Hindi to English Machine Translation

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science by Research
in
Computer Science and Engineering

by

Kunal Sachdeva
200902046
kunal.sachdeva@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500032, INDIA
February 2016

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled *"Hindi To English Machine Translation"* by *Kunal Sachdeva*, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Dipti Misra Sharma

To parents and friends ...

Acknowledgments

I would like to express my gratitude to my adviser *Prof. Dipti Misra Sharma* for showing the faith and confidence in me and guiding me during the course of my work.

I owe my thanks to a number of people, each of whom contributed in their own way towards the completion of this work. A special mention to Sambhav Jain and Riyaz Ahmad Bhat who inspired me a lot and gave me due direction with their experience. This work would have not been possible without the assistance from Karan Singla who helped me gain more knowledge in the field of Machine Translation.

I am grateful to my lab-mates Ankush, Kshitij, Rishabh, Rahul, Arpita, Urmi, Naman, Himanshu, Vandan and Maaz for creating such a wonderful working environment in LTRC. Apart from the lab-mates I would like to thank Rajesh, Data and Karthik for all the fun during last two years. This would have not been possible without the support of my wing-mates and being with me at all times.

I also thank, Rambabu, Srinivas Rao, Satish, Kumarswamy and Lakshmi Narayan for making administrative matters run smoothly.

Abstract

Machine Translation (MT) is a task in Natural Language Processing (NLP), where the automatic systems are used to translate the text from one language to another while preserving the meaning of source language. In this work, we provide our efforts in developing a rule-based translation system on the Analyze-Transfer-Generate paradigm which employs morphological and syntactic analysis of source language. We utilized shallow parser for Hindi language along with dependency parse labels for syntactic analysis of Hindi language, developed modules for transfer of Hindi to English and generation of English language. Due to wide difference in word order of the two languages (Hindi following SOV and English SVO word order), a lot of re-ordering rules need to be crafted to capture the irregularity of the language pair. As a result of drawbacks of the aforementioned approach, we shifted to statistical methods for developing a system. A wide variety of machine translation approaches have been developed in past years. As each model has its pros and cons, we propose an approach where we try to capture the advantages of each system, thereby developing a better MT system. We then incorporate semantic information in phrase-based machine translation using monolingual corpus where the system learns semantically meaningful representations.

Recent studies in machine translation support the fact that multi-model systems perform better than the individual models. In this thesis, we describe a Hindi to English statistical machine translation system and improve over the baselines using multiple translation models. We work on MOSES which is a free statistical machine translation framework, which allows automatically training translation model using parallel corpus. MOSES provides support for multiple algorithms for training a model. We propose an approach for computing the quality score for each translation by using automatic evaluation metric as our quality score. The computed quality score is used for guided selection among the translations from multiple models, thereby providing a better system. We have used support vector regression to train a model using syntactic,textual and linguistic features extracted from the source and target translation with evaluation metric as our regression output.

Quality Estimation of Machine Translation is a task where the system tries to predict the quality of output on the basis of features extracted from source and target languages. The system dynamically(run time) computes a quality score corresponding to each translation, which is the measurement for the correctness of the output. Different from MT evaluation, quality estimation systems do not rely on reference translations and are generally addressed using machine learning techniques. The approach offers a great advantage to the readers of target language as well as for professional translators.

The current phrase based machine translation model do not account the semantic property of the language while generating the translations. As a part of third work, we propose a methodology where we learn bilingual embeddings across two languages, thereby using it to compute the semantic similarities between source and target phrases. We compute the vector embeddings of two languages using large unlabeled monolingual corpus and learns linear transformation between vector spaces of language. We then use this transformation matrix to transform one of the language space to another and compute similarity between phrases using vector composition. The proposed approach shows significant improvement in BLEU score as it is able to capture syntactic and semantic property in its phrases.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The tremendous increase in industrial growth over the past decades has a huge impact on the global Machine Translation market which enables content to be available in all regional languages across the globe. Machine Translation is a sub-field of Computational linguistic which enables automatic translation of sentences or documents from one language to another. The first few years of research in this field was dedicated to developing rule-based systems using bilingual dictionary and some hand coded rules. The gradual transition from rule based to statistical trend was observed in 1980's when increased computational power was available. The statistical models rely on parallel data to build a statistical model which is more generic and cost effective as compared to rule based systems. According to global MT market analysis forecast the growth of Machine Translation by 25 % from 2014-2019.

A major aspect of MT development is to assess the quality of output, therefore the major research in Machine translation apart from developing basic systems revolves around quality estimation, domain adaptation, system combination, evaluation and post editing. Major workshops have been conducted in top conferences which emphasize on the broad aspects of MT but still this problem is not completely solved. Quality estimation is a major challenge in any field of study. Use of automatic systems for real world applications has always been a topic of debate in different communities. In the sub-field of MT, the system predicts the quality of translation from the information available in source sentence and its corresponding translation which then decides on the amount of post-editing needed to use the translation in a real application.

In the past decades multiple approaches for solving complex MT problems have been suggested with each approach having its pros and cons. The system combination approach combines the positives from the corresponding translation of each system thereby performing better than each of the individual systems.

A system trained on one domain fails to meet the desired standards of output when used on another domain due to the difference in syntax and lack of domain vocabulary. A new system can be trained on new domain, but this requires sufficient amount of data to be available for the new domain. Due to increasing number of domains this task is cost inefficient as preparing new resources for any domain requires large amounts of effort. This issue is resolved by domain adaptation wherein the research

community proposes a generalized method for adapting the existing systems to a new domain with limited amount of effort.

The evaluation is a key step of any system and the researchers need to come up with the right and standard metrics to compare our system against. A couple of evaluation metrics have been proposed in past decades viz. BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), NIST (Doddington, 2002), WER (word error rate) and TER (Snover et al., 2006). The BLEU metric is the most widely used but their is another MT community which prefer METEOR due to its high correlation with human judgement. In the current era, the researchers have shifted the major focus to inducing semantic properties in machine translation and use of complex neural networks to solve this complex problem.

English and Hindi are respectively reported to be the 3rd and 4th largest spoken languages[1] in the world. The fact that one of these language is known by 10% (approx.) of the world population, makes it an ideal pair for translation studies. In past most of the work (Naskar and Bandyopadhyay, 2005) has been focused on English to Indian language translation systems. However very few works (1-2 systems) have laid their emphasis on Hindi to English translation systems, which also finds its use in real world problems. The major use can be seen in areas of news articles translations and the translation of documents in judicial systems. The hearings and document works in regional law courts are performed in local language, which creates an issue when the case moves to high court where English is the official language. The time taken by the human translators to translate each document in local language to English postpones the justice and weakens the judicial system. Due to lack of quality Hindi-English systems and its use in real world becomes our major motivation for the thesis.

I. Key Contributions of the Thesis

The following are the three key contributions of this thesis :-

1.0.1 Rule-Based system

We explore the use of rule-based approach for Hindi-English machine translation where we integrate multiple modules on the Analyze-Transfer-Generate paradigm in a pipeline architecture. The developed system works well on the provided rules, but fails to perform on all possible outputs due to wide difference in word order of source and target language, which requires several linguistic rules to be formulated. Since developing the rules is costly we shifted our focus of research to statistical methods which requires more of computation knowledge rather than linguistic.

[1] http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

1.0.2 Use of Multi-model in Machine Translation

The key task in this work was the use of multiple models in a single Hindi-English statistical machine translation system. We work with *MOSES* toolkit which provides support for Phrase-based, Hierarchical , factored and tree based models. For our work we have explored phrase based and hierarchical models due to other models giving unsatisfactory results (6.3 BLEU points). The system dynamically selects the better translation of the two candidate translation using prediction strategies. We have worked with SVM regression to predict the translation using linguistic and textual features with automatic evaluation score as the corresponding regression value. This methodology shows significant improvement over the baseline with an increase of 0.64 BLEU score. We compare our system against the human judgment and also with the commercially available systems which provide the support for Hindi-English language pair.

1.0.3 Distributional semantics in the context of Statistical Machine Translation

We explore the use of phrase similarity in SMT with the help of distributional semantic models. We explore methods for computing word embeddings of Hindi and English from large monolingual corpus and adopt regression method for transforming one language vector space into other language vector space. These transformed vector space capture the bilingual semantics which is used for scoring semantic similarity between source and target phrases. The proposed method is quite generic and requires bilingual dictionary for training a model, which is the most easily available resource for any language pair. We have experimented with Hindi-English and English-Hindi language pair with varying dimensionality and context window across two state of the art vector representation learning models. The experimental setup shows the impact of increasing vector dimensionality on the word-similarity and MT systems.

1.0.4 Reducing Data Sparsity in SMT

The major contribution of this work is reducing the data sparsity in the context of English-Hindi SMT system. Hindi being a morphologically rich language suffers a further setback due to lack of large parallel corpora. Firstly we explore the use of recurrent neural network (RNN) based language model (RNNLM) (Mikolov et al., 2010) to re-rank a list of n-best translations. We integrated linguistic features [2] in the standard RNNLM model which further boosted the performance. As a second step we use WordNet to extend the coverage of source words by incorporating synonyms using two different approaches. The sense incorporated phrase table behaves as a factored translation models (Koehn and Hoang, 2007) which helps in reducing out of vocabulary (OOV) words in output translations. We are not presenting this work in detail as a part of this thesis and can be further referred at (Singla et al., 2014).

[2]We performed experiments with part of speech, lemma, number case with one at at a time as well as combined features.

II. Outline

The thesis is organized into five chapters and a brief outline of each is as follows :-

Chapter 2: In this chapter we discuss the background study required for the work. We study the three basic approaches of developing a machine translation system along with the pros and cons of each approach. Since most of our work revolves around statistical approach, we study the architecture of *MOSES (Koehn et al., 2007a)* toolkit. We also examine the automatic and human evaluation methodologies which is the most important step of any MT system.

Chapter 3: This chapter presents our efforts in developing a Rule-based Hindi-English MT system on the Analyze-Transfer-Generate paradigm. We explain the importance of each module in a pipeline architecture along with information flow across each module. We then discuss the challenges and drawbacks of the system, which is the main motivation behind shifting to statistical methods.

Chapter 4: The chapter presents our methodologies for developing the multi-model SMT system where we use multiple strategies for translation and choose one translation as the final output. We first explain our efforts in developing the baseline systems Hindi-English language pair. We then explain the detail method of selecting the best translation by employing the key idea behind quality estimation of MT output. We also compare our system with two available, most widely used commercial MT systems.

Chapter 5: In this chapter we explore the use of distributional semantics in statistical machine translation and employ the use of partial least squares regression to model the bilingual word embeddings. We present the methodology and results of incorporating semantic similarity between phrase as a feature while decoding, which helps in learning semantically and syntactically better phrases.

Chapter 6: This chapter concludes the thesis and discusses possible future work.

Chapter 2

Machine Translation

Introduction

Machine translation (MT) is an application of Natural Language Processing where automatic systems are used for translating one natural text to another restoring the meaning of source text in target language. Translation is not a word by word substitution, but requires extensive knowledge of syntax, grammar and semantics of source and target language. An automatic system processes an input sentence by extracting the relevant information needed for translation and outputs the target translation on the basis of pre defined steps. The work on machine translation started in 1950's where a group of researchers were involved in translation of sentences from Russian to English and claimed that within three to five years, this would be a solved problem. However, till date state of the art machine translation systems cannot be used with high confidence in real word systems.

