

Hybrid System Combination for Machine Translation: An Integration of Phrase-level and Sentence-level Combination Approaches

Wei-Yun Ma

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

COLUMBIA UNIVERSITY

2014

© 2014
Wei-Yun Ma
All rights reserved

Abstract

Hybrid System Combination for Machine Translation: An Integration of Phrase-level and Sentence-level Combination Approaches

Wei-Yun Ma

Given the wide range of successful statistical MT approaches that have emerged recently, it would be beneficial to take advantage of their individual strengths and avoid their individual weaknesses. Multi-Engine Machine Translation (MEMT) attempts to do so by either fusing the output of multiple translation engines or selecting the best translation among them, aiming to improve the overall translation quality. In this thesis, we propose to use the phrase or the sentence as our combination unit instead of the word; three new phrase-level models and one sentence-level model with novel features are proposed. This contrasts with the most popular system combination technique to date which relies on word-level confusion network decoding.

Among the three new phrase-level models, the first one utilizes source sentences and target translation hypotheses to learn hierarchical phrases — phrases that contain subphrases (Chiang 2007). It then re-decodes the source sentences using the hierarchical phrases to combine the results of multiple MT systems. The other two models we propose view combination as a paraphrasing process and use paraphrasing rules. The paraphrasing rules are composed of either string-to-string paraphrases or hierarchical paraphrases, learned from monolingual word alignments between a selected best translation hypothesis and other hypotheses. Our experimental results show that all of the three phrase-level models give superior performance in BLEU compared with the best single translation engine. The two paraphrasing models outperform the re-decoding model and the confusion network baseline model.

The sentence-level model exploits more complex syntactic and semantic information than the phrase-level models. It uses consensus, argument alignment, a supertag-based structural language model and a syntactic error detector. We use our sentence-level model in two ways: the first selects a translated sentence from multiple MT systems as the best translation to serve as a backbone for paraphrasing process; the second makes the final decision among all fused translations generated by the phrase-level models and all translated sentences of multiple MT systems. We proposed two novel hybrid combination structures for the integration of phrase-level and sentence-level combination frameworks in order to utilize the advantages of both frameworks and provide a more diverse set of plausible fused translations to consider.

Table of Contents

List of Figures	iv
List of Tables	vii
Acknowledgments	x
1. Introduction	1
1.1 MT background.....	5
1.2 MEMT Approach	6
1.2 Hybrid Combination	9
1.3 Overview of Thesis Contributions	9
2: Related Work	12
2.1 Confusion Network Decoding Model.....	14
3. Phrase-level Combination: Combination by Re-decoding	19
3.1 Related Work: Phrase-based Re-decoding Model	20
3.1.1 An example	22
3.2 Hierarchical Phrase-based Re-decoding Model.....	25
3.2.1 Hierarchical Phrase Extraction	26
3.2.2 Model	30
3.2.3 Decoding	31
3.2.4 Experiment.....	31
3.3 Conclusions.....	35
4. Phrase-level Combination: Combination by Paraphrasing	36
4.1 Related Work: Lattice Decoding Model	37
4.2 Paraphrasing Model	38

4.2.1 Paraphrase Extraction	42
4.2.2 Model	48
4.2.3 Decoding	49
4.2.4 Experiments	51
4.3 Hierarchical Paraphrasing Model	60
4.3.1 Hierarchical Paraphrase Extraction.....	63
4.3.2 Model	65
4.3.3 Decoding	67
4.3.4 Experiment.....	67
4.4 Conclusions.....	80
5. Sentence-level Combination.....	81
5.1 Related Work.....	82
5.2 Supertagged Dependency Language Model	84
5.2.1 LTAG and Supertag.....	86
5.2.2 Elementary Tree Extraction	86
5.2.3 Model	88
5.2.4 Experiment.....	89
5.3 Syntactic Error Detector	93
5.3.1 Background.....	95
5.3.2 Syntactic Error Detection.....	100
5.3.3 Syntactic Error Correction	105
5.3.4 Experiment.....	105
5.4 Argument Alignment.....	108
5.4.1 Approach	110

5.4.2 Experiment.....	111
5.5 Conclusions.....	112
6. Hybrid Combination.....	114
6.1 Homogeneously Hybrid Combination	115
6.1.1 Experiment.....	117
6.2 Heterogeneously Hybrid Combination	122
6.2.1 Experiment.....	123
6.3 Conclusions.....	126
7. Conclusions.....	127
7.1 Overview of Contributions	128
7.2 Future Work	133
Bibliography.....	135

List of Figures

Figure 1.1: Translation Example

Figure 1.2: The need of word reordering of translations for the task of translation fusion

Figure 1.3: Translation Example

Figure 2.1: Example of Confusion Network decoding

Figure 3.1: The system diagram of Phrase-based Re-decoding Model

Figure 3.2: A source sentence and its two translations provided by MT system h1 and h2.

Figure 3.3: Extracted phrases from the source sentence and the translation by MT system h1

Figure 3.4: Extracted phrases from the source sentence and the translation by MT system h2

Figure 3.5: The system diagram of Hierarchical Phrase-based Re-decoding Model

Figure 3.6: Algorithm of hierarchical phrase extraction for re-decoding

Figure 3.7: A source sentence and its two translations provided by MT system h1 and h2.

Figure 3.8: Extracted hierarchical phrases from the source sentence and the translation by MT system h1

Figure 3.9: Extracted hierarchical phrases from the source sentence and the translation by MT system h2

Figure 3.10: Derivation of a synchronous CFG by using rules in Figure 3.8 and Figure 3.9.

Figure 3.11: Comparing the performance of hierarchical phrase-based re-decoding model with all other systems

Figure 4.1: The system diagram of Paraphrasing Model

Figure 4.2: Comparison of combination models

Figure 4.3: An real alignment example using TERp

Figure 4.4: A backbone E_b and a system hypothesis E_h

Figure 4.5: The alignment between Eb and reordered Eh

Figure 4.6: The alignment between Eb and Eh with the original word order

Figure 4.7: A backbone sentence (the translation Eh1), the translation Eh2 and the word alignment between the two.

Figure 4.8: The extracted phrases from the translation by MT system h1.

Figure 4.9: The extracted phrases from the translation by MT system h2.

Figure 4.10: Different limits for maximum phrase length for NIST Chi-Eng Dataset.

Figure 4.11: The system diagram of Hierarchical Paraphrasing Model

Figure 4.12: Algorithm of hierarchical phrase extraction for paraphrasing

Figure 4.13: A backbone sentence (the translation Eh1), the translation Eh2 and the word alignment between the two.

Figure 4.14: Extracted hierarchical phrases from the source sentence and the translation by MT system h1

Figure 4.15: Extracted hierarchical phrases from the source sentence and the translation by MT system h2

Figure 4.16: Derivation of a synchronous CFG by using rules in Figure 4.14 and Figure 4.15.

Figure 4.17: Comparing the performance of hierarchical paraphrasing model with all other systems for GALE Chi-Eng Dataset.

Figure 4.18: Comparing the performance of hierarchical paraphrasing model with all other systems for NIST Chi-Eng Dataset.

Figure 4.19: Comparing the performance using BLEU of the MT systems, the paraphrasing model and the hierarchical paraphrasing model on the NIST Chi-Eng Dataset.

Figure 4.20: Comparing the performance using TER of the MT systems, the paraphrasing model and the hierarchical paraphrasing model on the NIST Chi-Eng Dataset.

Figure 4.21: Comparing the performance using MET of the MT systems, the paraphrasing model and the hierarchical paraphrasing model on the NIST Chi-Eng Dataset.

Figure 4.22: Hierarchical phrase extraction method D

Figure 4.23: Hierarchical phrase extraction method E

Figure 4.24: Hierarchical phrase extraction method F

Figure 5.1: Parse of “The hungry boys ate dinner”

Figure 5.2: Extracted elementary trees of “The hungry boys ate dinner”

Figure 5.3: Substitution of FB-LTAG

Figure 5.4: Adjunction of FB-LTAG

Figure 5.5: Elementary tree for “saw”

Figure 5.6: Grammatical and Ungrammatical sentences of “saw”

Figure 5.7: Elementary tree for “ask”

Figure 5.8: Grammatical and Ungrammatical sentences of “ask”

Figure 5.9: Parse of “Many young student play basketball”

Figure 5.10: The elementary trees of ‘Many young student play basketball’ and their relations

Figure 5.11: The elementary trees of ‘Many young student play basketball’, their relations and AVMs (simplified version).

