ،)
5	5	(fortune, فورچون)
6	6	(500,)
7	7	(executives, افسروں)
13	9	(cars, کاروں)
14	10	(and, اور)
15	11	(drivers, ڈرائیوروں)
10	15	(schoolboys, سکول)
10	16	(schoolboys, کے)
10	17	(schoolboys, بچوں)
9	18	(like, کی)
9	19	(like, طرح)
8	22	(drooled, خوش)
8	23	(drooled, تھے)
16	24	(., ۔)
8	20	(drooled, بہت)
8	21	(drooled, زیادہ)

First column represents the source index, the second column represents the target index and the final column is the aligned word pair.

5.2.2 Learning Automatic Rules

We developed a very simple algorithm based on the parse tree of source sentence and source-to-target alignments. The first step is to parse the source (English) sentence. Again, we use Stanford parse for parsing, the parsing gives us a parse tree say S_t. Given word alignments A, where A_{ij} defines an alignment between the i^{th} source word to the j^{th} target word. We extract the reordering rules in the following manner.

First we assign each node of the source tree S_t an initial value. The value of each leaf node is initialized with its order in the sentence and the value of parent node is calculated as the mean of all children nodes values. Each node value is calculated recursively till we reach the root node, see Figure 5.4 and 5.5.

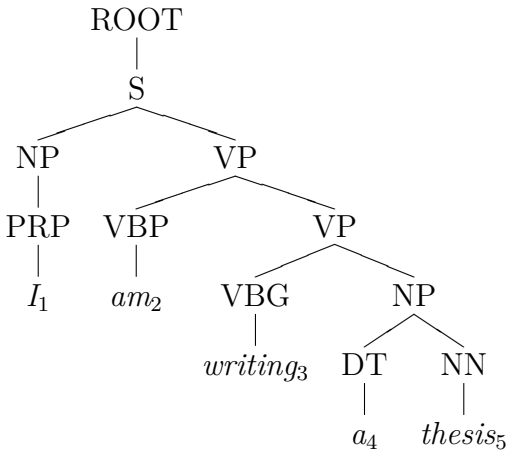


Figure 5.4: Step 1a: Initialization of leaf nodes

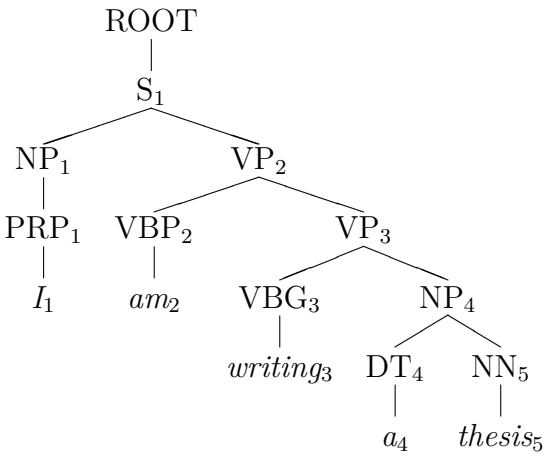


Figure 5.5: Step 1b: Initialization of parent nodes (for simplicity we use integer division)

In the second step we read the alignment A_{ij} corresponding to each leaf node

and replace the value i with j, if the alignment of a node is not defined there is no change in the value. Also once all the child nodes of a subtree are updated we calculate a new value for the parent (5.6).

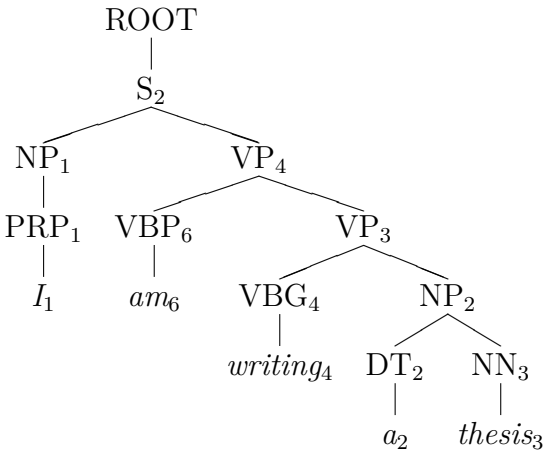


Figure 5.6: Step2: Updating the source nodes with target alignments

Sorting the children of each subtree according to the new values gives us a reordered tree as in Figure 5.7