Machine Translation finds its use for both human translators, who get a rough version of translation and the end users, who use these systems as a part of bigger system. Due to dearth of quality systems, an extra effort is required if they are to be used in high-end systems. Computer aided translation (CAT) and Human aided machine translation (HAMT) are examples of post-editing efforts where we apply these approach to improve the quality of systems output to meet the desired requirements. The former approach uses automatic tools like spell checker, grammar checker etc. to improve the quality, while the latter approach uses human effort.

Over the next sections we define various approaches used in machine translation along with their pros and cons.

2.1 Approaches for Machine Translation

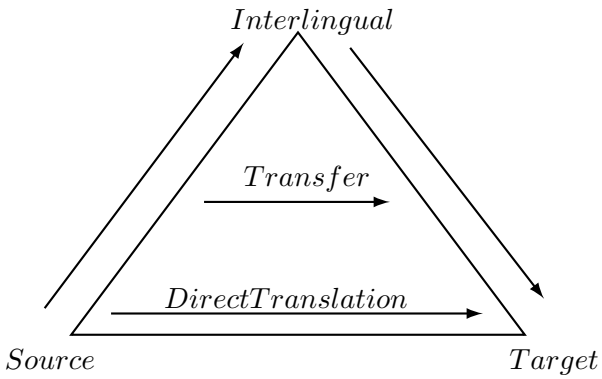

Figure 2.1: Machine Translation Pyramid

2.1.1 Rule-based

The MT pyramid[1] (figure 2.1) proposed by Bernard Vauquois depicts the depth of analysis of source language along with its representation between source and target language. Direct transfer approach offers the shallowest source language analysis followed by transfer and interlingual based approaches. All of these approaches are categorized under Rule based Machine Translation (RBMT) system. Rule-based MT rely on linguistic rules and bilingual dictionaries for each language pair wherein the rules capture the syntactic and semantic properties of a language. Rules are written with linguistic knowledge gathered from linguists. We explain below the three mentioned methods in detail:

2.1.1.1 Direct Translation

This is most simplest and unconventional method of machine translation where words are just substituted by a dictionary look-up. Words may or may not be substituted with the help of morphological analysis. This method is generally followed at phrase level rather than at sentence level as it fails to consider the semantic meaning of the text. This methodology fails to address non-compositionality and inflectional agreement and creating these dictionaries for all language pairs to get high coverage is an expensive task.

2.1.1.2 Transfer based

In this type of MT system, the source language is first analyzed syntactically (or semantically) and grammatically to obtain an intermediate representation. Transfer rules are then applied to this represen-

[1] https://en.wikipedia.org/wiki/Machine_translation

tation to convert source structure into target structure and finally rules are applied to generate target text as needed. Lexical transfer rules are also applied along with grammatical rules. Transfer rules cover the structural divergence between the two languages. The level of linguistic analysis can vary depending upon the family of language pair. For languages of same family or of the same type (Spanish-Catalan) syntactic analysis is more suitable, whereas for more distant language pairs (Spanish-English) a deeper transfer or semantic representations are needed.

Sampark, an Indian language to Indian language machine translation system is a classic example of this approach, which is built on the Analyze-Transfer-Generate paradigm.

2.1.1.3 Interlingual

This approach offers the maximum depth analysis among the three previously mentioned approaches. In this approach, the source language is converted into a meta-language i.e. an abstract language-independent representation which represents the semantic meaning of the sentence. The target language is then generated from this representation with less efforts as compared to transfer approach.

The above mentioned rule based approaches do not require bilingual texts and are fairly domain independent. The hand crafted rules gives the developers a total control over the system. However, writing these rules is an expensive task and requires human efforts. The rules are language-pair specific and cannot be easily adapted to other language pairs.

2.1.2 Statistical

Statistical Machine Translation (SMT) uses a statistical model to generate translations and is based on the analysis of bilingual corpus. The most important benefit of SMT over Rule based Machine RBMT is that it does not require manual development of linguistic rules, which is quite costly. We have listed below some of the important approaches which uses statistical methods:

2.1.2.1 Word-based

Word based models have been originated from the works on SMT by IBM, commonly known as the IBM models, where word was considered as a fundamental unit of translation. The simplest models use word mappings or alignments from one language to another which are extracted from corpus statistics. Most words have multiple translations but only certain word is more apt in a given context. Lexical translation probability is estimated from the count of each word and applying maximum likelihood estimation upon it which maximizes the likelihood of the data. The basic word based model or IBM model 1 had many flaws in terms of reordering, as well as adding and dropping words. The later models incorporated the advance features of reordering, alignment and fertility.

2.1.2.2 Phrase-based

Phrase based models are the most widely used MT approach in which rather than words, small sequence of words are translated at a time. The main motivation behind phrase based machine translation is that one word in some language may correspond to two words in other language making word based models to fail in such cases. In addition, phrase based models provide much better local reordering and capture more context in translation table. However, the word 'phrase' in PB-MT does not correspond to linguistic phrase rather to a group of words. *MOSES (Koehn et al., 2007b)* and *Phrasal (Spence Green and Manning, 2014)* are two most widely used frameworks which provide support for phrase based machine translation.

2.1.2.3 Syntax-based

This method of machine translation is based on translation of syntax rather than a word or phrase. The input bi-text is represented in a tree format (syntax) and parameters are estimated from a syntactic tree. The source and target languages can be independently or together be represented in the form of trees. This approach offers the advantage of learning better reordering models and allows long distance constraints. The major disadvantage with this approach is speed of translation as compared to other statistical approaches and syntax tree generation which itself is another research problem.

2.1.2.4 Hierarchical MT

Hierarchical machine translation is an extension of classic phrase based translation where hierarchical phrases (phrase within phrase) are used. The model uses synchronous context-free grammars and combines the strengths of phrase-based and syntax-based translation. The major work in this field has been done by Chiang (2007) where the grammar is learnt from unlabeled data. This method capitalizes upon the strengths of PB-MT, where not only just learning the reordering of words, re-orderings of phrases is also learnt.

2.1.3 Example-based Machine Translation (EBMT)

In EBMT system, the translation of new input text is gathered from the reuse of already existing translations. It is a translation by analogy, where systems are trained using bilingual parallel corpora . It is a 3 step process, where we first decompose the input text into phrases and find examples that are going to contribute to the translation. In the second step, the fragmented phrases are translated into target language phrases by analogy principle. Recombination is the final step which makes sure that the translated phrases from earlier steps are put together to generate a fluent output.

2.1.4 Hybrid MT

It is a method of machine translation where multiple MT approaches are used in a single system. The main motivation behind the use of hybrid approach was from the failure of any single system to achieve satisfactory results. Hybrid solutions tend to combine the advantages of individual approaches to achieve an overall better translation. Though hybrid systems combine the benefits of individual systems, they also add limitations of each approach. *SYSTRAN*, a machine translation company founded in 1968, implemented a hybrid technology where they coupled their earlier RBMT systems with SMT.

2.2 Evaluation of Machine Translation

Evaluation of machine translation is the most important step for any MT system. This section presents evaluation methods that have been used by the machine translation community.

2.2.1 Human Evaluation

In this method of evaluation, bilingual evaluators who have proficiency in both source and target language are presented with input and output of systems and are asked to rate the output on a predefined scale. Typically two parameters are considered while evaluating any translation:

- **Fluency:** This checks whether the output sentence is grammatically correct or not. The judgment is given on a scale of 1-4 with '1' signifying incomprehensible and '4' signifying flawless output.
- **Adequacy:** This checks whether the meaning conveyed by source sentence has been retained in target sentence or not. Similar to fluency this metric is also judged on a scale of 1-4 with '1' denoting no meaning and '4' denoting all meaning is represented in target translation with respect to source sentence.

This evaluation should be performed by multiple evaluators and on a considerable dataset (atleast 200 source-target translation pairs) to get reliable results. A detailed analysis is performed on the rated translations to get statistically significant results. *Kappa Coefficient* [2] is used to measure the correlation (inter annotator agreement) between the evaluators.

$$K = \frac{p(A) - p(E)}{1 - p(E)}$$

where 'p(A)' is proportion of time that evaluators agree and 'p(E)' is proportion of time evaluators agree by chance.

For comparing outputs among multiple translation, the above mentioned methods are not generally consistent. Instead of judging the sentences on an absolute scale, it is advisable to rank the translations with respect to each other. For our experiments on multi-model SMT (Chapter 4), we have used ranking based approach for human evaluation and showed a high agreement with our system.

[2] *https://en.wikipedia.org/wiki/Cohens_kappa*

2.2.2 BLEU

This is the most widely used method of automatic evaluation where we compute n-gram precision with respect to reference translation. We add *Brevity penalty (BP)* to account for shorter translations i.e. when words are dropped. The main problem with this method of evaluation is that it solely relies on matching n-grams between translation and reference output, and fails to capture word sequences which have similar meaning. On the other hand, it is possible to get a high BLEU score even though the meaning is entirely different with a little movement of n-grams. It is advisable to use multiple reference translations against a system translation to account for all acceptable translation for ambiguous parts.

$$Brevity\ Penalty = min\left(1, \frac{output - length}{reference - length}\right)$$

$$BLEU = Brevity\ Penalty * \prod_{i=1}^{4} precision_i$$

2.2.3 NIST

This evaluation metric is build upon the drawbacks of BLEU metric. BLEU score gives equal weightage to all n-grams, irrespective of its occurrence in corpus i.e. it treats trigram 'in the flight' and 'british airways flight' as giving equal information. NIST scoring method differs from above mentioned method in giving more weights to rare n-grams and a variation in BP .

2.2.4 METEOR

This method of evaluation has shown highest correlation with human judgment. It computes the harmonic mean of uni-gram precision and recall, with recall given more weight in contrast to precision. Unlike above two methods, this scoring technique does not simply check the overlap between two translations but also performs stemming and synonymy matching using WordNet. Words like 'good' and 'well' which were treated differently in previous two approaches are considered to be same in METEOR scoring which adds to score. The precision and recall is calculated as follows, where 'n' is unigram count in both candidate translation and reference translation. n_c and n_r are the unigram counts of candidate and reference translations respectively.

$$P = \frac{n}{n_c}$$

$$R = \frac{n}{n_r}$$

The F-measure is then calculated with recall given more weightage in comparison to precision.

$$F = \frac{10PR}{R + 9P}$$

In order to account for the together occurrence of larger segments, we add a penalty score 'p' as follows:

$$p = 0.5 * \left(\frac{no\,of\,chunks}{Mapped\,Unigrams} \right)$$

The final score for a sentence is computed as mentioned below:

$$METEOR = F * (1 - p)$$

2.3 MOSES

MOSES (Koehn et al., 2007a) is an open-source statistical machine translation framework which provides support for multiple SMT (Phrase-based, Hierarchical, Syntax-based, factored) algorithms. The system requires a parallel corpus of the language pair for training, development and testing procedures along with configurable steps, which the developer wants to follow. Figure 2.2 shows the basic architecture and the information flow of the MOSES. The individual steps in the system can be adjusted depending upon the language pair. We have explained below the key steps used in training a SMT model.