Figure 5.12: The reconstructed parse tree with AVMs of the sentence- “Many young student play basketball”

Figure 5.13: Examples of alignments between Chinese arguments and English argument

Figure 6.1: Homogeneously Hybrid Paraphrasing Model

Figure 6.2: Homogeneously Hybrid Hierarchical Paraphrasing Model

Figure 6.3: Heterogeneously Hybrid Combination

List of Tables

Table 2.1: Categories of past methods and my approaches

Table 3.1: Techniques of top five MT of GALE Chi-Eng Dataset

Table 3.2: Techniques of top five MT of NIST Chi-Eng Dataset

Table 3.3: Comparing the performance of hierarchical phrase-based re-decoding model with Top 1 MT system and phrase-based re-decoding model (baseline).

Table 4.1: Techniques of top five MT of NIST Ara-Eng Dataset

Table 4.2: Comparing the performance of the paraphrasing model with others for GALE Chi-Eng Dataset

Table 4.3: Comparing the performance of the paraphrasing model with others for NIST Chi-Eng Dataset.

Table 4.4: Comparing the performance of the paraphrasing model with others for NIST Ara-Eng Dataset

Table 4.5: Comparing the performance of paraphrasing model using different extraction methods for NIST Chi-Eng Dataset

Table 4.6: Comparing the performance of paraphrasing model using different features about syntactic paraphrases for NIST Chi-Eng Dataset.

Table 4.7: The combination performances using the selection of top 5 MT systems and other selections of MT systems on NIST Chi-Eng Dataset.

Table 4.8: The combination performances using the selection of top 3 MT systems and other selections of three MT systems on NIST Chi-Eng Dataset.

Table 4.9: Comparing the performance of the hierarchical paraphrasing model with others for GALE Chi-Eng Dataset.

Table 4.10: Comparing the performance of the hierarchical paraphrasing model with others for NIST Chi-Eng Dataset

Table 4.11: Comparing the performance of the hierarchical paraphrasing model with others for NIST Ara-Eng Dataset

Table 4.12: Comparing the performance of hierarchical paraphrasing model using different extraction methods for NIST Chi-Eng Dataset.

Table 4.13: Comparing the performance of hierarchical paraphrasing model using different features about syntactic paraphrases for NIST Chi-Eng Dataset.

Table 5.1: Result of sentence-level translation combination using SDLM

Table 5.2: Experimental results of human evaluation on 208 different combination results

Table 5.3: Examples of ideal grammatical detection

Table 5.4: Result of sentence-level translation combination using Syntactic Error Detection on NIST Chi-Eng Dataset

Table 5.5: The results of syntactic error detection and correction for GALE Chi-Eng Dataset

Table 5.6: Argument Alignment Mapping Table for PropBank.

Table 5.7: The probabilities of the aligned argument types of the target sentence given an argument type and its predicate of a source sentence.

Table 5.8: Results of using Argument Alignment to select the best translation

Table 6.1: The results of Homogeneously Hybrid Paraphrasing Models on NIST Chi-Eng Dataset

Table 6.2: The results of Homogeneously Hybrid Hierarchical Paraphrasing Models on NIST Chi-Eng Dataset

Table 6.3: TER-based diversity degree of the outputs of paraphrasing model

Table 6.4: TER-based diversity degree of the outputs of hierarchical paraphrasing model

Table 6.5: The results of Homogeneously Hybrid Paraphrasing Models on NIST Ara-Eng Dataset

Table 6.6: The results of Homogeneously Hybrid Paraphrasing Models on NIST Ara-Eng Dataset

Table 6.7: The results of Heterogeneously Hybrid Combination Models on NIST Chi-Eng Dataset

Table 6.8: The results of Heterogeneously Hybrid Combination Models on NIST Ara-Eng Dataset

Table 7.1: The performances of the best models for Phrase-level combination, Sentence-level combination and hybrid combination on NIST Chi-Eng Dataset

Table 7.2: The performances of the best models on NIST Ara-Eng Dataset

Acknowledgments

I want to thank my advisor, Kathleen McKeown for having me as her student. Her thoughtful guidance, constant encouragements and insistence on novelty have had shaped the way I write and do research. She has also supported me in a very practical way—financially. During the years of working with her, I have been backed up by continual project funding. Because of stable financial support, I can concentrate on studying without worries. She also cares for me by meeting up with me on a regular basis. I am truly gratefully and thankful for her.

I would also like to thank the other members of my thesis committee— Nizar Habash, Owen Rambow, Michael Collins and Noémie Elhadad— for the time and effort they generously devoted to the review of this thesis, and for their remarkable breadth and depth of comments and feedbacks. During my time at Columbia, I have benefited from many professors. I am grateful to Nizar Habash for his MT course, which brought me into the MT world, to Owen Rambow for the discussions about tree adjoining grammar and providing the elementary tree extractor, to Michael Collins for his NLP course, to Julia Hirschberg for her encouragements.

I want to thank National Science Foundation for supporting me financially via Grant No. 0910778 entitled “Richer Representations for Machine Translation”. I am fortunate to participate in the joint machine translation project of STAGES (Statistical Translation And GEneration using Semantics). I am grateful to Martha Palmer, Kevin Knight, Dan Gildea and Bert for their help along the way.

I also want to express my appreciation and gratitude to all my friends and colleagues at Columbia University for their help, ideas, and good work: Kapil Thadani, Kristen Parton, Or Biran, Yves Petinot, Sara Rosenthal, Jessica Ouyang, Weiwei Guo and Boyi Xie.

Interactions with people outside Columbia University have also influenced my research. I am

particularly indebted to Yun-Cheng Ju and Xiaodong He at Microsoft Research, who are not only thoughtful, insightful and fun mentors, but also have great influence on the way I think as a scientist.

I am thankful that I am not alone in the Big Apple during such a long time. My wife and son are the source of strength and power for me and my son is the joy of my heart. No matter what frustrations I have, I can overcome them just by seeing his face. Moreover, I also want to thank my parents Nan-Hsien Ma and Peng Li. They rejoiced with every little triumph I had and listened to my every worry and cared and prayed for me. They also gave me a lot of precious advices. I cannot finish this work without my families' love, support, and encouragement. This dissertation is equally their achievement.

I want to thank my church—Reformed Church of Newtown in Elmhurst. The couples of Pastor Su and Pastor Chiu supported me through prayer and delicious meals. Especially on the crossroad of choosing career, they have spent a lot of time to accompany me and talk with me. Brothers and sisters in my church kept praying for me, and they are also my best friends. We have had a lot of fun time and memories. They have made the long years of studying in New York less lonely, less stressful and more enjoyable. Besides, I also want to thank Chen Ge and Chen Jie in Youth Fellowship at Newtown church.

Last but not the least, I want to thank God for His guidance, teach and love. There are times that I am not sure how I can go on; there are also times I do not have faith in myself if I am on the right path; God supported and guided me along every step of the way.

Chapter 1

Introduction

A wide range of successful machine translation (MT) approaches have emerged recently, including phrase-based MT (Koehn et al 2007), hierarchical phrase-based MT (Chiang 2007) and syntax-oriented MT (Galley et al 2006, DeNeefe and Knight 2009). Different MT approaches have their strengths and weaknesses. Multi-Engine Machine Translation (MEMT) attempts to take advantage of strengths and avoid weaknesses by either fusing the output of multiple translation engines or by selecting the best translation among them, aiming to improve the overall translation quality. Recently, many MEMT approaches have been developed in parallel with the rapid development of MT. They play an important role in improving translation quality given the wide range of MT techniques that have emerged.

Figure 1.1 shows an example of selection of the best translation and fusion of the output of multiple translation engines. Given a source sentence in Figure 1.1 (a), we can obtain its different translations from some well-known online translation engines, shown in Figure 1.1 (b), i.e., Google Translate, Bing and Systrans Translate. Different translation engines have their own strengths in some parts of the translation, which are printed in bold and in different colors.