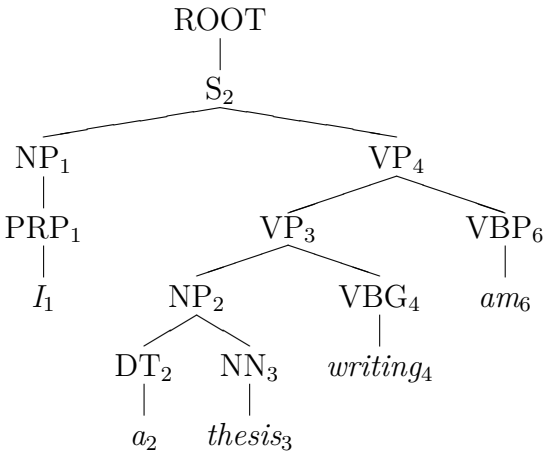


Figure 5.7: Step3: Sorting the children of each subtree

In the last step we compare each subtree with its initial order that gives us the required reordering rules, e.g. from the example in 5.7, we can extract the rules given in Table 5.4.

5.2.3 Experiments and Results

We run the algorithm as described in the previous section on the 100 manually aligned sentences which gives us 55 reordering rules, Appendix B lists all the

Rule 1	
Source side Grammar Rule	S → NP VP
Target side Transformation Rule	S → NP VP

Rule 2	
Source side Grammar Rule	VP → VBP VP
Target side Transformation Rule	VP → VP VBP

Rule 3	
Source side Grammar Rule	VP → VBG NP
Target side Transformation Rule	VP → NP VBG

Rule 4	
Source side Grammar Rule	NP → DT NN
Target side Transformation Rule	NP → DT NN

Table 5.4: Automatically extracted rules

extracted rules. We then place these rules in the transformation module of our manual reordering engine and preprocess our corpora. Then we follow the MOSES steps as described in Section 3.5. The results of automatic reordering rules are listed in Table 5.5 and 5.6. A comparison with the baseline is given in Figure 5.8 and 5.11, the automatic learning of the reordering performed significantly better for EMILLE corpus with 95% confidence interval. For Penn Tree bank corpus the system shows slight improvement but with 95% confidence interval the improvement is not significant.

Test Corpus	NIST	BLEU
EMILLE	4.9755	0.2055
Penn Treebank	5.5797	0.1808
PENN+EMILLE	5.4920	0.1827

Table 5.5: Results of automatic reordering experiments

5.2.4 Manual vs Automatic Reordering

We also performed paired bootstrap resampling for manual reordering and automatic reordering methods. The performance of the automatic reordering is

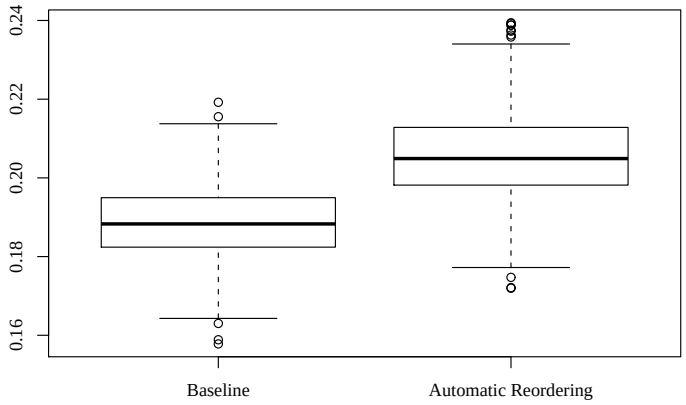
Test Corpus	Average BLEU	Lower limit	Upper limit
EMILLE	0.20567	0.18342	0.22966
Penn Treebank	0.18072	0.16402	0.19634
PENN+EMILLE	0.19738	0.18399	0.20953

Table 5.6: 95% confidence intervals for automatic reordering results

	Average BLEU	Confidence Interval
Baseline	0.18862	(0.17086 to 0.20777)
Automatic Reordering	0.20567	(0.18342 to 0.22966)
Difference	-0.01705	(-0.03106 to -0.00337)

	Baseline	Automatic Reordering
Baseline	x	<
Automatic Reordering	>	x

(a) Paired bootstrap resampling results with 95% confidence interval



(b) Boxplot of pairwise bootstrap resampling of baseline and automatic reordering with 95% confidence interval

Figure 5.8: Comparison of automatic reordering results with baseline results for EMILLE corpus

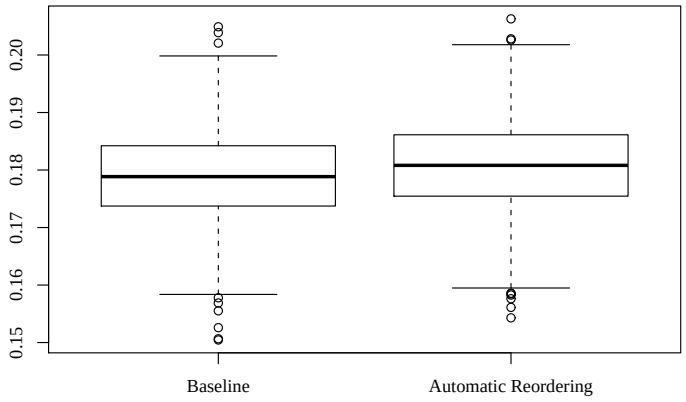
comparable with manual reordering for EMILLE corpus 5.10. However, for the Penn treebank corpus the manual reordering showed significantly better results with a 99% confidence interval.