Training

In training procedure, we follow the steps of corpus preparation where we apply basic steps of tokeinzation and truecasing to maintain a consistency within the corpus. The pre-processed training data is passed through an aligner which computes the word and phrase alignments along with probability scores for phrase and word substitution respectively. We have considered GIZA++ (Och and Ney, 2000a) for our experiments which is based on the IBM models and computes the alignment using Expectation-Maximization (EM) algorithm. Phrases are extracted from the phrase alignments and scored to build a phrase table which is the most important step in Phrase based SMT. Four different phrase translation scores are computed:

- Direct phrase translation probability
- Direct lexical weighting
- Inverse phrase translation probability
- Inverse lexical weighting

Language model (LM) checks for the fluency of the target translation by means of probability distribution. The likelihood of any word sequence (n-gram) is computed from an already developed model which in turn computes the likelihood of a sentence. We have used the SRILM (Stolcke and others, 2002a) language model toolkit for our experiments. Large amounts of target language monolingual corpus is required to build a good language model.

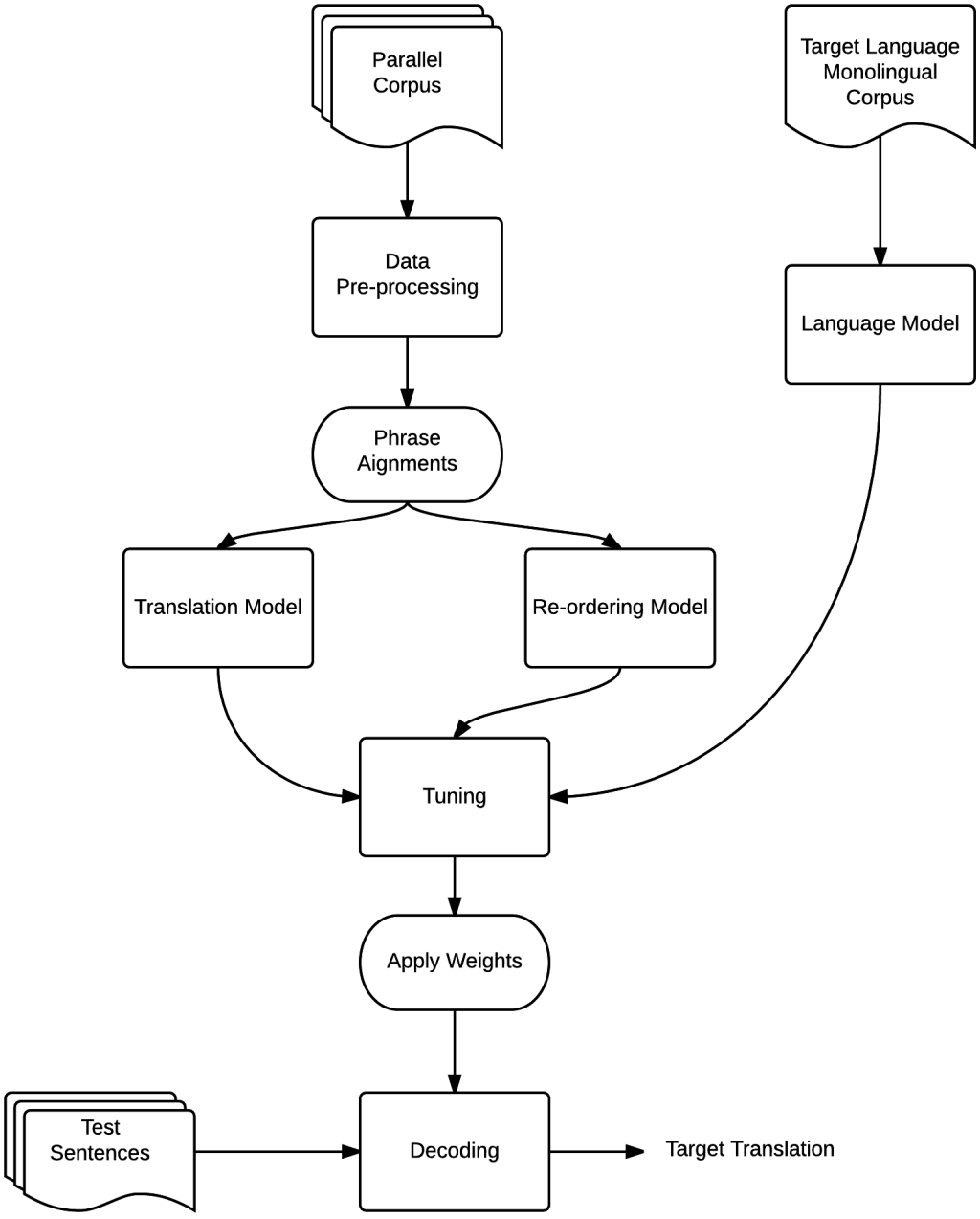

Figure 2.2: The information flow and architecture of MOSES

Tuning

Tuning is a procedure which finds the optimal weights to maximize the translation scores of development corpus. In general three main scores (phrase translation probability, Reordering probability, language modelling probability) are used with subdivision among them to generate the n-best list for any source sentence. The translation quality is calculated using BLEU (automatic scoring) from the best translation and corresponding reference sentence.

Minimum error rate training (MERT) is performed for batch tuning where, generally a sentence is decoded into n-best list and model weights are updated based on output. This procedure continues until a convergence criteria is achieved with these optimal weights used for all future translations.

Evaluation

The testing of system involves decoding a small corpus and comparing the best translation with the reference translation using automatic evaluation metric. The scores for n-best lattices from various modules are combined in a log-linear model with the sentence achieving highest score as system's output. MOSES provides support for use of extra features for improving translations which we have explored in chapter 5.

$$Score = exp\Big(\sum_i w_i x_i\Big)$$

2.4 Conclusion

In this chapter we explained the motivation behind each MT approach along with their pros and cons. We also explained in detail working of a statistical model as the major research work focuses on the improvements of these models. In the next chapter we provide the detailed description of our efforts in developing a rule based system for Hindi-English language pair and conclude with the drawbacks of the system which becomes our major motivation for shifting the focus to statistical methods.

Chapter 3

Rule Based MT System

Introduction

Rule based machine translation system (RBMT) is the classical MT approach started in 1970's which uses large amount of rules and dictionaries to translate text from source to target language. The rules(grammar) are manually crafted by language experts or linguists who map the source structure to target. Rules designed by experts cover the semantic and syntactic regularities of both source and target language, and are often edited and changed to make the translations better. The most important example of RBMT is the *SYSTRAN* which provides support for a wide range of European and Asian languages. Indian language machine translation (ILMT) system developed by consortium of NLP research group from India is also an example of rule-based system.

The main advantages of RBMT is that human skills is invested in development of system, so the automatic translations can be reliable and fairly good enough. These systems do not require bilingual corpus, so language pair which lack bilingual texts, rule based approach makes it possible for developing a MT system. Developers have a complete control over the system which makes sure that any error can be corrected with changes in rules.

There are three types of rule-based machine translation systems:

- **Direct Systems:** This is the most naive approach where translations are generated through simple dictionary look-up with morphological analysis and lemmatization.
- **Transfer Based:** This is the most widely used methods of RBMT systems, where shallow to complete analysis of source and target language is performed.
- **Interlingual Based:** In this method the source side is converted to an abstract language which represents the meaning of source side and target language is generated from this representation.

In this chapter we describe our efforts in the development of a transfer based Hindi-English MT system. The proposed MT system is based on Analyze-Transfer-Generate paradigm same as that of ILMT. First analysis of source language is performed, then a transfer of source language structure to

target language, and finally the target language is generated. The overall system has been divided into many modules. Each module takes the input and performs a small task on it. The developed modules are rule-based, statistical or combination of the two. However apart from performing the core task, a module might carry out some pre-processing or post-processing to represent input or output in correct format[1]. All modules operate on a common representation called Shakti Standard Format (SSF)

3.1 Shakti Standard Format(SSF)

The SSF format is used for representing the analysis of a sentence and maintains a common structure for input specifications to each module. Each module adds some information which is used by other modules down the pipeline.

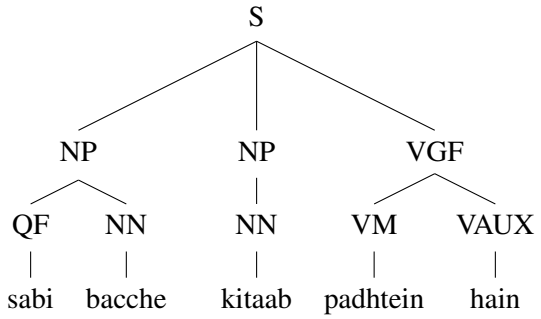

Figure 3.1: Chunk level tree structure of sentence.

The SSF representation of the tree structure is shown in Figure 3.2. The Figure has 4 columns which describe the following property:

- Column 1 indexes the node id for better understanding and readability.
- Column 2 indexes the token or start/end of a word-group (chunk).
- Column 3 indexes the Part of speech tag or the chunk name.
- Column 4 indexes the feature structure (fs) of each word or the fs corresponding to each chunk. Feature structure contains features of each word in sentence, such as root, lexical category, gender, person, case and head word (for chunks only) along with other information. The position is fixed for each attribute and comma is used as a separator. Apart from the basic features more attributes can be added which helps define a word better.

[1]https://researchweb.iiit.ac.in/ rashid.ahmedpg08/ilmt.html

```
<Sentence id="1">
1       ((      NP      <fs af='बच्चा,n,m,pl,3,d,0,0' head="बच्चे">
1.1     सभी     QF      <fs af='सभी,n,m,sg,3,d,0,0' poslcat="NM">
1.2     बच्चे   NN      <fs af='बच्चा,n,m,pl,3,d,0,0' name="बच्चे">
        ))
2       ((      NP      <fs af='किताब,n,f,sg,3,d,0,0' head="किताब">
2.1     किताब   NN      <fs af='किताब,n,f,sg,3,d,0,0' name="किताब">
        ))
3       ((      VGF     <fs af='पढ,v,m,pl,1,,ता_है,wA' vpos="tam1_2" head="पढते">
3.1     पढते    VM      <fs af='पढ,v,m,pl,any,,ता,wA' name="पढते">
3.2     है      VAUX    <fs af='है,v,any,pl,1,,है,hE' poslcat="NM">
3.3     |       SYM     <fs af='',punc,,,,,,'>
        ))
</Sentence>
```

Figure 3.2: SSF representation of a sentence.

3.2 System Architecture

The system is divided into modules, each of which performs some task. Pipeline based architecture is used to manage the flow of information across the modules. The same architecture is followed by ILMT system which is available as a web service. The pipeline architecture facilitates easy debugging and is cost effective. An error produced in the translation can be easily traced back by looking the output of each module. The overall architecture of system is shown in Figure 3.3.

3.3 Description of Individual Modules

We have provided the specification of each module along with its input format and what attributes it modifies before feeding it to next module.

3.3.1 Analysis of Hindi

The analysis modules extracts the linguistic information needed to translate any source language sentence to the target language. We are using rules along with some state of the art statistical models for syntactic analysis of the language. We have adopted the analysis modules directly from the ILMT system as Hindi serves as a source language for many language pairs available in ILMT system. The various modules deployed in analysis pipeline are explained below:

Tokenizer

Tokenizer breaks the sentence into words, punctuation marks and other symbols also called tokens. Tokens are separated by white-space characters, line breaks or punctuation markers. Special tokens are handled separately to avoid wrong tokenizations.