皮埃里还表示,虽然意大利法庭可以进行缺席审判,但意大利警方也不可能到国外把派列娃抓回来服刑。

(Piailli also said that although the Italian court can hold a trial in absentia, the Italian police will not be able to go abroad to catch Pyleva and bring her back for serving the sentence.)

Figure 1.1 (a): A source sentence and its reference translation.



(<https://translate.google.com/>)

Piailli also said that the Italian court in absentia, but the Italian police also impossible to send **Leva** caught **servicing abroad**.



(<http://www.bing.com/translator/>)


Alvaro pierri, Italy court trials **in absentia, but Italy police is unlikely to back** Vera Zvonareva served abroad.



(<http://www.systransoft.com/>)

Pieri also said that although the Italian court can carry on the trial by default, but Italian Police as impossible to **grasp** to serve **a prison sentence** Pailiewa as the overseas.

Figure 1.1 (b): Translations of the source sentence in Figure 1.1 (a) by Google Translate, Bing Translate and Systran Translate.

Ideal selection (Select translation of  SYSTRANet):

Pieri also said that although the Italian court can carry on the trial by default, but Italian Police as impossible to grasp to serve a prison sentence Pailiewa as the overseas.

Ideal fusion:

Pieri also said that although the Italian court can carry on the trial in absentia, but Italy police is unlikely to grasp Leva abroad back serving a prison sentence

Figure 1.1 (c): Ideal selection and Ideal fusion given the translations in Figure 1.1 (b)

Assume our task is to select the best translation among the three translation engines for this sentence. Although all of the three translations are very poor translations for this sentence, the translation by Systran Translate is relatively closer to the original meaning of the source sentence and more understandable than the other two engines. So the ideal selection could be the translation by Systran Translate, shown in Figure 1.1 (c). And at least the first clause of the translation by Systran Translate has a verb, while the other two engines have no verb in their first clauses at all.

If our task is to fuse the translations by the three translation engines for this sentence, the ideal fusion result could be the translation in Figure 1.1 (c), where the better parts of the three translation engines, shown in different colors, are fused to form a new translation. Although it is still not a perfect grammatical translation, it is already very close to an understandable translation, where every word comes from the three very poor translations. Taking a closer look at the second clause of the fusion translation, we can find the ideal fusion actually requires some word reordering, shown in Figure 1.2. This observation suggests that a good fusion model is not only responsible for interleaving strings, but also for dealing with word reordering that can involve transposing words.

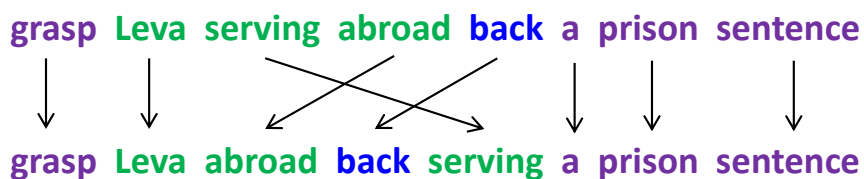


Figure 1.2: The need of word reordering of translations for the task of translation fusion.

The translations provided by the three online systems in Figure 1.1 are actually pretty poor translations. The ideal fusion shows that there is a possibility to improve them and thus provide a relatively acceptable, but still far from perfect translation. In the next example, shown in Figure

1.3, the translations provided by the three online systems are close to perfect translations. The ideal fusion shows that there is a possibility to produce a perfect translation based on fusing these systems' translations.

69 歲的莫迪亞諾在法國是知名作家,但在他國較鮮為人知。
(69-year-old Modiano is a famous writer in France, but less well known in other countries.)

Figure 1.3 (a): A source sentence and its reference translation.



(<https://translate.google.com/>)

69-year-old Modiano is well-known writer in France, but relatively little-known in his country.



(<http://www.bing.com/translator/>)

69 Modiano in France is a famous writer, but less well known in other countries.



(<http://www.systransoft.com/>)

69-year-old Modiano in France is noted author, but is rarely known in other country.

Figure 1.3 (b): Translations of the source sentence in Figure 1.3 (a) by Google Translate, Bing Translate and Systran Translate.

Ideal selection (Select translation of [bing](#)):

69 Modiano in France is a famous writer, but less well known in other countries.

Ideal fusion:

69-year-old [Modiano is a famous writer in France, but less well known in other countries.](#)

Figure 1.3 (c): Ideal selection and Ideal fusion given the translations in Figure 1.3 (b)

Assume our task is to select the best translation among the three translation engines for this sentence. The translation by Bing Translate is relatively closer to the original meaning of the source sentence and more understandable than the other two engines. So the ideal selection could be the translation by Bing Translate, shown in Figure 1.3 (c). If our task is to fuse the translations by the three translation engines for this sentence, the ideal fusion result could be the translation in Figure 1.3 (c), where the better parts of the three translation engines, shown in different colors, are fused to form a new translation, which turns out to be a perfect translation. In fact, the translation by Bing Translate is already very close to the reference except it makes the translation mistake of “69-year-old”. By using that part of “69-year-old” provided by Google Translate, the mistake can be fully fixed.

1.1 MT background

Initially, MT systems were built by computational linguists. These rule-based MT systems relied on hand-built translation rules to do the translation. However, in recent years, as parallel corpora and monolingual corpora became more and more available, statistical machine translation (SMT) models became the state-of-the-art in MT. They use machine learning techniques to automatically learn translation rules from parallel corpora and monolingual corpora.

SMT models range widely, and they can be divided into three categories based on their translation models, including phrase-based MT (Koehn et al 2003), hierarchical phrase-based MT (Chiang 2007) and syntax-based MT (Galley et al 2006, DeNeeffe and Knight 2009).

Phrase-based MT: The term “phrase” indicates a string of words which is not necessary a linguistic unit. So a translation rule is basically a mapping between a word string in source and a word string in target. By capturing the mappings of word strings, a phrase-based MT model is

able to exploit context to reduce translation ambiguity.

Hierarchical phrase-based MT: The term “hierarchical phrase” indicates a phrase (a word string) that contain subphrases. Hierarchical phrase-based MT uses a synchronous context-free grammar dynamically learned from source sentence and target hypotheses to represent the translation rules. It directly models possible word re-orderings in the translation rules, whereas Phrase-based MT phrase-based SMT systems typically model word reordering within a fixed window.

Syntax-based MT: The translation rules consist of the mappings from syntactically well-formed trees to strings or vice versa, or the mapping from syntactically well-formed trees to syntactically well-formed trees (tree-to-tree) The motivation for syntax-based models is that their translation rules should be more accurate mappings, and thus, string-to-tree and tree-to-tree models should produce more fluent translations because the target side is constrained to be syntactically well-formed trees. But one of the biggest challenges of Syntax-based MT is that it could include too strict constraints on translation rules, compared with Phrase-based MT and Hierarchical phrase-based MT. This results in a relatively small number of translation rules are and thus could possibly lack some reasonable translation information.

1.2 MEMT Approach

MEMT approaches can be classified into three types based on the unit of fusion – word, phrase and sentence. The word-level fusion framework, such as the *confusion network decoding model*, is the most popular approach (Matusov et al., 2006; Rosti et al., 2007b; He et al. 2008; Karakos et al. 2008; Sim et al. 2007; Xu et al. 2011). However, using the word as the unit of fusion rather

than the phrase, has a higher risk of breaking coherence and consistency between the words in a phrase. In addition, it is difficult to consider syntax and semantics in a word-level fusion framework because the minimum unit of syntactic and semantic analysis is a phrase or a sentence rather than a word. Therefore, in addition to word-level combination approaches, some phrase-level combination approaches have also recently been developed with the goal of retaining coherence and consistency between the words in a phrase.

The most common phrase-level combination approaches are re-decoding methods: by constructing a new phrase translation table from each MT system’s source-to-target phrase alignments, the source sentence can also be re-decoded using the new translation table (Rosti et al., 2007a; Huang and Papineni, 2007; Chen et al., 2007b; Chen et al., 2009b). We call this strategy the *phrase-based re-decoding model*. One of the challenges with these approaches is that, with a new phrase table, the translated word order is computed entirely by the reordering model of the re-decoder, which usually only has the capability of local reordering and does not fully utilize existing information about word reordering present in the target hypotheses; thus these approaches lack the ability to reorder words across long distances. To address the problem, in this thesis, we propose the use of hierarchical phrases — phrases that contain subphrases (Chiang 2007) — for re-decoding-based combination. We learn hierarchical phrases from each MT system’s source-to-target phrase alignments and rely on the hierarchical phrases to directly model possible word re-orderings. We call this technique the *hierarchical phrase-based re-decoding model*. Our experiments show that it improves over the baseline combination technique of the *phrase-based re-decoding model*.