Given the fact that manual reordering rules are developed with a lot of efforts and with cumbersome analysis of both the languages. The results for automatic learning are highly encouraging as we used only 100 sentences to learn the

	Average BLEU	Confidence Interval
Baseline	0.17892	(0.16231 to 0.19504)
Automatic Reordering	0.18072	(0.16402 to 0.19634)
Difference	-0.00181	(-0.01207 to 0.00783)

	Baseline	Automatic Reordering
Baseline	x	?
Automatic Reordering	?	x

(a) Paired bootstrap resampling results with 95% confidence interval



(b) Boxplot of pairwise bootstrap resampling of baseline and automatic reordering with 95% confidence interval

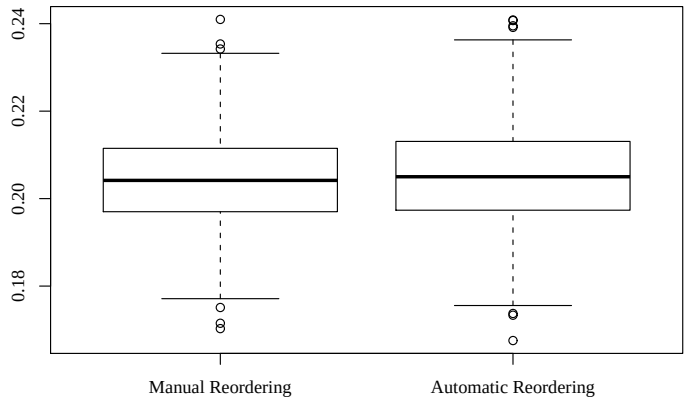
Figure 5.9: Comparison of automatic reordering results with baseline results for Penn Treebank corpus

reordering rules without using any linguistic knowledge of both the languages.

	Average BLEU	Confidence Interval
Manual Reordering	0.20441	(0.18473 to 0.22657)
Automatic Reordering	0.20535	(0.18272 to 0.22889)
Difference	-0.00093	(-0.01436 to 0.01162)

	Manual Reordering	Automatic Reordering
Manual Reordering	x	?
Automatic Reordering	?	x

(a) Paired bootstrap resampling results with 95% confidence interval



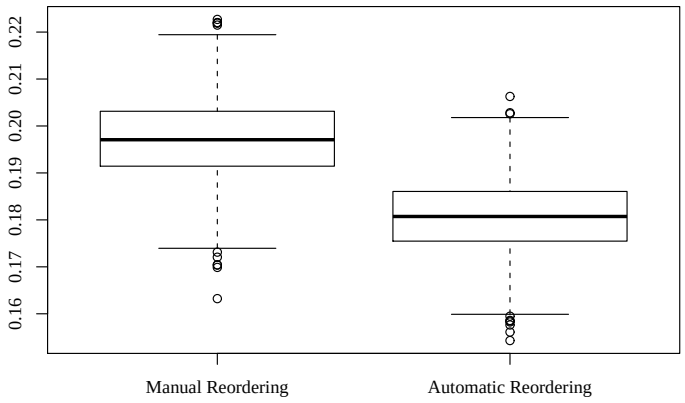
(b) Boxplot of pairwise bootstrap resampling of manual and automatic reordering with 95% confidence interval

Figure 5.10: Comparison of manual reordering results with automatic reordering results for EMILLE corpus

	Average BLEU	Confidence Interval
Manual Reordering	0.19714	(0.17312 to 0.21945)
Automatic Reordering	0.18070	(0.15858 to 0.20092)
Difference	0.01643	(0.00064 to 0.03207)

	Manual Reordering	Automatic Reordering
Manual Reordering	x	>
Automatic Reordering	<	x

(a) Paired bootstrap resampling results with 99% confidence interval



(b) Boxplot of pairwise bootstrap resampling of manual and automatic reordering with 99% confidence interval

Figure 5.11: Comparison of manual reordering results with automatic reordering results for Penn Treebank corpus