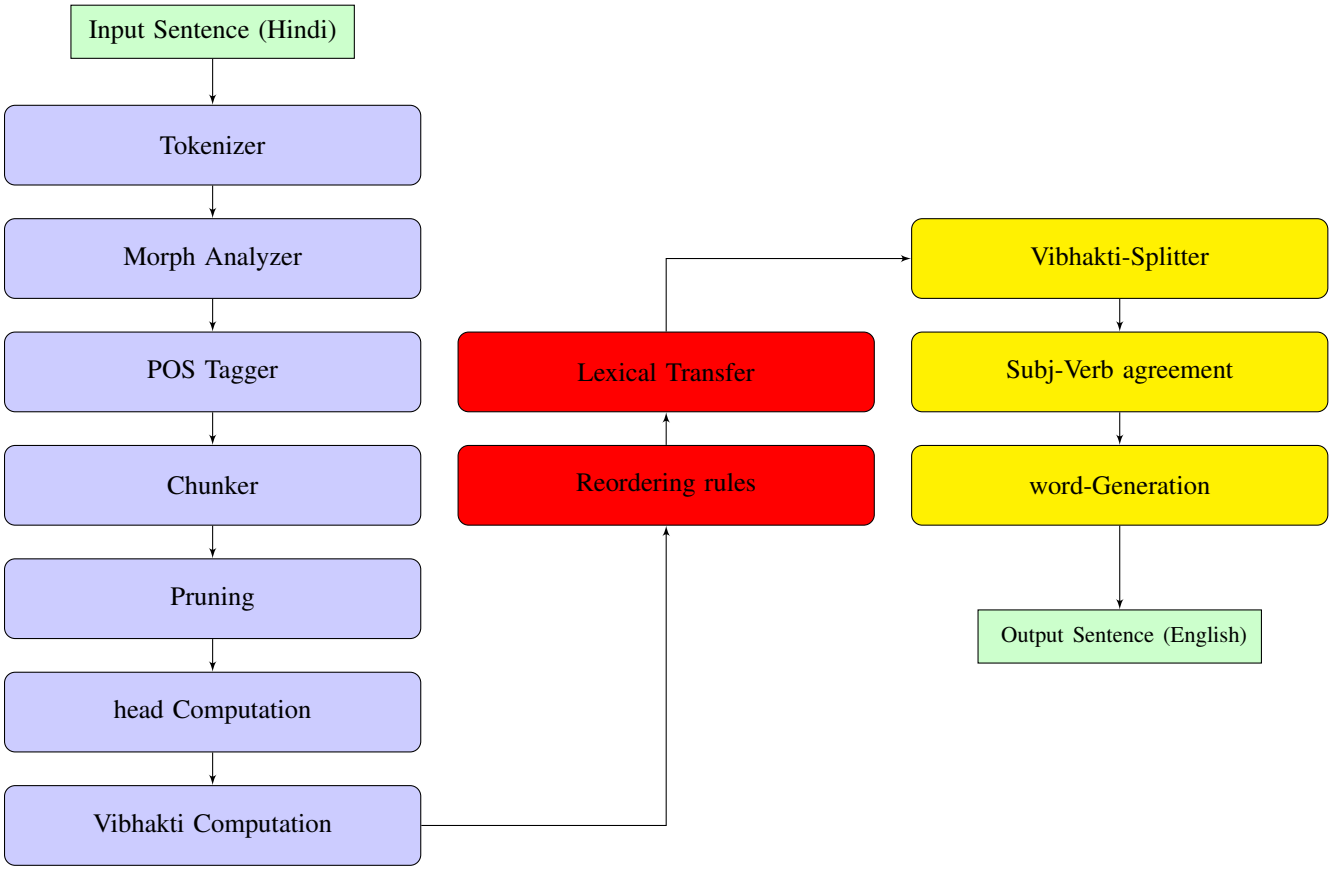

Figure 3.3: Architecture of Hindi-English RBMT

Morph Analysis

Morph Analyzer studies each token and extracts the morpheme (smallest meaningful unit) along with other structure information. For our system building an inflectional morphological Analyzer, based on paradigm model is applied. It uses the combination of paradigms and a root dictionary to provide root form and other grammatical features such as gender,number,person etc. This analyzer might give more than one morph information for any particular word. This is because a word can be used in multiple sense in Indian language context.

POS Tagger

POS tagger gives the syntactic category of every word in a sentence which helps in analyzing the role of each word. Conditional Random Fields (CRF) based POS tagger is developed using Hindi Treebank (Bhatt et al., 2009) containing 450,000 tokens out of which 70,15 and 15% of data was used for training, development and testing respectively. Tags for POS tagger has been taken from Anncorra (Bharati et al., 2006). POS tag corresponding to each token is added to column 3 in SSF and passed onto next module. The precision and recall for the POS tagger on a subset of Hindi treebank data are 0.95 and 0.96 respectively. In Figure 3.2 the third column indicates the part of speech corresponding to each identified token.

Chunker

Chunking divides a sentence on the basis of syntactically correlated words in a sentence. This eases the task of parsing a sentence by identifying the main informational structures in text. For Hindi the chunker divides the text into various phrases including noun, verb, adverbs etc. The main task is to identify the boundary and chunk tags for each chunk. Hindi chunker uses CRF for training with same proportion of data sets as used in POS tagging. The accuracy of the module is 93.4 % when automatic POS tags are provided and 95.1 % when gold POS tags are provided as features on the same test set used for testing POS tagger. In Figure 3.2 the start and end of chunk boundary are marked by round brackets along with chunk name in the third column along with starting chunk boundary.

Pruning

Morphological analyzer gives multiple morph information for a given word if necessary. But depending on the context of the word, only one morph is correct. This module finds the correct morph information and passes the information further down the pipeline. The first module in pruning is a POS tag based module. It checks for the compatibility of the POS tag and the lexical category of the various morphs given by morphological analyzer and rules out the incompatible. A predefined table has been created which provides set of compatible lexical categories for each POS tag. The output of this module is fed to a feature-disambiguator where rules are developed using context to prune remaining feature structures. If after the processing from above steps more than one morph remains, first morph is picked and passes on for further processing.

Head Computation

This module identifies the most important unit of a chunk. This marks the head of the chunk based on the constituents of the chunk and the phrase type given by the chunker. The head information is used in target generation where chunk agreement is studied with the help of head features.

Vibhakti Computation

It is also known as local word grouper and does the main task of vibhakti computation. Hindi being a free-word order language needs vibhakti markers to disambiguate various thematic roles associated with the verb of sentence. The post positions, also called noun vibhakti are also identified efficiently and root words are grouped with case/tam markers for each noun/verb chunk.

3.3.2 Transfer of Hindi to English

This subsection defines the steps followed to transform the analyzed Hindi input into an intermediate representation of English language.

Lexical and TAM (Tense, Aspect and Modality) Substitution

In this module, we substitute the Hindi root word identified by the Morphological Analyzer with the English root word using the bilingual dictionary [2]. A single word in Hindi can correspond to multiple meanings in English depending upon the context. E.g. The word fal in Hindi can correspond to *fruit (n)* and *to bear (v)* in English in different contexts. We use the lexical category to prune out multiple possibilities and select the first one of the remaining root words.

The module along with lexical substitution replaces the correct TAM in English corresponding to its TAM in Hindi. The TAM dictionary is a one-to-one mapping of all possible TAM markers in Hindi along with their target TAM markers. This dictionary was created as a part of *Shakti* MT system.

Reordering

Reordering is an intermediate stage where we order the source language according to the target language to generate fluent output. Hindi being a Subject-Object-Verb (SOV) word order language and English a Subject-Verb-Object (SVO) requires high level of ordering. We develop the reordering rules on the lines of *Shakti*, which is an English-Hindi MT system. The phrase structure rules are drafted manually, considering the linguistic irregularities between the language pair. The rule format is as follows

$$A\ B -> C\ D$$

which means that the source phrase components *A* and *B* correspond to target phrase components *C* and *D*. The short or the intra-chunk re-ordering is handled separately where we have defined rules on the post-position and the auxiliary verbs.

- NP [bacche (n.)] NP [bagiche (n.) mein (psp.)] VGF [khel (v.) rahe (vaux.) hain (vaux.)]
  children garden in play are
  children are playing in garden

[2] http://www.shabdkosh.com/

The above example illustrates the handling of three basic rules. To keep the system simple the short reordering (point 2 & 3) are being performed at later stage (explained below).

1. Long range re-ordering rule which changes the structure of the output sentence.

$$NP\ NP\ VGF \Rightarrow NP\ VGF\ NP.$$

2. Change of post-position to preposition.

3. The auxiliary verbs (vaux) which are appearing after the main verb in Hindi sentence appears just before the main verb in English.

The major issue with this module is that the rules need to be crafted manually and requires deep linguistic knowledge. Hindi being a free word order language needs large amount of rules to accommodate all possible structures. As the rule size keeps on increasing, multiple rules starts firing which increases the search space to achieve the optimal reordering.

3.3.3 Generation of English

The below defined modules takes the re-ordered output in its crude form and generates the translation which follows the correct English syntax.

Vibhakti Splitter

The vibhakti's (Auxiliary verbs, tense markers and the post-positions) computed by the vibhakti computation module needs to transformed into its correct form and position according to the rules of English language. The short re-orderings (discussed earlier) are being performed in this module, where we remove the TAM and the post-position from the feature structure and inserts them at the correct position.

Subject-Verb Agreement

The main verb identified by the chunker and the noun phrase[3] to which this main verb is attached needs to agree in terms of number. To make sure the agreement is their in the final transition, we replace the feature structure of verb phrase with feature structure of noun phrase. This module also takes care of some of the other agreement features[4] required to generate fluent output.

[3]For identifying the NP attached to main verb phrase we are using the syntactico-semantic parser developed in house using the Hindi treebank.

[4]http://grammar.yourdictionary.com/sentences/20-Rules-of-subject-verb-agreement.html

word-generation

The correct word needs to be generated from the root word along with the suffix information. The word generator module takes the root word along with the gender and number information. The module returns the correct word form using dictionaries in back-end which contains information about the gender, and number. This module also looks upon the previous words in the chunk to get he correct tense information added to the generated word.

3.4 Evaluation and Error Analysis

The evaluation of the system has been performed on a limited number of input sentences as the system supports limited reordering rules. The test set consists of 15 sentences with structures complying to those rules which we have included in our system. The overall evaluation for the test set resulted in 32.14 BLEU score.

We analysed the output translations for the 15 test sentences (will be referred to as test set1) along with output translations of some randomly picked input sentences from a news corpus (will be referred to as as test set2). The output of the test set2 resulted in unsatisfactory output as our system did not provide support for handling the input sentence structures. The major errors in test set1 were due to wrong word or no word being substituted from the bilingual dictionary. The system in some input sentences is not able to differentiate between the correct sense of target word for a given source word and lexical category combination due to multiple target words present in bilingual dictionary.

3.5 Challenges in developing a RBMT

We have presented a detailed explanation of developing a rule based Hindi-English MT system. The system relies on multiple dictionaries and countless user defined rules in order to achieve satisfactory translation across different sentence structures. The system works well for the rules developed, however its difficult to perform thorough evaluation of the system due to limited number of rules. The major advantage offered by the system is its domain independence viz. any input sentence can be easily translated as long as we have its domain words in our bilingual dictionary. The system provides more control to the users as errors in a translation can be easily traced back by looking output of any individual module and can be easily rectified.