Another phrase-level combination approach relies on a *lattice decoding model* to carry out the combination (Feng et al 2009; Du and Way 2010). In a lattice, each edge is associated with a phrase (a single word or a sequence of words) rather than a single word. The construction of the

lattice is based on the extraction of phrase pairs from word alignments between a selected best MT system hypothesis (the backbone) and the other translation hypotheses. Feng et al (2009) designed some heuristic rules to extract phrase pairs while Du and Way (2010) rely on TER-Plus (TERp) to extract certain types of phrase pairs. For lattice decoding models, the word order of the backbone determines the word order of consensus outputs, thus they are able to use the existing word ordering of the backbone; however, lattice decoding models lack the ability to reorder words of the backbone.

To improve these models, in this thesis, we propose another phrase-level combination approach, called the *paraphrasing model* (Ma and McKeown. 2012a). It extracts string-to-string paraphrases from the backbone and other hypotheses, and then uses these paraphrases to paraphrase the backbone. A reordering model can be integrated into the paraphrasing model. In order to further capture more complicated paraphrasing phenomena between the backbone and other target hypotheses, such as longer phrase reordering or the occurrences of discontinuous phrases, we also propose the use of hierarchical phrases — phrases that contain subphrases (Chiang 2007) — for paraphrasing-based combination. We learn hierarchical paraphrases from monolingual word alignments between a selected backbone hypothesis and other hypotheses. These hierarchical paraphrases can model more complicated paraphrasing phenomena, and thus enable more utilization of consensus among MT engines than non-hierarchical paraphrases do. We call this technique the *hierarchical paraphrasing model*.

As for our sentence-level model, because the whole sentence can be used to evaluate the translation quality, it is easier to integrate more sophisticated syntactic and semantic features. We do relatively deeper analysis to evaluate the translation quality and represent our syntactic and semantic features in a log linear model. We hypothesize that, for a good translation, most of the predicate-argument structures are retained in order to preserve the semantics. That is,

predicate-argument structures and argument types in source and target should be the same in most cases. Based on this assumption, we develop several measures of how likely arguments are to be aligned. In addition, in order to identify ungrammatical hypotheses from a set of candidate translations, we utilize grammatical knowledge in the target language, including using a supertag-based structural language model that expresses syntactic dependencies between words, and a syntactic error detector based on a feature-based lexicalized tree adjoining grammar (FB-LTAG) to recognize ungrammatical translations.

1.2 Hybrid Combination

We design two hybrid combination structures for the integration of phrase-level and sentence-level combination frameworks in order to utilize the advantages of both frameworks and provide a more diverse set of plausible fused translations to consider.

The first structure is the *homogeneously hybrid combination*, where the same phrase-based techniques is used to generate outputs for the sentence-level combination component to select, and the other structure is *heterogeneously hybrid combination*, where different phrase-based techniques are used to generate outputs for the sentence-level combination component to select.

1.3 Overview of Thesis Contributions

Our contributions for the MT combination research community include:

1. Novel Models

We propose three novel phrase-level models. For the re-decoding combination framework, we present a hierarchical phrase-based decoding technique, based on synchronous context-free grammar, in order to better model work reordering information provided by

various MT translations and enable more utilization of consensus among MT engines. For the paraphrasing combination framework, two new paraphrasing methods are presented to paraphrase the backbone translation hypothesis: one uses string-to-string paraphrases and the other utilizes hierarchical paraphrases. Either kind of paraphrase is learned from monolingual word alignments between a selected backbone hypothesis and other hypotheses.

2. Novel Features

For the sentence-level model, we present novel syntactic and semantic features in a log linear model to evaluate the quality of a translation hypothesis. Our new features include argument alignments, a supertag-based structural language model and a syntactic error detector.

3. Phrase Level V.S. Word Level

We want to compare both fusion units under the same feature settings. Our expectation is that phrase is a more reasonable unit for fusion than word because it can carry more syntactic and semantic information with it. By setting the phrase length to be one, we can get the word-level version of each phrase-level model. In addition to comparing our phrase-level model with a word-level model, we also investigate the impact of phrase length in our models.

4. Hybrid Architectures

We propose two different hybrid combination architectures to integrate our phrase-level models and a sentence-level model. Our experimental results demonstrate that this

integration can yield an improvement in results.

Chapter 2

Related Work

In the past several years, many machine translation (MT) combination approaches have been developed. According to (Rosti et al., 2012), system combination methods proposed in the literature can be roughly divided into three categories: (i) hypothesis selection (Rosti et al., 2007a; Hildebrand and Vogel, 2008), (ii) re-decoding (Frederking and Nirenburg, 1994; Jayaraman and Lavie, 2005; Rosti et al., 2007a; He and Toutanova, 2009; Devlin et al., 2011), and (iii) confusion network decoding (Matusov et al 2005, Rosti et al 2007b). This division is a good summary of the past major methods proposed in the literature, but it lacks two dimensions: *lattice decoding model* (Feng et al 2009, Du and Way 2010) and *paraphrasing model*, proposed in this thesis. We use Table 2.1 to summarize our methods and past system combination methods according to different fusion units.

Table 2.1 shows that we proposed some novel models for phrase-level combination and new features for sentence-level combination. These include a hierarchical-phrase model based on redecoding, two novel paraphrasing approaches and a sentence-level model based on some new features of the evaluation of translation quality. In this section, we would introduce Confusion Network decoding only and leave other related approaches to be introduced under relevant subsections later on.

	word	phrase	hierarchical phrase	sentence
Hypothesis Selection model	-	-	-	Hildebrand and Vogel 2008 Callison-Burch et al., 2012 This thesis (Ma and McKeown, 2012b; 2012c;2013)
Re-decoding	-	Rosti et al., 2007a Huang and Papineni, 2007 Chen et al., 2007b Chen et al., 2009	This thesis (Ma and McKeown, submitted 2014)	-
Confusion Network Decoding model	Matusov et al., 2006 Rosti et al., 2007b He et al. 2008 Xu et al. 2011 Chen et al. 2012 ...	-	-	-
Lattice Decoding model	-	Feng et al 2009 Du and Way 2010	-	-
Paraphrasing model	-	This thesis (Ma and McKeown, 2012a)	This thesis (Ma and McKeown, submitted 2014)	-

Table 2.1: Categories of past methods and my approaches

2.1 Confusion Network Decoding Model

Confusion Network decoding is one of the most popular approaches (Matusov et al., 2006; Rosti et al., 2007b; He et al. 2008; Karakos et al. 2008; Sim et al. 2007; Xu et al. 2011, Chen et al. 2009a). Chen et al. (2009a) divides Confusion Network decoding into four steps: 1. Backbone selection: to select a backbone (also called “skeleton”) from all hypotheses. The backbone defines the word orders of the final translation. 2. Hypothesis alignment: to build word alignment between backbone and each hypothesis. 3. Confusion network construction: to build a confusion network based on hypothesis alignments. 4. Confusion network decoding: to decode the best translation from a confusion network. In the following, we explain each step and highlight the difference between Confusion Network decoding and our approaches, illustrating the process using the example in Figure 2.1.

Backbone selection: As the selected backbone determines the word orders of the final fusion translation, the quality of the combination output also depends on which hypothesis is chosen as the backbone. The common selection strategy is through Minimum Bayes Risk (MBR) decoding (Sim et al., 2007; Rosti et al., 2007b; He et al 2008). The basic idea is to choose the hypothesis that best agrees with other hypotheses on average as the backbone. Translation edit rate (TER) (Snover et al., 2006) or modified BLEU score are often used as the loss function in MBR decoding. Taking TER score as the example, the hypothesis resulting in the lowest average TER score:

$$E_b = \operatorname{argmin}_{\hat{E} \in H} \sum_{E \in H} TER(\hat{E}, E)$$

$$TER(\hat{E}, E) = \frac{\#Insertion + \#Deletion + \#Substitution + \#Shift}{length(\hat{E})}$$

where H is a hypotheses set; *Insertion*, *Deletion*, *Substitution* are three different kinds of word edit. *Shift* is a shift of a sequence of words and it is counted as a single edit. The minimum translation edit alignment is found through a beam search.