Example 1	
Source	cold weather payments can help with extra heating expenses in cold weather .
Reference	کولڈ ویدر پیمنٹس سے سرد موسم میں (گھر کو) گرم کرنے koləḍ wedər pymənəṭəs se sərəd mosəm meñ (ghər ko) gərəm kərəne کے اضافی اخراجات میں مدد مل سکتی ہے - ke əẓāfī əxərājāt meñ mədəd məl səkətī he . Matching n-grams: 1-grams: 14, 2-grams: 9, 3-grams: 6, 4-grams: 4
Manual Reordering Output	کولڈ ویدر پیمنٹس سے کولڈ ویدر میں کرنے کے اضافی اخراجات میں مدد koləḍ wedər pymənəṭəs se koləḍ wedər meñ kərəne ke əẓāfī əxərājāt meñ mədəd کر سکتے ہیں - kər səkəte heñ . Matching n-grams: 1-grams: 14, 2-grams: 9, 3-grams: 6, 4-grams: 4
Automatic Reordering Output	کولڈ ویدر پیمنٹس سے سرد موسم میں (گھر کو) گرم کرنے koləḍ wedər pymənəṭəs se sərəd mosəm meñ (ghər ko) gərəm kərəne کے اضافی اخراجات میں مدد مل سکتی ہے - ke əẓāfī əxərājāt meñ mədəd məl səkətī he . Matching n-grams: 1-grams: 22, 2-grams: 21, 3-grams: 20, 4-grams: 19
Example 2	
Source	it is important that we can be sure of your identity when you make a claim .
Reference	یہ بات اہمیت رکھتی ہے کہ جب آپ کلیم کریں تو ہمیں آپ yəh bāt əhəmīt rəkhətī he kəh jəb āp kəlym kəreñ to] həmeñ āp کی شناخت کا یقین ہو - kī šənāxət kā yəqen ho . Matching n-grams: 1-grams: 14, 2-grams: 9, 3-grams: 6, 4-grams: 4
Manual Reordering Output	یہ بات اہم ہے جو کہ ہم آپ کی شناخت کا یقین ہو yəh bāt əhəm he jo kəh həm āp kī šənāxət kā yəqen ho جب آپ کلیم کریں jəb āp kəlym kəreñ Matching n-grams: 1-grams: 14, 2-grams: 9, 3-grams: 6, 4-grams: 4
Automatic Reordering Output	یہ ضروری ہے کہ ہو سکتا ہے کہ ہم آپ کی شناخت کا yəh ẓərorī he kəh ho səkətā he kəh həm āp kī šənāxət kā یقین ہو جب آپ کلیم کریں - yəqen ho jəb āp kəlem kəreñ . Matching n-grams: 1-grams: 17, 2-grams: 10, 3-grams: 6, 4-grams: 4
Example 3	
Source	the sun concluded that mr . pierce is only part of the problem and a part that ’ gone s .
Reference	سن نے نتیجہ اخذ کیا کہ مسٹر پیئرس صرف مسئلہ کا حصہ ہے sən ne nətejəh əxəż kəyā kəh məsəṭər peīrəs şərəf məsəeləh kā ḥəşəh he - اور ایک حصہ جو الگ ہو چکا ہے - - or ek ḥəşəh jo əlag ho čəkā he . Matching n-grams: 1-grams: 22, 2-grams: 12, 3-grams: 7, 4-grams: 4
Manual Reordering Output	یہ سن نے نتیجہ اخذ کیا کہ مسٹر - - - صرف مسئلہ yəh sən ne nətejəh əxəż kəyā kəh məsəṭər . . . şərəf məsəeləh اور ایک حصہ کا حصہ ہے کہ کی گئی ہے - - or ek ḥəşəh kā ḥəşəh he kəh kī gəeī he . . Matching n-grams: 1-grams: 22, 2-grams: 12, 3-grams: 7, 4-grams: 4
Automatic Reordering Output	یہ سن نے نتیجہ اخذ کیا ہے کہ مسٹر - صرف مسئلے کا yəh sən ne nətejəh əxəż kəyā he kəh məsəṭər . şərəf məsəele kā حصہ ہے ، اور ایک حصہ چلے گئے کہ کی ہے - ḥəşəh he , or ek ḥəşəh čəle gəe kəh kī he . Matching n-grams: 1-grams: 19, 2-grams: 10, 3-grams: 5, 4-grams: 2

Table 5.7: Sample output of reordering system

6. Conclusions and Future Work

This thesis focused on different preprocessing techniques for statistical machine translation, the emphasis being on using only source side processing to handle low resource target side languages (Urdu in this study). Two main research objectives were formulated: 1) To find improvements in word alignment by adding artificial markers in English language. 2) How to learn source side reordering rules automatically from the given parallel corpus to handle long distance reordering for English-to-Urdu statistical machine translation? In the pursuit, we proposed a method to add Urdu style post markers in English to investigate our first goal, and we developed a source side reordering technique using tree-to-string alignment method to answer our second goal.