The re-ordering module requires numerous manually crafted rules which requires deep linguistic knowledge. The language structure keeps on changing which means the system needs to be updated with rules. The amount of increasing rules add to further complexity as multiple rules start firing for a single sentence which requires additional probabilistic models to prune the less likely rules. The analysis and the generation modules can be plugged in for any other language pair (where Hindi is the source language or English as the target language), but the transfer modules need to be crafted afresh.

The SMT system requires large amount of parallel corpus for building a system. The system can be developed in less time as compared to the RBMT system as no linguistic rules are required and most of the information is extracted using statistical knowledge from the parallel corpus. Since it depends heavily on parallel corpus the notion of domain dependence comes into picture and the noise present in parallel corpus is passed on to the system as well. Past research have shown that the SMT systems perform better for languages which are close in word order.

Due to a trade-off between the SMT and and the RBMT, we decided to shift the focus of our research to statistical models due to the available resources and the knowledge available.

Chapter 4

Multi-Model SMT

Introduction

In the previous chapter we explored the rule based approach to develop a Hindi-English MT system and discussed some limitations of the system towards the end of the chapter. Due to the drawbacks of the rule based systems, we decided to explore statistical methods for our work. Due to availability of clean parallel corpus for Hindi-English (which is the most important resource for SMT) provided further motivation for switching on to statistical machine translation. In Statistical methods the information is extracted from the parallel corpus and can be easily extended to other language pairs. These methods also facilitate the use of linguistic information to further enhance the overall system.

In this chapter we present our efforts in exploring statistical methods for our Hindi - English MT system. Our approach uses statistical machine translation (SMT) where we use multiple strategies for translation and choose one translation as the final output. Since each technique has its own pros and cons, as certain sentence can get translated better using a technique as compared to other, we decided to go for ensemble MT. The selection is done using prediction strategies which analyze the candidate translations and predict the best suited translation.

We initially explored Phrase Based, Hierarchical and Factored models for SMT. For our experiments we considered only phrase based and hierarchical systems as factored model, taking POS as a factor, gave unsatisfactory (6.3 BLEU score)(Papineni et al., 2002) results compared to the former two systems ($>$21 BLEU score). We manually observed the translation of same 150 input sentences of different structures from each of the system and observed certain sentences get better translated from one system and some from the other. This led to the motivation of using ensemble learning in MT. The further insights were taken from the recent research focus on the quality estimation of machine translation (Bojar et al., 2013).

The aforementioned prediction methodology dynamically selects the best translation based on the presence or absence of certain features in the translations. We study and pick features that bear a high

correlation with a better translation. The features aim to effectively determine better translations from the candidates. We implemented the methodology by training a regression model on these features with the evaluation metric measure as the corresponding regression value. The regression model thus acquired is later utilized for guided selection of the translations from multiple models. The one having higher regression value i.e. correctness score, was taken to be the output of the system. The proposed method shows an increase of 0.64 BLEU score and high agreement with human evaluation.

4.1 Related Work

The only two reported Hindi-English MT systems are *Anubharati* (Sinha, 2004) and *Hinglish* (Sinha and Thakur, 2005). The former uses a combination of example-based, corpus based and elementary grammatical analysis while latter is an extension of the former. Other general translation systems like Google and Bing Translate have support for Hindi-English translation.

The SMT research community has shown interest towards the quality prediction of the translations, including two shared tasks organized in last few years in WMT[1]'12 (Callison-Burch et al., 2012), WMT'13 (Bojar et al., 2013) and WMT'14 (Bojar et al., 2014). The generic idea among the participants has been to model certain rich features to judge translation quality. Hardmeier et al.(2012) utilized nearly 99 features to train a regression model using 'tree kernels' for quality estimation. Avramidis (2012) also modeled the problem as regression, as well as classification problem and deduced that the latter doesn't perform well on unseen data. Though the chosen features truly governs the success of such a system, but previous studies have shown an inclination towards choosing regression modeling over classification.

In the work by (Hildebrand and Vogel, 2008), 23 features from N-best list of several MT systems was considered to improve the translation quality. They have shown a consistent increase in BLEU score by combining all systems progressively. Other significant works using multiple systems have been reported in system combination task of WMT'11 (Callison-Burch et al., 2011)

4.2 Translation Models

In this section we present the details of the corpus used for our experiments along with key steps adopted for training the baseline models.

[1]WMT: Workshop on Machine Translation

4.2.1 Corpus and data division

We have used the ILCI corpora (Jha, 2010), which contains parallel sentences for 11 languages (including Hindi and English) from the health domain with Hindi as their source language. We have used 25000 parallel sentences (23951 after cleaning) for our experiments. The sentences are encoded in utf-8 format. We converted the Hindi sentences to wx[2] notation for easy tokenization. The corpus is split into training (75%, 18319 sentences), development (15%, 3235 sentences) and testing sets (10%, 2397 sentences). From the test set we randomly select 100 sentences for a separate human evaluation.

Division	**No.of Sentences**
Training	18319
Development	3235
Testing	2397
Test-Human Evaluation	100

Table 4.1: Data Division and Corpus Statistics

4.2.2 Training Translation Models

We trained two Hindi to English translation models, phrase-based (Koehn et al., 2003a) and hierarchical (Chiang, 2005), using Moses (Koehn et al., 2007a). In phrase based modeling, both source and target sentences are divided into separate phrases while in hierarchical modeling the phrases can be recursive as well. Giza++ (Och and Ney, 2000a) is used for phrase alignments and SRILM (Stolcke and others, 2002b) with Kneser-ney smoothing (Kneser and Ney, 1995a) is used for training the language model (LM) of order 5. Mert (Och, 2003a; Bertoldi et al., 2009) is used for minimum error-rate training i.e. to tune the model using the development data-set. Top 1 result is considered as the default output. We obtained a BLEU score of **21.18** and **21.10** for phrase-based and hierarchical models respectively over the test set. For further experiments we have used these models as our baseline system.

4.3 Translation Selection

The two translations obtained from the phrase based and hierarchical systems are then analyzed for their translation quality. The procedure involves calculating the feature vectors from the obtained translations and feeding them to a pre-trained regression model. The candidate translation giving higher regression value is selected as the final output.

[2]http://sanskrit.inria.fr/DATA/wx.html

Next we present the methodology for training the regression model, mentioned earlier. The training data for this task is same as the development set used for tuning the translation models. The target value, corresponding to a feature vector, is calculated using MT evaluation metric scores (BLEU, METEOR or NIST) over the system output with reference data. Each data entry corresponds to a sentence translation and the value is the estimation of its quality.

4.3.1 Features

Among studied features, the following are employed for regression modeling.

- **Token count:** Number of tokens in source and target sentence and their ratio.
- **Language Modeling (LM):** From the LM of source and target language, log-LM score and perplexity value (computed using SRILM).
- **Part of speech (POS) language modeling:** From Source and target POS LM, log-LM score and perplexity value (computed using SRILM).
- **Out of vocabulary (OOV) words:** Number of OOV words in translated output.

Apart from these syntactic and textual features, a linguistically motivated feature has also been included:

- **Entropy of Parse tree:** We have considered entropy of label, attachment and joint entropy of label+attachment. This score corresponds to the correctness of the parse tree, detailed down to each edge of the parse.

  Parse tree confusion score

  For calculating entropy of parse tree, we use an augmented version of MaltParser (Nivre et al., 2007), built in-house, to dynamically compute a confusion score for dependency arcs, in typed dependency parsing framework. This is based on the methods proposed by (Jain and Agrawal, 2013), where quantification of confusion is done by calculating entropy with class membership probabilities of the parser actions. The augmented version predicts confusion score according to different types of training behaviors. Maltparser provides three different ways of predicting the output and thus accordingly augmented version predicts confusion score namely, separately for arc-labels and arc formations and a joint model for predicting arc-labels and formations simultaneously.

4.3.2 Estimation Using Regression

4.3.2.0.1 Preprocessing The data is normalized by scaling the values between $[0, 1]$. However, with simple *min-max* scaling, the system was observed to perform clumsily due to presence of outlier values. To overcome this, we utilized interquartile-range[3] to first map the outliers to the min-max bounds.

The aforementioned regression model is built with support vector regression (SVR) using LIBSVM toolkit (Chang and Lin, 2011). Tuned parameters are attained with *gridregression*[4] script. The cost/margin trade-off, the epsilon in loss function and the kernel type are set to optimized values and all other parameters are left at default.

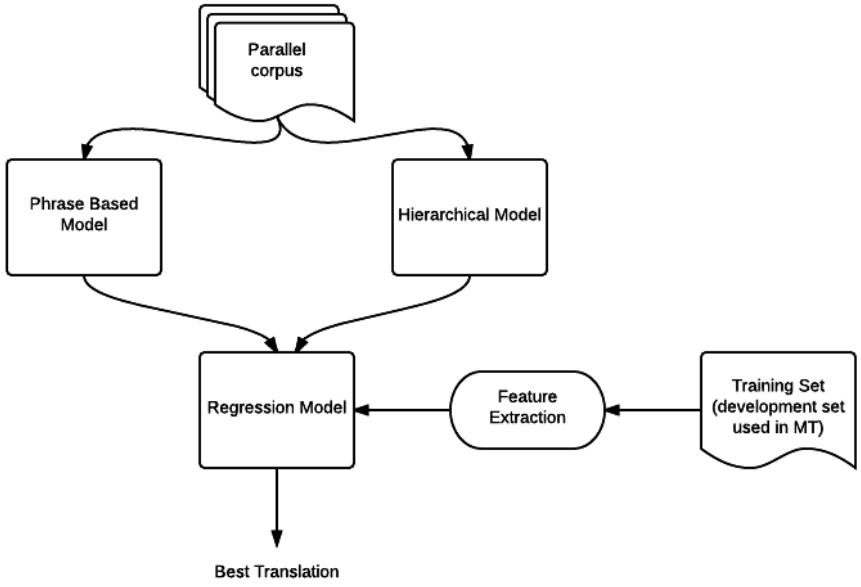

Figure 4.1: Architecture of multi-model SMT.

4.4 Experiments and Results

The Figure 4.1 shows the basic architecture of multi-model SMT. The results of some of our experiments using BLEU as target regression value are reported in Table 4.2. We experimented with radial basis function (RBF), polynomial and linear kernels using different feature sets (Table 4.4). Using RBF kernel with feature set #6 give better results than the baseline phrase and hierarchical models. After

[3] www.en.wikipedia.org/wiki/Interquartile_range

[4] www.csie.ntu.edu.tw/ cjlin/ libsvmtools/gridsvr /gridregression.py

adding LM feature i.e. feature set #8, a slight improvement in the BLEU scores is observed, however, contradictory results are found for the RBF kernel. Combining the complete feature set with linear kernel yield best results in terms of BLEU score of **21.82** indicating an increase of **0.64** from the baseline systems.

We also study the effect of each feature by creating an independent regression model for it. The LM feature gives the best BLEU score of **22.21**, indicating an increase in BLEU score upon the baseline systems. The reason for improvement is the high correlation between the observed (LM score) and predicted value (BLEU score). But this system does not go along well with human evaluation as discussed later.