In the Figure 2.1 example, assume we are given three different hypotheses. Each of them is coming from a certain MT system, i.e, Sys1, Sys2 and Sys3. After backbone selection, we assume the hypothesis of Sys1 is selected based on MBR decoding.

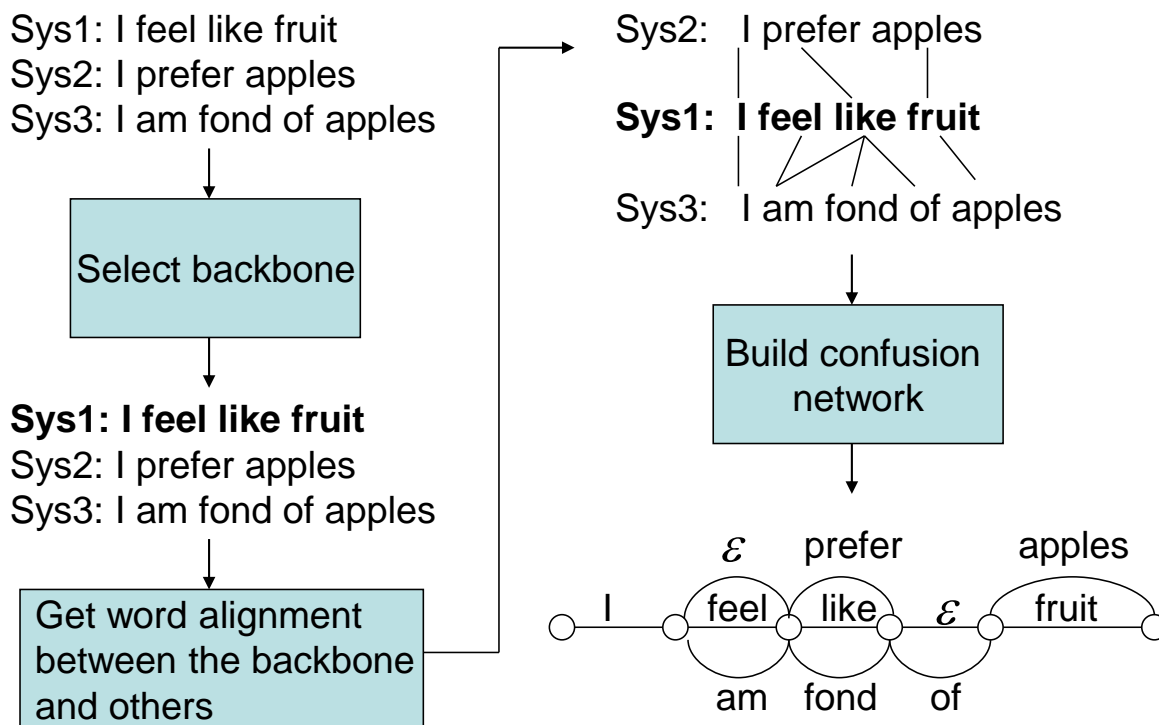


Figure 2.1: Example of Confusion Network decoding

Besides TER-based MBR decoding, in section 4.6, we also investigate the effect of utilizing our sentence-level model as the backbone selection module for our phrase-level combination approaches and analyze their performances.

Hypothesis alignment: After selecting the backbone, the next step is to obtain the word alignments between the backbone and all other system hypotheses in order to construct the confusion network. Many techniques have been studied to address this issue. Bangalore et al. (2001) utilized an edit distance alignment algorithm for this task, and it only allows monotonic alignment. Jayaraman and Lavie (2005) proposed a heuristic-based matching algorithm which allows nonmonotonic alignments to align the words. More recently, Matusov et al. (2006, 2008) used GIZA++ to produce word alignments of hypotheses pairs. Sim et al. (2007), Rosti et al. (2007a), and Rosti et al. (2007b) depend on the TER alignment toolkit to obtain word alignments. Karakos et al. (2008) used an ITG-based method to produce word alignments. He et al. (2008) proposed an IHMM-based word alignment method which the parameters are estimated indirectly from a variety of sources. Chen et al. (2009a) and Rosti et al. (2012) did systematic comparisons of these well known hypothesis alignment algorithms for MT system combination via confusion network decoding, and both of them found IHMM-based word alignment method can achieve the best performance.

Our research is not focusing on the design of hypothesis alignment algorithms. In our phrase-level combination models, any hypothesis alignment algorithm can be used, so we adopt TERp-based word alignment toolkit to serve our mission, which is a released toolkit and has similar performance close to IHMM-based word alignment method.

In the Figure 2.1 example, after hypothesis alignment, both “I” of Sys2 and Sys3 align to “I” of Sys1; “prefer” of Sys2 aligns to “like” of Sys1; “am” of Sys3 aligns to “feel like” of Sys1; “am fond of ” of Sys3 aligns to “like” of Sys1; both “apples” of Sys2 and Sys3 align with “fruit” of Sys1.

Confusion Network Construction: Hypothesis alignments algorithms, such as GIZA++ and IHMM-based word alignment methods, produce n-to-1 mappings between the hypothesis and

backbone. But because confusion network is built from one-to-one word alignments, the word alignments need to be normalized to one-to-one word alignment by removing duplicated links before constructing the confusion network. Researchers usually implement that by keeping the highest similarity measure based on a certain score function. For example, in Figure 2.1, “am fond of ” of Sys3 aligns to “like” of Sys1, and if the similarity measure of “fond” and “like” is higher than either the similarity measure of “am” and “like” or the similarity measure of “am” and “of”, the link of “am” and “like” and the link of “of” and “like” will be removed. After normalizing n-to-1 word alignment to one-to-one word alignment, the hypothesis words need to be reordered to match the word order of the backbone according to their alignment indices. To reorder the null-aligned words, we need to first insert the *null* words into the proper position in the backbone and then reorder the null-aligned hypothesis words to match the *nulls* on the backbone side. For example, in Figure 2.1, “of” of Sys3 aligns to an inserted *null* word of Sys2, which is between “like” and “fruit”.

Given the monotone one-to-one word alignments of hypotheses, the transformation to a confusion network as described by (Bangalore et al., 2001) is straightforward. It is explained by the example in Figure 2.1. Each arc represents an alternative word at that position in the sentence.

In Figure 2.1, we find that although “am fond of ” and “feel like” have the same meaning, the confusion network-based approaches face the risk of producing degenerate translations, such as “am like of” and “feel fond of”. In our phrase-level combination models, we use the phrase as the fusion unit instead of the word, and fully utilize the information of these n-to-1 mappings between the hypothesis and backbone to form the phrases. In other words, we fully utilize the information that “am like of” and “feel fond of” have the same meaning and are not supposed to be separated. Therefore, in our phrase-level combination models, the step of normalization of n-to-1

word alignment to one-to-one word alignment is not necessary. We will illustrate this point in detail in later sections.

Confusion Network Decoding: Confusion network decoding aims to find the path with the highest confidence in the network. The path is extracted from the confusion network through a beam-search algorithm with a log-linear combination of a set of feature functions. The feature functions which are usually employed in the search process include a language model, word penalty, votes on word arcs and N-gram posterior probabilities (Zens and Ney, 2006). The weights of feature functions are optimized to maximize the scoring measure (Och, 2003).

Because confusion network decoding is a word-level fusion framework, it is difficult to integrate syntax and semantics in the design of feature functions. That is one of our motivations of developing the phrase-level approaches, described in the following sections.

Chapter 3

Phrase-level Combination: Combination by Re-decoding

Confusion networks require one-to-one word alignment between the words of hypotheses, so they have difficulty in handling the common phenomenon in which several words are connected to another several words. For example, in Figure 2.1, “am fond of ” and “feel like” are paraphrases and are not supposed be separated. Based on this motivation, phrase-level combination approaches have also been developed recently. Their goal is to retain coherence and consistency between the words in a phrase. Phrase-level approaches can be classified according to whether they use information from the source (re-decoding methods) or whether they paraphrase the target. They are described in Chapter 3 and Chapter 4, respectively, and we propose our novel models in both categories.