All the proposed methods have been implemented and compared with a baseline system. The approach for adding artificial markers in English provided promising theoretical basis and results for word alignments also showed interesting improvements (Section 4.3.2). However, the automatic evaluation score for the statistical machine translation system with added artificial markers has been shown to be limited, in spite of the mentioned improvements in the alignments. Random analysis of the output showed that there are improvements in translation quality in terms of human understanding, but the single reference evaluation for BLEU and NIST metrics is unable to reflect the accuracy of the system. This raised a question mark on the evaluation technique used, and a further analysis in this regard is required.

Another factor in explaining the undesirable results for the marker approach is the use of very basic and limited set of rules to add markers. This resulted in generation of extra markers in the output, which not only penalized the BLEU score but also lowered the n-gram counts.

The automatic reordering technique showed significant improvement over the baseline for EMILLE corpus and also showed comparable results for the EMILLE corpus with the manual reordering rules.

6.1 Future Work

The thesis opens up various possibilities for future research. The main aim would be of course to improve the automatic learning algorithm to extract more reordering patterns. A possible improvement is to generate the manually aligned corpus intelligently to cover a wide range of grammatical structure. Probabilistic

methods can also be utilized to learn and apply the reordering rules. Dependency based reordering patterns are also an interesting direction for further research.

The addition of artificial markers also requires further analysis. More linguistically pruned rules are required to add markers. The rules can also be extracted from parallel corpus. It would be also potentially beneficial to test the markers approach with reordering.

Another very notable opportunity for further research would be the experimentation with different language pairs. Specially the south asian languages like hindi, punjabi, etc. are sister languages to Urdu and this research can be directly applied to these languages.

Proper evaluation methods also raised some doubts, it is therefore required a proper human evaluation to test the significance of all the methods. Adopting other evaluation metrics can also be a direction of further research.

A. Manual Reordering Rules

Source Rule	Target Order
S → ADVP VP	→ reverse
S → ADVP VP NP	→ 2 0 1
SBAR → WHNP S	→ nochange
SINV → ADVP VP NP	→ 2 0 1
SINV → MD NP VP	→ 1 2 0
SQ → MD NP VP	→ 1 2 0
SQ → VB* RB NP VP	→ nochange
NP → NP PP	→ reverse
NP → NP PP .	→ 1 0 2
NP → DT NN RB	→ 2 0 1
NP → DT NN S	→ 2 0 1
NP → NP NN NNS	→ 2 1 0
NP → NP PRN PP	→ 2 0 1
NP → NP LRB PP RRB	→ 0 3 2 1
NP → RB JJ PRN	→ 2 0 1
NP → default	→ nochange
VP → TO VP	→ reverse
VP → VB* NP	→ reverse
VP → VB* PP	→ reverse
VP → VB* NP UCP	→ 1 0 2
VP → VB* ADJP	→ reverse
VP → VB* ADVP	→ reverse
VP → VB* ADVP ADVP	→ 1 2 0
VP → VB* S	→ nochange
VP → VB* : S	→ nochange
VP → VB* S : S	→ nochange
VP → VB* : SQ	→ nochange
VP → VB* PP : SQ	→ 1 0 2 3
VP → VB* ADJP	→ nochange
VP → VB* ADJP , ADJP	→ 1 2 3 0
VP → VB* ADVP VP	→ 1 2 0
VP → ADVP VB* NP	→ 0 2 1
VP → ADVP VB* PP	→ 2 0 1
VP → ADVP VB* PP SBAR	→ 2 0 1 3
VP → VB* RB VP	→ 2 0 1
VP → MD RB VP	→ 2 0 1

VP → MD ADVP VP	→ 1 2 0
VP → MD , ADVP , VP	→ nochange
VP → MD RB ADVP VP	→ 2 3 0 1
VP → MD RB PP VP	→ 2 3 0 1
VP → VB* NP PP	→ reverse
VP → VB* PP NP S	→ 1 2 0 3
VP → VB* NP PP PP	→ 1 2 3 0
VP → VB* PP PP	→ 1 2 0
VP → VB* PP PP , SBAR	→ 1 2 0 3 4
VP → VB* NP NP	→ 1 2 0
VP → VP CC VP	→ nochange
VP → ADVP VP CC VP	→ nochange
VP → VP , CC VP	→ nochange
VP → VP , VP CC VP	→ nochange
VP → VP CC VP CC VP	→ nochange
VP → VP , CC VP PP	→ nochange
VP → , CC VP :	→ nochange
VP → VB* CC VB* NP	→ 0 1 3 2
VP → VP , NP	→ 2 1 0
VP → VB* , PP , S	→ 1 2 3 0 4
VP → VB* ADJP S	→ nochange
VP → VB* ADJP S SBAR PP	→ 1 0 2 3 4
VP → VB* ADVP PP	→ 1 2 0
VP → VB* SBAR	→ nochange
VP → VB* PRN	→ nochange
VP → VB* PRN SBAR	→ nochange
VP → VB* PP SBAR	→ 1 0 2
VP → VB* RB ADJP SBAR	→ 2 1 0 3
VP → VB* NP ADVP	→ 1 2 0
VP → VB* NP ADVP SBAR	→ 1 2 0 3
VP → VB* NP ADVP PP	→ 1 3 2 0
VP → VB* NP ADVP PP SBAR	→ 1 3 2 0 4
VP → ADVP VP NP ADVP	→ 3 2 0 1
VP → VB* PRT NP SBAR	→ nochange
VP → VB* NP PRT PP PP	→ 2 3 4 1 0
VP → VB* PRT NP ADVP , SBAR	→ 3 2 1 0 4 5
VP → VB* ADVP ADJP S	→ 1 2 0 3
VP → RB VP CC ADVP VP	→ nochange
VP → VB* NP PP , CC VB* NP	→ 2 1 0 3 4 6 5
VP → default	→ reverse