			Hierarchical		Phrase		Evaluation		
System	Algorithm	Feature Set	MSE	SCC	MSE	SCC	BLEU	NIST	Meteor
-	RBF	#6	0.0679	0.0144	0.0637	0.0143	21.44	6.68	56.39
$sys3$	Linear	#6	0.0664	0.0231	0.0628	0.0128	20.66	6.43	56.20
-	Polynomial	#6	0.0672	0.0142	0.0642	0.0059	20.90	6.47	56.08
-	RBF	#8	0.0620	0.1088	0.0585	0.1131	21.14	6.47	56.24
-	Linear	#8	0.0652	0.0698	0.0605	0.0760	20.97	6.45	56.29
-	Polynomial	#8	0.0687	0.0719	0.0651	0.0619	21.14	6.46	56.14
-	RBF	#11	0.0557	0.1814	0.0527	0.1771	21.67	6.55	56.54
$sys2$	Linear	#11	0.0603	0.1362	0.0558	0.1273	**21.82**	6.57	**56.73**
-	Polynomial	#11	0.0643	0.1004	0.0587	0.0959	21.37	6.52	56.39
$sys1$	RBF	#3	0.0649	0.0622	0.0608	0.0599	*22.21*	*6.71*	56.54

Table 4.2: Regression results. Mean Squared Error (MSE), Squared correlation coefficient (SCC)

4.5 Evaluation

4.5.1 Human Evaluation

Improvements in BLEU score does not ensure a better MT system (Zhang et al., 2004). To ascertain that, this multi-model system actually gives better translations than the baseline systems, we conducted a separate manual evaluation over 100 sentences selected randomly from the test set. Five human evaluators[5] are provided with source Hindi sentence and output of phrase and hierarchical systems. They are asked to select the better translation among the two candidate translations from our phrase based and hierarchical models. The better translation output of the two is decided by "Max-Wins" voting strategy.

[5]Hindi as their mother tongue and proficient in English

Out of those 100 translations, 63 are marked better for the phrase based and rest for the hierarchical system. Selecting only the translations of phrase based system maximizes the agreement with human evaluators, however overall quality of translation of document is reduced as 37 sentences are selected from the hierarchical system according to human judgment.

Table 4.3 presents the automatic evaluation scores (BLEU, METEOR and NIST) and percentage agreement with human judgment, for three best performing systems. Here $sys1$ is the model generated by considering only LM as the feature using 'RBF kernel', $sys2$ considers all the aforementioned features using 'linear kernel' and $sys3$ is created using feature set #6 (refer Table 4.4) using 'linear kernel'. $sys2$ gives higher BLEU score than baselines and the highest agreement with the human evaluation (61 out of 100 sentences).

System	BLEU	METEOR	NIST	Agreement(%)
Phrase	21.18	56.53	6.61	-
Hierarchical	21.10	56.00	6.52	-
$sys1$	22.21	56.54	6.71	47
$sys2$	21.82	56.73	6.57	**61**
$sys3$	20.66	56.20	6.43	59

Table 4.3: Evaluation scores and agreement with human evaluation of various translation systems.

Although using only LM feature ($sys1$) shows a slight improvement in automatic evaluation due to high correlation between LM score and BLEU score, this system does not show an accordance with human evaluation (47 out of 100 sentences). This correlation coefficient is high as BLEU score computes n-grams to evaluate a translation. The evaluation scores for $sys2$ are high and show high agreement with human judgment. Though the automatic evaluation scores obtained for $sys3$ are low, yet this system also shows a high agreement with human judgment.

Feature Set	Features
#3	log-lm score
#6	Entropy of label, attachment and joint entropy of label+attachment, token count of source and target language, ratio of token count of target language to source
#8	*feature set #6* + log-lm score and perplexity value
#11	*feature set #8* + POS log-lm score and perplexity and count of OOV words

Table 4.4: Description of Feature Sets.

4.5.2 Comparison with Google and Bing Translate

We also compared our system with the Google and Bing translate. We tested the output of these two systems on our test sentences. The BLEU score of Google and Bing translations turned out to be 14.75 and 15.10 respectively. Translations from our system ($sys2$) are observed to be way better than these systems for the test set. However this could be due to the difference in domain of training corpus.

4.6 Conclusion

In this chapter we introduced an approach to estimate the quality of machine translation and dynamically select the better translation at run-time. Combining the text analysis and linguistic features, results in a system which shows improvement over the baseline system and shows high agreement with human judgment. Since these models are computationally expensive we plan to execute these models in a distributed manner which will then combine the outputs from each system. The approach is fairly general and is dependent on resources of individual language.

Chapter 5

Effect of semantic similarity in Phrase-based Machine Translation

Introduction

The current state of the art Statistical Machine Translation (SMT) systems (Koehn et al., 2003b) do not account for semantic information or semantic relatedness between the corresponding phrases while decoding the n-best list. The phrase pair alignments extracted from the parallel corpora offers further limitation of capturing contextual and linguistic information. Since the efficiency of a statistical system depends on the quality of parallel corpora, low resourced language pair fails to meet the desired standards of translation.

Word representation is being widely used in many Natural Language Processing (NLP) applications like information retrieval, machine translation and paraphrasing. The word representation computed from continuous monolingual text provide useful information about the relationship between different words. Distributional semantics offers a notion of capturing semantic similarity between words occurring in similar context, where similar meaning words are grouped closely in a high dimension word space model. Each word is associated with an n-dimensional vector which represents its position in a vector space model and similar words are at small distance in comparison to relatively opposite meaning words.

The recent work in word vectors have shown to capture the linguistic relations and regularities. The relation between words can be expressed as a simple mathematical relation between their corresponding word vectors. The recent paper by Mikolov (Mikolov et al., 2013c) have shown through a word analogy task that the vec (”man”) - vec (”woman”) + vec (”king”) should be close to vec (”queen”). Capturing of these relations along with word composition have shown significant improvements in various NLP and information retrieval tasks.

In this chapter, we present our ideas of capturing the semantic similarity between phrase pairs in context of SMT and use the scores as features while decoding the n-best list. We make use of word representation computed from two different methods: word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) and show the effect of varying the context window and vector dimension for Hindi-English language pair. We use partial least square (PLS) regression to learn the bilingual

embeddings using the easily available resources for any language pair. Figure 5.1 shows the information flow of data when computing semantic similarity between phrases. The proposed approach is fairly general and can be extended for other language pairs as well.

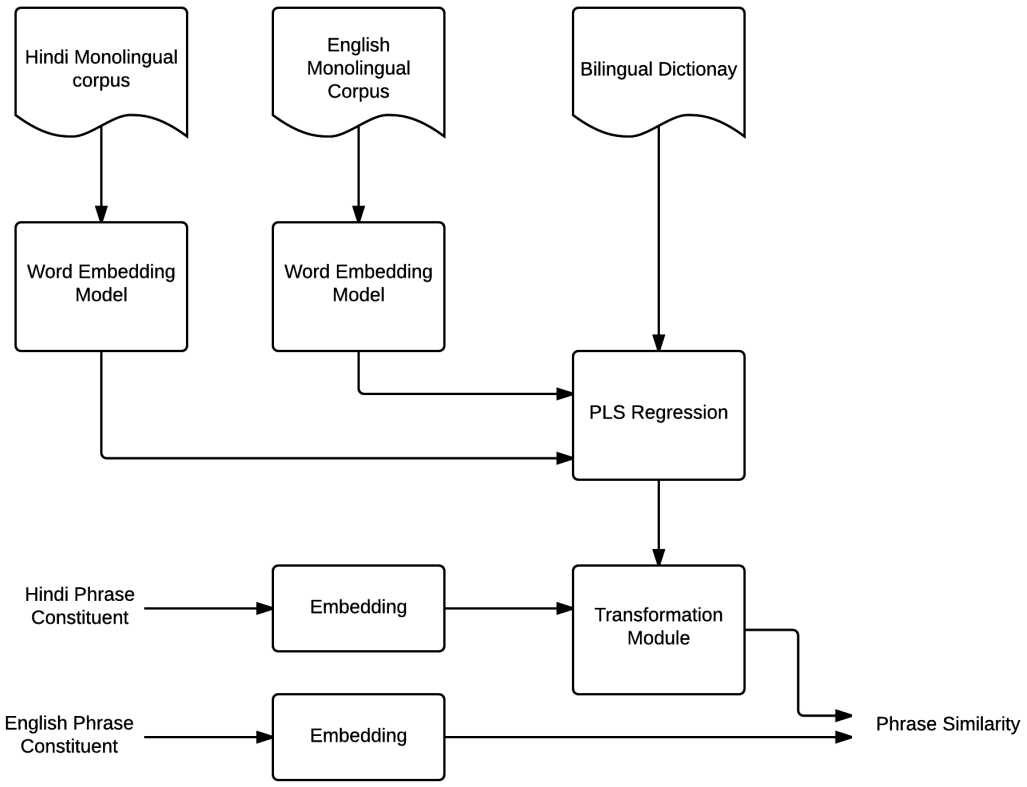

Figure 5.1: The figure represents the flow of information while computing semantic features for phrase table.

5.1 Related Work

The current research community has shown special interest towards vector space models by organizing various dedicated workshops in top rated conferences. Word representations have been used in many NLP applications like information extraction (Paşca et al., 2006; Manning et al., 2008), sentiment prediction (Socher et al., 2011) and paraphrase detection (Huang, 2011).

In the past various methodologies have been suggested to learn bilingual word embeddings for various natural language related tasks. (Mikolov et al., 2013b) and (Zou et al., 2013) have shown significant improvements by using bilingual word embeddings in context of machine translation experiments. The former applies linear transformation to bilingual dictionary while the latter uses word alignments

knowledge. Zhang (2014) proposed an auto-encoder based approach to learn phrase embeddings from the word vectors and showed improvements by using semantic similarity score in MT experiments. The phrase vector is generated by recursively combining the two children vector into a same dimensional parent vector using the method suggested by (Socher et al., 2011).

The work of (Gao et al., 2013) proposes a method for learning the semantic representation of phrase using features (multi-layer neural network) which is then used to compute the distance between them in a low dimensional space. The learning of weights in the neural network is guided by the BLEU score (ultimate goal to improve the quality of translation through increase in BLEU score) which makes it sensitive towards the score. Wu (2014) proposed an approach of using supervised model of learning context-sensitive bilingual embedding where the aligned phrase pairs are marked as true labels.

Since these defined methods depends heavily on the quality of word vectors, a number of approaches have been suggested in past to learn word representations from monolingual corpus: word2Vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014) and (Huang et al., 2012).

In this work, we extend the phrase similarity work by using the regression approach to learn the bilingual word embeddings. We employ vector composition approach to compute the phrase vector, where we add vectors of each constituent word to achieve the phrase vector. We also present the comparison of using different word embedding models along with varying context window and vector dimension which has not been shown (in detail) in any of the previous works. As pointed by (Mikolov et al., 2013b) linear transformation works well for language pairs which are closely related, however in this work we experiment with PLS regression which also establishes a linear relationship between words but is much more efficient than the simple least squares regression (explained in 5.3.2).

5.2 Learning word representation

We have used a part of WMT'14[1] monolingual data and news crawled monolingual data to learn word representations for English and Hindi respectively. We added the ILCI bilingual corpus (Jha, 2010) of English and Hindi to the monolingual data. The corpus statistics (after cleaning) are provided in table 5.1. The vocabulary refers to the words in embeddings with a minimum frequency of five within the corpus.