Re-translation to combine MT outputs is the most common phrase-level combination approaches. By collecting or extracting MT system’s source-to-target phrase alignments, one can re-decode the source sentence using information from phrase alignments. Section 3.1 will introduce related work in this division and also highlight the difficulties that these approaches face, followed by our motivation and proposed solution - Hierarchical Phrase-based Re-decoding

Model, described in Section 3.2.

3.1 Related Work: Phrase-based Re-decoding Model

Most phrase-level combination approaches rely on the strategy of source re-decoding: by constructing a new phrase translation table from each MT system’s source-to-target phrase alignments, they re-decode the source sentence using the new translation table (Rosti et al., 2007a; Huang and Papineni, 2007; Chen et al., 2007b; Chen et al., 2009b). We call this strategy the *phrase-based re-decoding model*, which system diagram is shown in Figure 3.1.

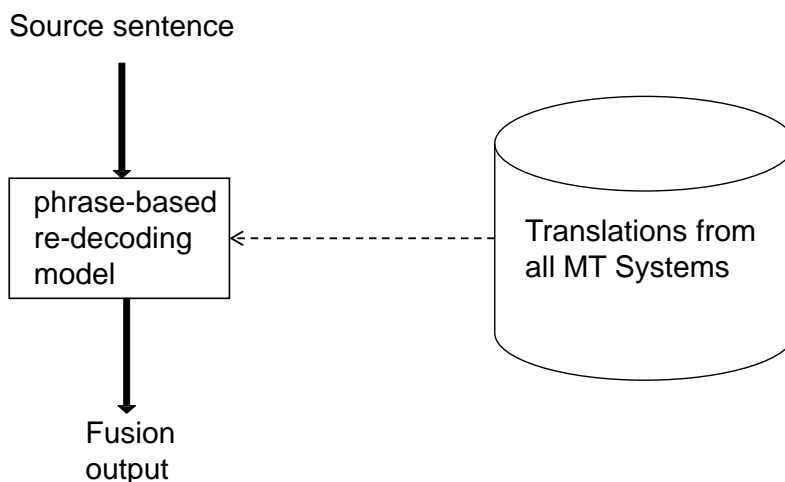


Figure 3.1: The system diagram of Phrase-based Re-decoding Model

Take the same example given in Figure 2.1. Assume both “am fond of” and “feel like” aligns to the same source phrase in Chinese – “喜歡”. After re-decoding “喜歡” in Chinese, the output will be either “am fond of” or “feel like”.

The source-to-target phrase alignments could be available from the individual systems (Rosti et al., 2007a). If the phrase alignments are not available, they can be extracted by applying the standard phrase extraction rules (Chen et al., 2009). The standard phrase extraction rules (Koehn et al., 2003) aim to extract all phrases that are word-continuous and consistent with word alignments, which are automatically generated; for example, by using GIZA++ (Och and Ney,

2003). This means that words in a legal phrase pair are not aligned to words outside of the phrase pair, and should include at least one pair of words aligned with each other.

(Koehn et al., 2003)’s definition of consistency can be formally stated as follows: assume there is a source sentence F and a MT system hypothesis \bar{E} . f is a phrase of F , and \bar{e} is a phrase of \bar{E} . A phrase pair (f, \bar{e}) is consistent with the word alignment matrix A if

$$\forall w_i \in f : (w_i, x) \in A \Rightarrow x \in \bar{e}$$

$$\text{and } \forall \bar{w}_j \in \bar{e} : (y, \bar{w}_j) \in A \Rightarrow y \in f$$

$$\text{and } \exists w_i \in f, \bar{w}_j \in \bar{e} : (w_i, \bar{w}_j) \in A$$

where w_i is a word of f , \bar{w}_j is a word of \bar{e} .

Once obtaining the source-to-target phrase alignments and constructing the new translation table, the definition of confidence scores for phrases in the translation table plays a crucial role. For example, Rosti et al., (2007a) derive the confidence scores from sentence posteriors with system-specific total score scaling factors and similarity scores based on the agreement among the phrases from all systems. The agreement is measured by levels of similarity. The confidence of the phrase table entry is increased if several systems agree on the target words. The phrasal decoder used in the phrase-level combination is based on standard beam search, and their decoder features include a trigram language model score, number of target phrases, number of target words, phrase distortion, phrase distortion computed over the original translations and phrase translation confidences. The total score for a hypothesis is computed as a log-linear combination of these features.

One of the challenges with these approaches is that, with a new phrase table, the translated word order is computed entirely by the reordering model of the re-decoder, which usually only has the capability of local reordering and does not fully utilize existing information about word

reordering present in the target hypotheses; thus they lack the ability to record word reordering across long distances. Especially when different MT systems usually have different reordering models, it is common that words in the source sentence would be translated in different orders for different MT systems. Researchers have studied this problem through a reordering cost function that encourages search along with decoding paths from all MT engines' decoders (Huang and Papineni 2007). However, to the best of our knowledge, no one has investigated using more powerful grammars of translation rules able to directly model the information of existing word reordering of the target hypotheses.

3.1.1 An example

We use the Chinese-to-English example of Figure 3.2 to illustrate the re-decoding process. Assume we are given a Chinese sentence – “他喜歡你買的書 (He likes the book that you bought)”, the translation provided by MT system h1 – “He likes you buy the book” and the translation provided by MT system h2 – “He like the books that you bought”, we can obtain the word alignments between the source sentence and the two translation by using GIZA++ on the corresponding corpus. Phrases can then be extracted from the given source-to-target word alignments by using the standard bilingual phrase extraction rules (Koehn et al, 2003), shown in Figure 3.3 and Figure 3.4.

If we re-decode the source using a phrase-based decoder without any reordering model, the best translation we can get is “He likes the books that you bought” by using the rule from “<他喜歡 , He likes>” from MT system h1 and the rule “<你 買 的 書 , the books that you bought >” from MT system h2. The mistake of the translation is that “books” should be “book” and this mistake is due to the lack of the reordering ability.

他 喜歡 你 買 的 書
 He likes you bought 's book
 (He likes the book that you bought)

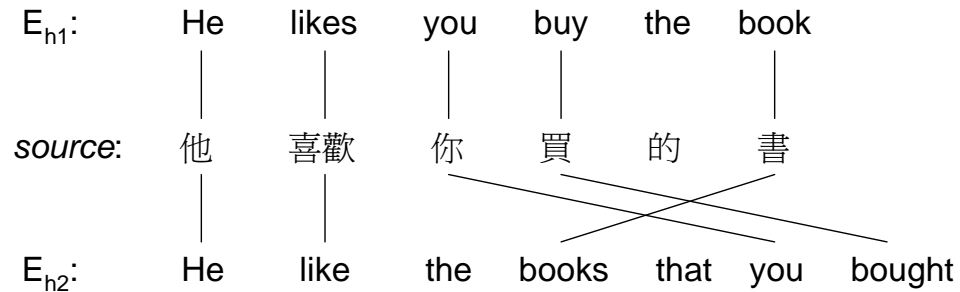


Figure 3.2: A source sentence and its two translations provided by MT system h1 and h2.