PP	→ IN NP	→ reverse
PP	→ TO NP	→ reverse
PP	→ IN S	→ reverse
ADJP	→ NP JJ	→ nochange
ADJP	→ JJ NP	→ nochange
ADJP	→ JJ PP	→ nochange
ADJP	→ JJ SBAR	→ nochange
ADJP	→ JJ *	→ nochange
ADJP	→ RB VB* S	→ nochange
ADJP	→ VB* RB ADJP	→ nochange
ADJP	→ JJ CC JJ	→ nochange
ADJP	→ JJ , JJ CC JJ	→ nochange
ADJP	→ RBS JJ	→ nochange
ADJP	→ JJ RB	→ nochange
ADJP	→ JJ S	→ nochange
ADJP	→ ADVP PP	→ nochange
ADJP	→ default	→ reverse
ADVP	→ ADVP PP	→ reverse
ADVP	→ RBR IN RB	→ reverse
PRN	→ LRB * RRB	→ reverse
WHPP	→ IN WHNP	→ reverse

B. Automatically Extracted Reordering Rules

Source Rule	Target Order
NP → DT NN NN	→ 0 1 2
ADJP → RB JJ	→ 0 1
VP → MD VP	→ 1 0
VP → VBZ VP	→ 1 0
VP → VBD NP PP	→ 1 2 0
NP → JJ NNS	→ 0 1
VP → VB NP PP	→ 1 2 0
VP → VBD S	→ 1 0
NP → NP VP	→ 0 1
NP → NN NN	→ 0 1
QP → $ CD CD	→ 1 0 2
NP → NN NNS	→ 0 1
NP → DT NNS	→ 0 1
SBAR → WHNP S	→ 0 1
PP → TO NP	→ 0 1
NP → DT NN POS	→ 0 1 2
VP → TO VP	→ 0 1
VP → VBG NP	→ 1 0
NP → CD NN	→ 1 0
NP → NP CC NP	→ 0 1 2
VP → VBZ NP	→ 1 0
S → NP VP	→ 0 1
VP → VBN PP	→ 1 0
NP → DT JJ NN NN	→ 0 1 2 3
VP → VB NP	→ 0 1
VP → VP CC VP	→ 0 2 1
NP → NP PP	→ 0 1
S → S , NP VP .	→ 0 1 2 3 4
VP → VBP VP	→ 1 0
VP → VBN NP	→ 0 1
SBAR → IN S	→ 0 1
NP → CD NNS	→ 0 1
NP → NP , NP	→ 0 1 2

NP → NP NP	→ 0 1
NP → NP NN	→ 0 1
VP → VBD VP	→ 1 0
S → NP VP .	→ 0 1 2
NP → NP SBAR	→ 0 1
NP → JJ NN NNS	→ 0 1 2
NP → NP . NP .	→ 0 1 2 3
NP → DT JJ NN	→ 0 1 2
NP → DT JJ NNS	→ 0 1 2
VP → VBD SBAR	→ 0 1
NP → NN POS	→ 0 1
NP → NP : NP	→ 0 1 2
VP → VBP NP	→ 1 0
NP → JJ NN	→ 0 1
NP → DT NN	→ 0 1
NP → NN NN NN	→ 0 1 2
VP → VBD NP	→ 1 0
PP → IN NP	→ 0 1
NP → NP , NP ,	→ 0 1 2 3
VP → VBD PP PP	→ 1 2 0
S → PP , NP VP .	→ 0 1 2 3 4

References

T. Ahmed. Ablative, Sociative and Instrument Markers in Urdu, Punjabi and Sindhi. In *Conference on Language and Technology (CLT'07)*, Peshawar, Aug. 2007.