Language	**# of Words**	**Vocabulary**
English	250M	274K
Hindi	80M	184K

Table 5.1: Monolingual corpus statistics

[1] http://www.statmt.org/wmt14/translation-task.html

5.2.1 word2Vec

The word2Vec model proposed by (Mikolov et al., 2013a) computes vectors by skip-gram and continuous bag of words (CBOW) model. These models use a single layer neural networks and are computationally much more efficient than any previously proposed model. The CBOW architecture of model predicts the current word based on the context whereas the skip-gram model predicts the neighboring words depending on the current word. Experiments have shown CBOW architecture to perform better on the syntactic task and skip-gram based architecture on the semantic tasks. We have used the skip-gram architecture of word2Vec in our experiments as it has been shown to perform better for semantic related tasks.

5.2.2 GloVe

The Global Vector model of learning word representation was proposed by (Pennington et al., 2014) which computes the word vectors from a global word-word co-occurrence matrix. The relationship between words is extracted by using the ratio of co-occurrence probability with various probe words, which distinguishes between the relevant and irrelevant words. The co-occurrence probability of word *'i'* to that of word *'j'* is studied on the basis of a probe word *'k'* which is computed on the basis of a ratio P_{ik}/P_{jk}. The ratio is expected to be higher if word *'k'* is more related to word *'i* and low if it is related to word *'j'*. Stating an example from the original authors, for i = *ice* and j = *steam* the ratio would be higher if k = *solid* as ice is solid. However for k = *gas* the ratio would be low as steam is more related to gas. The author shows significant improvement over the word2Vec model on various NLP tasks (word similarity, word analogy and named entities recognition).

For training both the models we have altered the vector size and the context window, while all other parameters are set to default.

5.3 Experiments

5.3.1 Baseline MT System

We have used the ILCI corpora (Jha, 2010) which contains 50000 Hindi-English parallel sentences (49300 after cleaning) from health and tourism domains. The corpus is randomly split (equal variation of sentence length) into training (48300 sentences), development (500 sentences) and testing (500 sentences). The corpus division is different from our previous work on multi-model SMT as more parallel corpus was prepared over a period of time and hence results are not comparable to the previous one.

Division	# of sentences
Training	48300
Development	500
Testing	500

Table 5.2: MT system corpus statistics

We trained two Phrase based (Koehn et al., 2003b) MT systems (Hindi - English and English - Hindi) using the Moses toolkit (Koehn et al., 2007b) with phrase-alignments (maximum phrase length restricted to 4) extracted from GIZA++ (Och and Ney, 2000b). We have used the SRILM (Stolcke and others, 2002a) with Kneser-Ney smoothing (Kneser and Ney, 1995b) for training a language model of order five and MERT (Och, 2003b) for tuning the model with development data. We achieve a BLEU (Papineni et al., 2002) score of 19.89 and 22.82 on English-Hindi and Hindi-English translation systems respectively. These translation scores serves as our baseline for further experiments.

5.3.2 Partial Least Square (PLS) Regression

We generate the word embeddings of both Hindi and English using monolingual corpus using two previously mentioned methods (section 5.2). Since both the word embeddings are in different space (computed independently), there is a need to map the source vector space to target vector space or vice versa.

We employ the PLS (Abdi, 2003) regression to learn the transformation matrices. The observable variables (X) are the word embeddings of one language, while the predictable variables (Y) are the word embeddings of the other language. The observable and the predictable are $n \times d$ matrices, where 'n' is the number of words used (explained in subsection 5.3.3) and 'd' is the word embedding dimension. Our task is to compute a transformation matrix of $d \times d$ dimension which will be used to transform any given language word vector to its corresponding other language vector.

The PLS[2] regression algorithm works by projecting both X and Y matrices to a new space, and decomposes them into a set of orthogonal factors. The observables are first decomposed as $T = XW$ where 'T' and 'W' are the factor score matrix and weight matrix respectively. The predictable 'Y' is then estimated as $Y = TQ+E$ where 'Q' and 'E' are regression coefficient matrix and error term. We have the final regression model as $Y = XB+E$ where $B = WQ$ acts as our transformation matrix.

5.3.3 Learning Transformation matrix

We employ PLS regression to learn bilingual word embeddings using a English-Hindi bilingual dictionary [3]. We have used 15000 words for training the regression model and another set of 1500 words for testing purpose. The bilingual pair of training words are selected based on the frequency of

[2]http://www.statsoft.com/Textbook/Partial-Least-Squares

[3]http://www.shabdkosh.com/

those words occurring in a large plain text which consist of 10000 words from high frequency and 2500 words each of low and medium frequencies.

The observable variable and the predictable variables in the PLS regression are the word vectors of each word pair from their respective language word embedding models. We finally achieve two transformation models which transform source to target vector space and target to source vector space. We have presented average similarity score on the test set in table 5.3 after transforming English words to Hindi word space.

Dimension	**word2Vec**		**GloVe**	
	CW 5	CW 7	CW 5	CW 7
50	0.53	0.51	0.48	0.49
100	0.47	0.49	0.43	0.44
150	0.44	0.47	0.41	0.42
200	0.42	0.45	0.38	0.41
250	0.41	0.43	0.38	0.39
300	0.40	0.41	0.37	0.39
400	0.40	0.38	0.35	0.36
500	0.38	0.37	0.34	0.36

Table 5.3: Average word cosine similarity scores on test set. Context Window (CW)

5.3.4 Decoding with semantic similarity score

In the phrase based MT system we add two features (semantic similarity scores) to the bilingual phrase pairs. Since we need the vector representation of a phrase, we employ the works of (Mitchell and Lapata, 2008) on compositional semantics (adding the vectors) to compute the phrase representation. For a give phrase pair (*s,t*), we transform each constituent word of the source phrase *'s'* to the target word space and add the the transformed word embedding to the resultant source vector. We ignore the word if it does not occur in the word embeddings vocabulary. Similarly, we compute the phrase representation of the target phrase *'t'* by simply adding the word vectors to the resultant target vector. We then compute the cosine similarity between the two vectors which acts as a feature for the MT decoder. We also include the similarity score of transforming the target word phrase to source phrase as another feature. The phrase table is tuned with the previously used development data (development set used for tuning baseline MT system) using the MERT algorithm to compute the weight parameters for the baselines features and semantic similarity features.

5.4 Results and Discussion

The results of word similarity scores on the test set (bilingual dictionary words section 5.3.3) are presented in table 5.3 using the computed transformation matrix for English to Hindi. The similarity scores are continuously decreasing with increase in dimension, which shows that the proposed approach works better at lower dimensions for word similarity task. The word2Vec model is performing better than the GloVe model on word-similarity task. Within the same model the word2vec model with context window of five performs better than the model with context window of seven, while it is opposite for the GloVe model.

The results of our experiments (on the same test data used for evaluating the baseline MT systems) with varying dimensionality and context window are presented in table 5.4, 5.5, 5.6 and 5.7. Each of the bold marked values in the tables indicate an increase in BLEU score over the baseline. The figure 5.2, 5.3, 5.4 and 5.5 presents the comparison of BLEU score for each of the model. The highest BLEU score achieved for English-Hindi translation system is **20.53** (increase of **0.64** BLEU score over the baseline) using GloVe model with a 500 dimension vector and a context window of 5, whereas the highest score for Hindi-English system is **23.56** (increase of **0.74** BLEU score over the baseline) using word2Vec model and context window of 7.

We also evaluate our system on a separate set of 462 sentences by finding the best set of hyper-parameters from the above mentioned test set. We achieve a BLEU score of 22.85 for Hin-Eng and and 19.91 for Eng-Hin system on the set of 462 sentences. Incorporating the semantic feature trained from the best performing system in Hin-Eng (Dim:500, CW:7 and Model:word2Vec) we achieve a BLEU score of 23.43 and for Eng-Hin (Dim:500, CW:5 and Model:GloVe) a score of 20.45. The proposed approach shows an increase of 0.58 for Hin-Eng and 0.54 for Eng-Hin systems.

It is quite interesting to note that the increasing dimensionality and context window does not ensure increasing BLEU scores. It is evident that at a certain dimensionality the decoder algorithm (combining feature scores using log-linear model) can start distinguishing between the good and bad translations. The Hindi-English system shows improvements for almost all the cases, whereas English-Hindi system does not show similar behavior. Though the word similarity scores indicates better performance at lower dimensions, the MT experiments BLEU scores does follow the same trend. Since this language pair has not been widely explored, the results on word similarity and MT scores are not directly comparable to the earlier proposed methods.

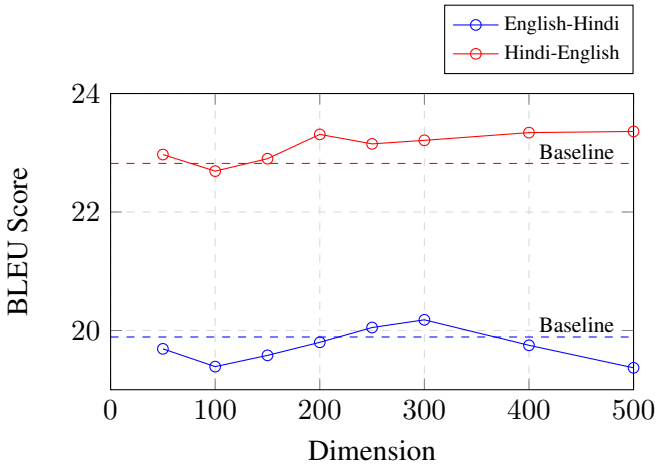

Figure 5.2: Plot of BLEU score variation using Word2Vec with a context window of 5

Dimension	Eng-Hin	Hin-Eng
50	19.69	**22.97**
100	19.39	22.69
150	19.58	**22.90**
200	19.80	**23.31**
250	**20.05**	**23.15**
300	**20.18**	**23.21**
400	19.75	**23.34**
500	19.37	**23.36**

Table 5.4: BLEU score of system using Word2Vec model with a context window of 5.

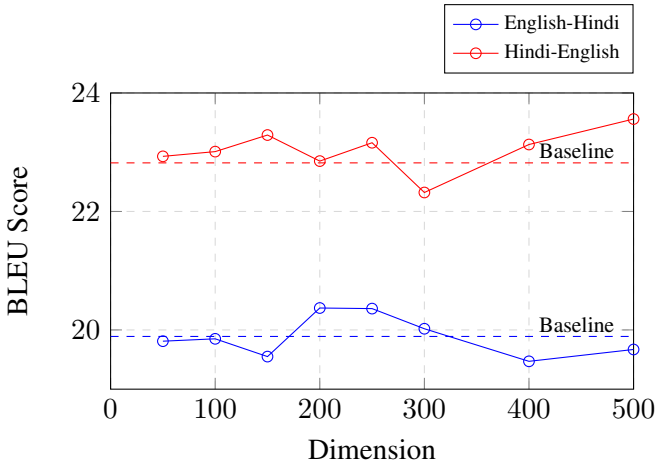

Figure 5.3: Plot of BLEU score variation using Word2Vec with a context window of 7

Dimension	Eng-Hin	Hin-Eng
50	19.81	**22.93**
100	19.85	**23.01**
150	19.55	**23.29**
200	**20.37**	**22.85**
250	**20.36**	**23.16**
300	**20.02**	22.32
400	19.47	**23.13**
500	19.67	**23.56**

Table 5.5: BLEU score of system using Word2Vec model with a context window of 7.