- | | |
|---|-------------------------------|
| < 他 , He > | < 你 , you > |
| < 他 喜歡 , He likes > | < 你 買 , you buy > |
| < 他 喜歡 你 , He likes you > | < 你 買 , you buy the > |
| < 他 喜歡 你 買 , He likes you buy > | < 你 買的 , you buy > |
| < 他 喜歡 你 買 , He likes you buy the > | < 你 買的 , you buy the > |
| < 他 喜歡 你 買的 , He likes you buy > | < 你 買的 書 , you buy the book > |
| < 他 喜歡 你 買的 , He likes you buy the > | < 買 , buy > |
| < 他 喜歡 你 買的 書 , He likes you buy the book > | < 買 , buy the > |
| < 喜歡 , likes > | < 買的 , buy > |
| < 喜歡 你 , likes you > | < 買的 , buy the > |
| < 喜歡 你 買 , likes you buy > | < 買的 書 , buy the book > |
| < 喜歡 你 買 , likes you buy the > | < 的 書 , the book > |
| < 喜歡 你 買的 , likes you buy > | < 的 書 , book > |
| < 喜歡 你 買的 , likes you buy the > | < 書 , book > |
| < 喜歡 你 買的 書 , likes you buy the book > | |

Figure 3.3: Extracted phrases from the source sentence and the translation by MT system h1

< 他 , He >	< 你買的, you bought >
< 他 喜歡 , He like >	< 你買的, that you bought >
< 他 喜歡 你買的書, He like the book that you bought >	< 你買的書, books that you bought >
< 喜歡 , like >	< 你買的書, the books that you bought >
< 喜歡 你買的書, like the books that you bought >	< 買的, bought >
< 你, you >	< 買, bought >
< 你買, you bought >	< 的書, books >
< 你買, that you bought >	< 的書, the books >
	< 書, books >
	< 書, the books >

Figure 3.4: Extracted phrases from the source sentence and the translation by MT system h2

If the phrase-based decoder has a reordering model, it will have a chance of getting the correct translation - “He likes the book that you bought” by using the rule of “<他 喜歡, He likes>” from MT system h1, the rule of “<你 買 的 , that you bought >” from MT system h2 and the rule of “<書, book >” from MT system h1. However, because of the concern of time complexity, most phrase-based decoders only allow limited reordering, such as the *relative distance reordering model*, which restricts reordering to short local movements or permit moves within a window of a few words. Thus, for long-reordering phenomena, such as the pattern “與 (with)...有(have)...” in Chinese (Chiang 2005) or verb-final grammar (the verbs occurs at the end of the sentence) in Japanese or German, limited reordering often fails to produce a good translation. This is a particular problem for verb-final grammar, where the decoder needs to move the verb from the end of the sentence to the position just after the subject at the beginning of the sentence; that move could be over a large number of words, leading to be penalized heavily by the relative distance reordering model (Koehn 2010). In the next section, in order to address this issue and increase the diversity of consensus patterns, we will propose our solution,

which follows (Chiang 2007)’s hierarchical phrase-based model for statistical machine translation.

3.2 Hierarchical Phrase-based Re-decoding Model

In this section, we propose the use of hierarchical phrases—phrases that contain subphrases (Chiang 2007) and the use of a synchronous context-free grammar dynamically learned from source sentence and target hypotheses to represent the translation information. We learn hierarchical phrases from each MT system’s source-to-target phrase alignments and rely on the phrases to directly model possible word re-orderings. Through re-decoding the source sentence with the hierarchical phrases, we are able to obtain the combination result. We call this technique the *hierarchical phrase-based re-decoding model*, which system diagram is shown in Figure 3.5.

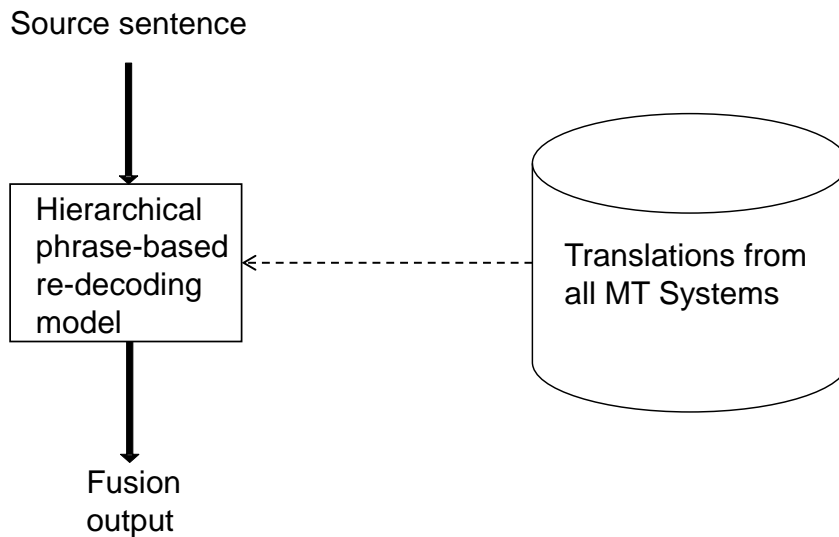


Figure 3.5: The system diagram of Hierarchical Phrase-based Re-decoding Model

The combination process involves the following steps:

1. Collect the translation hypotheses from multiple MT systems. In our work, the source-to-target word alignments are available from the individual systems. If the word alignments are not available, they can be automatically generated using GIZA++ (Och and Ney, 2003).
2. Extract phrases from the given source-to-target word alignments. We follow the standard bilingual phrase extraction rules (Koehn et al, 2003): we extract all phrases that are word-continuous and consistent with the word alignment for each MT system.
3. Extract hierarchical phrases from the given extracted phrases in step 2. The formal extraction algorithm is provided in Section 3.2.1.
4. Assign each hierarchical phrase a confidence estimation, as described in Section 3.2.2.
5. Re-decode the source using the extracted hierarchical phrases with confidence estimations as described in Section 3.2.3.

3.2.1 Hierarchical Phrase Extraction

We formulate our hierarchical phrase extraction as a weighted synchronous context-free grammar (SCFG). A formal definition of a synchronous CFG (Aho and Ullman, 1969) takes the form:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle$$

where X is any non-terminal in the grammar; γ and α are strings of terminals and non-terminals; \sim is a one-to-one correspondence between non-terminals in γ and non-terminals in α . Each rule in a synchronous CFG is a rewrite rule with aligned pairs of right-hand sides. At each step, two coindexed non-terminals are rewritten using the two components of a rule.

Following Chiang (2007), our hierarchical phrase-level translation rules are designed as a synchronous-CFG, extracted from the source sentences and the given translations of multiple MT systems.

For the i -th sentence, we use F^i and f^i to represent the source sentence and one of its phrases, respectively. E_h^i represents the translation of MT system h , and e_h^i is one phrase of E_h^i . We use T_h^i to denote the set of translation rules for the i -th sentence and MT system h , and show how to collect T_h^i as follows:

If $\langle f^i, e_h^i \rangle$ is consistent with word alignment, then $X \rightarrow \langle f^i, e_h^i \rangle$ is added to T_h^i .

If $X \rightarrow \langle \gamma, \alpha \rangle$ is a rule in T_h^i , and $\langle f^i, e_h^i \rangle$ is consistent with monolingual word alignment

such that $\gamma = \gamma_1 f^i \gamma_2$ and $\alpha = \alpha_1 e_h^i \alpha_2$ then $X \rightarrow \langle \gamma_1 X_k \gamma_2, \alpha_1 X_k \alpha_2 \rangle$ is added to T_h^i ,

where k is an index.

Then we add the following two special “glue” rules to T_h^i

$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle, \quad S \rightarrow \langle X_1, X_1 \rangle$

Figure 3.6: Algorithm of hierarchical phrase extraction for re-decoding

We use the following Chinese-to-English example to show the results of using the extraction algorithm.

他 喜歡 你 買 的 書
 He likes you bought 's book
 (He likes the book that you bought)

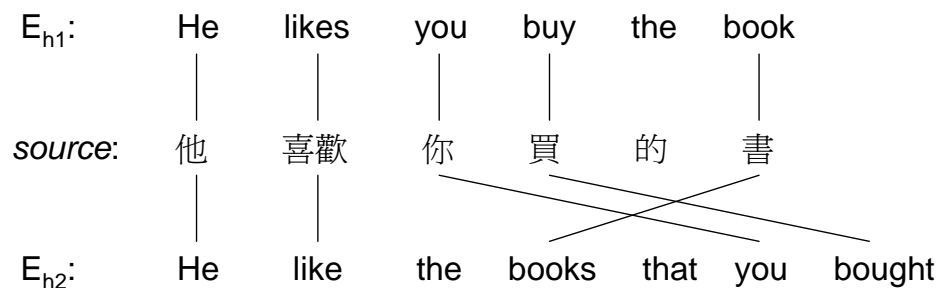


Figure 3.7: A source sentence and its two translations provided by MT system h1 and h2.