Y. Al-Onaizan and K. Papineni. Distortion Models for Statistical Machine Translation. In *Proceeding ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 529–536. Association for Computational Linguistics, 2006.

A. R. Aminzadeh and W. Shen. Low-resource speech translation of Urdu to English using semi-supervised part-of-speech tagging and transliteration. *Spoken Language Technology*, 2008.

N. Ata, B. Jawaid, and A. Kamran. Rule Based English to Urdu Machine Translation. In *Conference on Language and Technology (CLT'07)*, Peshawar, Aug. 2007. Department of Computer Science, University of karachi.

K. Baker, S. Bethard, M. Bloodgood, R. Brown, C. Callison-Burch, G. Coppersmith, B. Dorr, W. Filardo, K. Giles, A. Irvine, M. Kayser, L. Levin, J. Martineau, J. Mayfield, S. Miller, A. Phillips, A. Philpot, C. Piatko, L. Schwartz, and D. Zajic. Semantically Informed Machine Translation (SIMT). *Summer Camp for Applied Language Exploration*, Nov. 2009.

P. Baker, A. Hardie, T. McEnery, H. Cunningham, and R. Gaizauskas. EMILLE, a 67-Million Word Corpus of Indic Languages: Data Collection, Mark-up and Harmonisation. In *Proceedings of LREC*, pages 819–827, Lancaster, 2002.

J. A. Bilmes and K. Kirchhoff. Factored Language Models and Generalized Parallel Backoff. In *Proceedings of HLT-NAACL*, Edmonton, May 2003.

A. Birch. *Reordering Metrics for Statistical Machine Translation.* PhD thesis, School of Informatics, University of Edinburgh, 2011.

O. Bojar, M. Ercegovčevic, M. Popel, and O. F. Zaidan. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, July 2011.

P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June 1990.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18, 1992.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics - Special issue on using large corpora*, 19(2), June 1993.

M. Carpuat and D. Wu. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 61–72, Prague, June 2007.

E. Charniak, K. Knight, and K. Yamada. Syntax-based Language Models for Statistical Machine Translation. In *Proceedings of Machine Translation Summit IX*, New Orleans, Sept. 2003.

D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

N. Chomsky. Quine's empirical assumptions. *Synthese*, 19:53–68, Dec. 1968.

M. Collins, P. Koehn, and I. Kučerová. Clause restructuring for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, June 2005.

M.-C. De Marneffe, B. MacCartney, and C. B. Manning. Generating Type Dependency Parses from Phrase Structure Parses. In *Proceedings of the fifth International Conference on Language Resources and Evaluation*, Genoa, May 2006.

Factored Language Models Tutorial, Department of EE, University of Washington, Feb. 2008. Department of EE, University of Washington.

G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLT '02 Proceedings of the second international conference on Human Language Technology Research*, 2002.

M. Federico, N. Bertoldi, and M. Cettolo. IRSTLM: An Open Source Toolkit for Handling Large Scale Language Models. *9th Annual Conference of the International Speech Communication Association*, 2008.

G. Foster and R. Kuhn. Stabilizing Minimum Error Rate Training. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 242–249, Athens, Mar. 2009.

G. Foster, S. Gandrabur, and P. Langlais. Statistical Machine Translation: Rapid Development with Limited Resources. In *Proceedings of Machine Translation Summit IX*, 2003.

S. Hussain. Resources for Urdu Language Processing. *The 6th Workshop on Asian Language Resources*, 2008.

J. Hutchins. Machine translation: A concise history. *Computer aided translation: Theory and practice, ed. Chan Sin Wai. Chinese University of Hong Kong*, 2007.

Z. Islam. *English to Bangla Phrase-Based Statistical Machine Translation*. PhD thesis, Erasmus Mundus European Masters Program in Language and Communication Technologies, Aug. 2009.

B. Jawaid. Statistical Machine Translation between Languages with Significant Word Order Difference. Master's thesis, UFAL, Prague, Sept. 2010.

B. Jawaid and D. Zeman. Word-Order Issues in English-to-Urdu Statistical Machine Translation . *The Prague Bulletin of Mathematical Linguistics*, pages 87–106, Apr. 2011.

D. Klein and C. Manning. The stanford parser: A statistical parser. In *International Conference, ASEA*, page 240. Springer-Verlag New York Inc, 2009.

K. Knight and D. Marcu. MACHINE TRANSLATION IN THE YEAR 2004. In *(ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, pages 965–968. IEEE, 2005.

P. Koehn. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP*, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2004. Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

P. Koehn. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876, Prague, 2007.

P. Koehn, F. J. Och, and D. Marcu. Statistical Phrase-Based Translation. *Proceedings of HLT-NAACL*, pages 48–54, May 2003.