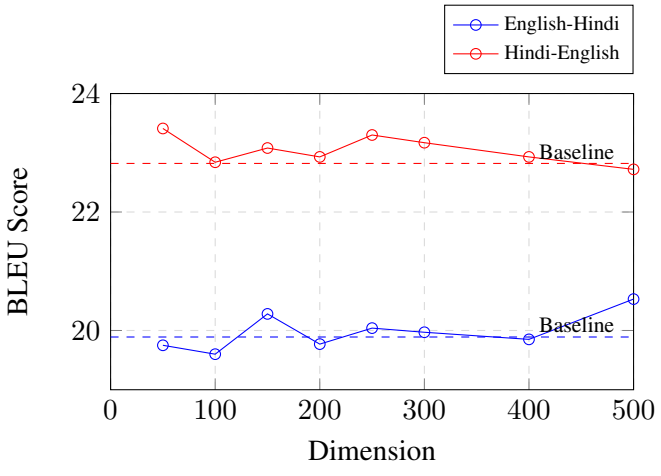

Figure 5.4: Plot of BLEU score variation using GloVe with a context window of 5

Dimension	Eng-Hin	Hin-Eng
50	19.75	**23.41**
100	19.60	22.84
150	**20.28**	**23.08**
200	19.77	**22.93**
250	**20.04**	**23.30**
300	**19.97**	**23.17**
400	19.85	**22.93**
500	**20.53**	22.72

Table 5.6: BLEU score of system using GloVe model with a context window of 5.

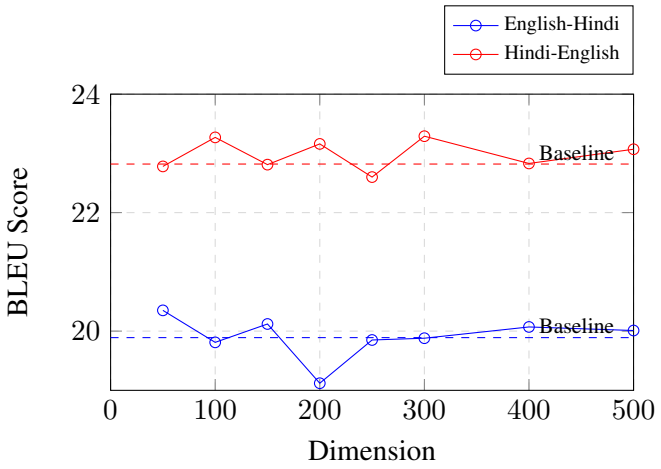

Figure 5.5: Plot of BLEU score variation using GloVe with a context window of 7

Dimension	Eng-Hin	Hin-Eng
50	**20.35**	22.78
100	19.81	**23.27**
150	**20.12**	22.81
200	19.12	**23.16**
250	19.85	22.60
300	**19.88**	**23.29**
400	**20.07**	**22.83**
500	**20.01**	**23.07**

Table 5.7: BLEU score of system using GloVe model with a context window of 7.

5.5 Conclusion

In this chapter we explored the use of semantic similarity between phrase pairs as features while decoding the n-best list. The bilingual word embeddings are computed through PLS regression using a bilingual dictionary (which is an easily available resource considering low resourced language pairs as well) with limited vocabulary size. This method shows an increase in BLEU score for both English-Hindi and Hindi-English MT systems. This approach is quite effective in terms of overall complexity as the models developed by Zou (2013) and Zhang (2014) require much larger time for training.

Chapter 6

Conclusions and Future Work

In this thesis we have presented the key approaches of developing a machine translation system. We have presented in detail our approach of developing a rule based system for Hindi-English and then discuss in depth our major motivation for shifting towards statistical methods.

Multiple statistical algorithms for machine translation have been discussed in the past. Since each algorithm has its pros and cons, we proposed a multi-model approach where we dynamically selected the best sentence from multiple translations. This is guided by a regression model which consumes the features from the source sentence and its corresponding translation from multiple models and selects the best sentence from the predicted score. The experimental results show an increase in performance over the baseline system. In future, we plan to integrate a few more linguistic and other statistical features, extracted at the decoding stage, which can be considered to improve the selection criteria. Prediction of the quality score using active learning is an interesting area to be looked into. Sequentially running both the phrase and hierarchical system may result in increase in time of computation as parse tree and other feature computation add to decoding time. To overcome this issue we have employed distributed computing so as to compute all features in parallel for phrase and hierarchical systems.

In our next work, we used semantic information as a feature in machine translation while decoding. The phrases which are extracted by the automatic alignment tools do contain some noise and as a result degrades the performance of the system. The semantic information filters the phrases while decoding which is show by an increase in the performance of the system. As a part of future work, we propose the use of auto-encoders (Socher et al., 2011) to learn phrase representations as currently we are treating 'black'+'forest' and 'forest'+'black' to be having the same vector representation while in reality they are different. Since the words in one language can not be just linearly transformable to another language we will try to explore the use of feed-forward neural networks to learn non-linear transformations while minimizing the euclidean distance between the word embedding pairs. We also plan to extend the work by including the linguistic information in the word embeddings and taking the advantage of Hindi being a morphologically rich language.

Related Publications

[1] **Kunal Sachdeva** and Dipti Misra Sharma. 2015. Exploring the effect of semantic simlarity for Phrase based Machine Translation In *3rd Workshop on Continuous Vector Space Models and their Compositionality organized in conjunction with ACL* , (CVSC 2015), Beijing, China.

[2] **Kunal Sachdeva**, Rishabh Srivastava, Sambhav Jain, Dipti Misra Sharma. 2014. Hindi to English Machine Translation: Using Effective Selection in Multi-Model SMT In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference*, (LREC 2014), Reykjavik, Iceland.

[3] Karan Singla, **Kunal Sachdeva**, Diksha Yadav, Srinivas Bangalore and Dipti Misra Sharma 2014. Reducing the impact of data sparsity in statistical Machine Translation In *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation organized in conjunction with EMNLP*, (SSST-8 2014), Doha, Qatar.

[4] Sambhav Jain, **Kunal Sachdeva**, and Ankush Soni. 2013. Effect of Transliteration on Readability. In *Proceedings of the Proceedings of the 15th International Conference on Human Computer Interaction*, (HCI International 2013), Las Vegas, USA.

References

[Abdi2003] Abdi, H. (2003). Partial least squares regression (pls-regression).

[Avramidis2012] Avramidis, E. (2012). Quality estimation for machine translation output using linguistic analysis and decoding features. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 84–90. Association for Computational Linguistics.

[Banerjee and Lavie2005] Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

[Bertoldi et al.2009] Bertoldi, N., Haddow, B., and Fouet, J.-B. (2009). Improved minimum error rate training in moses. *The Prague Bulletin of Mathematical Linguistics*, 91(1):7–16.

[Bharati et al.2006] Bharati, A., Sangal, R., Sharma, D. M., and Bai, L. (2006). Anncorra annotating corpora guidelines for pos and chunk annotation for indian languages. *LTRC-TR31*.

[Bhatt et al.2009] Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D. M., and Xia, F. (2009). A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189. Association for Computational Linguistics.

[Bojar et al.2013] Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 workshop on statistical machine translation. In *8th Workshop on Statistical Machine Translation*.

[Bojar et al.2014] Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics Baltimore, MD, USA.

[Callison-Burch et al.2011] Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

[Callison-Burch et al.2012] Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

[Chang and Lin2011] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

[Chiang2005] Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

[Chiang2007] Chiang, D. (2007). Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.

[Doddington2002] Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

[Gao et al.2013] Gao, J., He, X., Yih, W.-t., and Deng, L. (2013). Learning semantic representations for the phrase translation model. *arXiv preprint arXiv:1312.0482*.

[Hardmeier et al.2012] Hardmeier, C., Nivre, J., and Tiedemann, J. (2012). Tree kernels for machine translation quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 109–113. Association for Computational Linguistics.

[Hildebrand and Vogel2008] Hildebrand, A. S. and Vogel, S. (2008). Combination of machine translation systems via hypothesis selection from combined n-best lists. In *MT at work: Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 254–261. Citeseer.

[Huang et al.2012] Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

[Huang2011] Huang, E. (2011). Paraphrase detection using recursive autoencoder.

[Jain and Agrawal2013] Jain, S. and Agrawal, B. (2013). A dynamic confusion score for dependency arc labels. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1237–1242.

[Jha2010] Jha, G. N. (2010). The tdil program and the indian language corpora initiative (ilci). In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010). European Language Resources Association (ELRA)*.

[Kneser and Ney1995a] Kneser, R. and Ney, H. (1995a). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

[Kneser and Ney1995b] Kneser, R. and Ney, H. (1995b). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

[Koehn and Hoang2007] Koehn, P. and Hoang, H. (2007). Factored translation models. In *EMNLP-CoNLL*, pages 868–876. Citeseer.

[Koehn et al.2003a] Koehn, P., Och, F. J., and Marcu, D. (2003a). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

[Koehn et al.2003b] Koehn, P., Och, F. J., and Marcu, D. (2003b). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

[Koehn et al.2007a] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007a). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

[Koehn et al.2007b] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007b). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

[Manning et al.2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

[Mikolov et al.2010] Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

[Mikolov et al.2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al.2013b] Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

[Mikolov et al.2013c] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Mitchell and Lapata2008] Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *ACL*, pages 236–244. Citeseer.

[Naskar and Bandyopadhyay2005] Naskar, S. and Bandyopadhyay, S. (2005). Use of machine translation in india: Current status. *Proceedings of MT SUMMIT-X, Phuket, Thailand*, pages 465–470.

[Nivre et al.2007] Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

[Och and Ney2000a] Och, F. J. and Ney, H. (2000a). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447. Association for Computational Linguistics.

[Och and Ney2000b] Och, F. J. and Ney, H. (2000b). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447. Association for Computational Linguistics.

[Och2003a] Och, F. J. (2003a). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

[Och2003b] Och, F. J. (2003b). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

[Papineni et al.2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[Paşca et al.2006] Paşca, M., Lin, D., Bigham, J., Lifchits, A., and Jain, A. (2006). Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 809–816. Association for Computational Linguistics.

[Pennington et al.2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*.

[Singla et al.2014] Singla, K., Sachdeva, K., Yadav, D., Bangalore, S., and Sharma, D. M. (2014). Reducing the impact of data sparsity in statistical machine translation. *Syntax, Semantics and Structure in Statistical Translation*, page 51.

[Sinha and Thakur2005] Sinha, R. M. K. and Thakur, A. (2005). Machine translation of bi-lingual hindi-english (hinglish) text. *10th Machine Translation summit (MT Summit X), Phuket, Thailand*, pages 149–156.

[Sinha2004] Sinha, R. (2004). An engineering perspective of machine translation: Anglabharti-ii and anubharti-ii architectures. In *Proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS-2004)*, pages 10–17.

[Snover et al.2006] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

[Socher et al.2011] Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

[Spence Green and Manning2014] Spence Green, D. C. and Manning, C. D. (2014). Phrasal: A toolkit for new directions in statistical machine translation. *ACL 2014*, page 114.

[Stolcke and others2002a] Stolcke, A. et al. (2002a). Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

[Stolcke and others2002b] Stolcke, A. et al. (2002b). Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

[Wu et al.2014] Wu, H., Dong, D., He, W., Hu, X., Yu, D., Wu, H., Wang, H., and Liu, T. (2014). Improve statistical machine translation with context-sensitive bilingual semantic embedding model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 142–146.

[Zhang et al.2004] Zhang, Y., Vogel, S., and Waibel, A. (2004). Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *LREC*.

[Zhang et al.2014] Zhang, J., Liu, S., Li, M., Zhou, M., and Zong, C. (2014). Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics*.

[Zou et al.2013] Zou, W. Y., Socher, R., Cer, D. M., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.