P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, and M. Osborne. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings for IWSLT*, 2005.

P. Koehn, M. Federico, W. Shen, N. Bertoldi, O. Bojar, C. Callison-Burch, B. Cowan, C. Dyer, H. Hoang, R. Zens, A. Constantin, C. C. Moran, and E. Herbst. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. Technical report, Sept. 2007a.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, and N. Bertoldi. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180, Prague, June 2007b.

A. Lavie, S. Vogel, and L. Levin. Experiments with a Hindi-to-English Transfer-based MT System under a Miserly Data Scenario. *ACM Transaction on Computational Logic*, 5, Sept. 2003.

G. Leusch. *Evaluation Measures in Machine Translation*. PhD thesis, RWTH Aachen, July 2005.

Y. Liu, Q. Liu, and S. Lin. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 609–616, Sydney, July 2006. Association for Computational Linguistics.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: the penn treebank. *Association for Computational Linguistics*, 19(2):313–330, June 1993.

M. Nagao. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In *Proceedings of the International NATO Symposium on Artificial and Human Intelligence*, New York, USA, 1984.

F. J. Och. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. PhD thesis, Department of Computer Science, RWTH, Aachen, Oct. 2002.

F. J. Och. Minimum Error Rate Training in Statistical Machine Translation. In *ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167. Association for Computational Linguistics, 2003.

F. J. Och and H. Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1), Mar. 2003.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, July 2001. Association for Computational Linguistics.

C. Quirk, A. Menezes, and C. Cherry. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 271–279, 2005.

A. Ramanathan, P. Bhattacharyya, J. Hegde, R. M. Shah, and S. M. Simple syntactic and morphological processing can help english-hindi statistical machine translation. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, 2008.

A. Ramanathan, H. Choudhary, A. Ghosh, and P. Bhattacharyya. Case markers and morphology: addressing the crux of the fluency problem in English-Hindi SMT. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 800–808, Suntec, Aug. 2009. IIT Bombay, Association for Computational Linguistics.

A. Spencer. Case in Hindi. In *Proceedings of LFG Conference*, 2005.

A. Stolcke. SRILM-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language*, 2002.

C. Tillmann. A unigram orientation model for statistical machine translation. In *HLT-NAACL-Short '04: Proceedings of HLT-NAACL 2004: Short Papers*. Association for Computational Linguistics, May 2004.

M. Turchi, T. De Bie, and N. Cristianini. Learning performance of a machine translation system: a statistical and computational analysis. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 35–43, Columbus, June 2008.

K. Visweswariah, J. Navratil, J. Sorensen, V. Chenthamarakshan, and N. Kambhatla. Syntax Based Reordering with Automatically Derived Rules for Improved Statistical Machine Translation. *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1119–1127, Aug. 2010.

K. Visweswariah, R. Rajkumar, A. Gandhe, A. Ramanathan, and J. Navratil. A Word Reordering Model for Improved Machine Translation. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 486–496, July 2011.

S. Vogel, H. Ney, and C. Tillmann. HMM-Based Word Alignment in Statistical Translation. In *Proceedings of the 16th conference on Computational linguistics*, 1996.

C. Wang, M. Collins, and P. Koehn. Chinese Syntactic Reordering for Statistical Machine Translation. In *Proceedings of the EMNLP-CoNLL*, 2007.

F. Xia and M. McCord. Improving a statistical MT system with automatically learned rewrite patterns. In *COLING '04 Proceedings of the 20th international conference on Computational Linguistics*, 2004.

D. Xiong, Q. Liu, and S. Lin. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 521–528, Sydney, July 2006.

P. Xu, J. Kang, M. Ringgaard, and F. Och. Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages 245–253, Boulder, June 2009. Google Inc.

K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, Stroudsburg, 2001.

R. Yeniterzi and K. Oflazer. Syntax-to-Morphology Mapping in Factored Phrase-Based Statistical Machine Translation from English to Turkish. In *Proceedings*

of the 48th Annual Meeting of the Association for Computational Linguistics, pages 454–464, Uppsala, July 2010. Association for Computational Linguistics.

D. Zhang, M. Li, C.-H. Li, and M. Zhou. Phrase Reordering Model Integrating Syntactic Knowledge for SMT. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 533–540, Prague, June 2007.

Y. Zhang. *Chinese-English Statistical Machine Translation by Parsing*. PhD thesis, Mansfield College, University of Oxford, 2006.

Y. Zhang, S. Vogel, and A. Waibel. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system. In *Proceedings of LREC*, 2004.

A. Zollmann, A. Venugopal, F. J. Och, and J. Ponte. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1145–1152, Manchester,, Aug. 2008.