- $S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$ (1)
- $S \rightarrow \langle X_1, X_1 \rangle$ (2)
- $X \rightarrow \langle \text{他 喜歡}, \text{He likes} \rangle$ (3)
-
- $X \rightarrow \langle \text{你 買 的 書}, \text{you buy the book} \rangle$ (4)
- $X \rightarrow \langle \text{你 買}, \text{you buy} \rangle$ (5)
- $X \rightarrow \langle \text{書}, \text{book} \rangle$ (6)
- $X \rightarrow \langle X_1 \text{ 的 書}, X_1 \text{ the book} \rangle$ (7)
- $X \rightarrow \langle \text{你 買 的 } X_1, \text{you buy the } X_1 \rangle$ (8)
- $X \rightarrow \langle X_1 \text{ 的 } X_2, X_1 \text{ the } X_2 \rangle$ (9)

Figure 3.8: Extracted hierarchical phrases from the source sentence and the translation by MT system h1

$$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle \quad (10)$$

$$S \rightarrow \langle X_1, X_1 \rangle \quad (11)$$

$$X \rightarrow \langle \text{他 喜歡}, \text{He like} \rangle \quad (12)$$

.....

$$X \rightarrow \langle \text{你 買 的 書}, \text{the books that you bought} \rangle \quad (13)$$

$$X \rightarrow \langle \text{你 買}, \text{you bought} \rangle \quad (14)$$

$$X \rightarrow \langle \text{書}, \text{books} \rangle \quad (15)$$

$$X \rightarrow \langle X_1 \text{ 的 書}, \text{the books that } X_1 \rangle \quad (16)$$

$$X \rightarrow \langle \text{你 買 的 } X_1, \text{the } X_1 \text{ that you bought} \rangle \quad (17)$$

$$X \rightarrow \langle X_1 \text{ 的 } X_2, \text{the } X_2 \text{ that } X_1 \rangle \quad (18)$$

Figure 3.9: Extracted hierarchical phrases from the source sentence and the translation by MT system h2

Given the extracted hierarchical phrase of Figure 3.8 and Figure 3.9, *hierarchical phrase-based re-decoding model* would have the chance of getting the correct translation - “He likes the book that you bought”. Figure 3.10 shows the derivation of a synchronous CFG by using rules in Figure 3.8 and Figure 3.9.

$$\begin{aligned} \langle S_1, S_1 \rangle &\Rightarrow \langle S_2 X_3, S_2 X_3 \rangle && \text{using (1) or (9)} \\ &\Rightarrow \langle X_4 X_3, X_4 X_3 \rangle && \text{using (2) or (10)} \\ &\Rightarrow \langle \text{他 喜歡 } X_3, \text{He likes } X_3 \rangle && \text{using (3)} \\ &\Rightarrow \langle \text{他 喜歡 } X_5 \text{ 的 } X_6, \text{He likes the } X_6 \text{ that } X_5 \rangle && \text{using (18)} \\ &\Rightarrow \langle \text{他 喜歡 你 買 的 } X_6, \text{He likes the } X_6 \text{ that you bought} \rangle && \text{using (14)} \\ &\Rightarrow \langle \text{他 喜歡 你 買 的 書}, \text{He likes the book that you bought} \rangle && \text{using (6)} \end{aligned}$$

Figure 3.10: Derivation of a synchronous CFG by using rules in Figure 3.8 and Figure 3.9.

3.2.2 Model

To model our *Hierarchical Phrase-based Re-decoding Model*, we need to first provide definitions for the estimation of confidence scores.

Definition 1. For the i -th input sentence, one of the extracted translation rules j can be represented as $X \rightarrow \langle \gamma_j^i, \alpha_{h,j}^i \rangle$ and its confidence score for the system h can be represented as an indicator:

$$CS(\gamma_j^i, \alpha_{h,j}^i) = \begin{cases} 1 & \text{if } X \rightarrow \langle \gamma_j^i, \alpha_{h,j}^i \rangle \text{ occurs in } T_h^i \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

Definition 2. For the i -th input sentence and one of the extracted translation rules j , we can represent its overall confidence score as a weighted summarization over all MT systems' individual confidence score toward it:

$$\sum_{h=1}^{N_s} \lambda_h * CS(\gamma_j^i, \alpha_{h,j}^i) \quad (3.2)$$

Where N_s is the total number of MT systems, and λ_h denotes the weight of MT system h

Definition 3. For the i -th input sentence F^i , we can define the confidence score for its combination result \bar{E}^i as follows:

$$\log p(\bar{E}^i | F^i) = \sum_{j=1}^J \left(\sum_{h=1}^{N_s} \lambda_h * CS(\gamma_j^i, \alpha_{h,j}^i) \right) + \lambda_p * J + \lambda_l * \log(LM(\bar{E}^i)) + \lambda_w * length(\bar{E}^i) \quad (3.3)$$

J is the total number of phrases for the given sentence. λ_h is the weight of MT system h . λ_p is phrase penalty. λ_l is LM weight and λ_w is word penalty. All weights as well as word and

phrase penalties are trained discriminatively for Bleu score using Minimum Error Rate Training (MERT) procedure (Och 2004).

3.2.3 Decoding

Given an input source and the corresponding hierarchical phrases of MT systems, the decoder performs a search for the single most probable derivation via the CKY algorithm with a Viterbi approximation. The path of the search is our combination result. The single most probable derivation can be represented as

$$\bar{E}_{best}^i = \arg \max_{\bar{E}^i} \log p(\bar{E}^i | F^i) \quad (3.4)$$

3.2.4 Experiment

The experiments are conducted and reported on two datasets: One dataset includes Chinese-English system translations and references from DARPA GALE 2008 (GALE Chi-Eng Dataset). The other one includes Chinese-English system translations and references and from NIST 2008 (NIST Chi-Eng Dataset).

3.2.4.1 Setting

We described our experimental setting as follows:

GALE Chi-Eng Dataset: The GALE Chi-Eng Dataset consists of source sentences in Chinese, corresponding machine translations of 12 MT systems and four human reference translations in English. It also provides word alignments between source and translation sentences. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 422 sentences and

the test set also includes 422 sentences.

MT System name	Approach	BLEU	TER	MET
Sys nrc	phrase-based SMT	30.95	59.31	59.06
Sys rwth-pbt-aml	phrase-based SMT + source reordering	31.83	58.09	58.85
Sys rwth-pbt-jx	phrase-based SMT + Chinese word segmentation	31.78	62.04	57.51
Sys rwth-pbt-sh	phrase-based SMT + source reordering + rescoring	32.63	58.67	58.98
Sys sri-hpbt	hierarchical phrase-based SMT	32.00	58.97	58.84

Table 3.1: Techniques of top five MT of GALE Chi-Eng Dataset

From Table 3.1, we can see that “rwth-pbt-sh” performs the best in BLEU, “rwth-pbt-aml” performs the best in TER, and “nrc” performs the best in MET. Since we are tuning toward BLEU, we regard “rwth-pbt-sh” as the top MT system.

NIST Chi-Eng Dataset: The NIST Chi-Eng Dataset also consists of source sentences in Chinese, corresponding machine translations of multiple MT systems and four human reference translations in English, but word alignments between source and translation sentences are not included. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 524 sentences and the test set includes 788 sentences.

MT System name	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
Sys 15	30.06	55.16	54.49
Sys 20	28.15	57.97	52.36
Sys 22	29.94	56.10	54.19
Sys 31	29.52	56.29	54.31

Table 3.2: Techniques of top five MT of NIST Chi-Eng Dataset

From Table 3.2, we can see that “Sys 03” performs the best in BLEU, “Sys 15” performs the best in TER, and “Sys 15” performs the best in MET. Since we are tuning toward BLEU, we regard “Sys 03” as the top MT system.

We compare our *hierarchical phrase-based re-decoding model* with its baseline combination approach - *phrase-based re-decoding model* in this section. The estimations of confidence scores are the same as those described in section 3.2.2. The only difference is that the baseline uses phrases (continuous words) rather than hierarchical phrases.

3.2.4.2 Results

	BLEU	TER	MET
Sys rwth-pbt-sh	32.63	58.67	58.98
<i>phrase-based re-decoding model (baseline)</i>	31.02	60.62	57.32
<i>hierarchical phrase-based re-decoding model</i>	32.11	59.19	58.40

Table 3.3: Comparing the performance of *hierarchical phrase-based re-decoding model* with Top 1 MT system and *phrase-based re-decoding model (baseline)*.

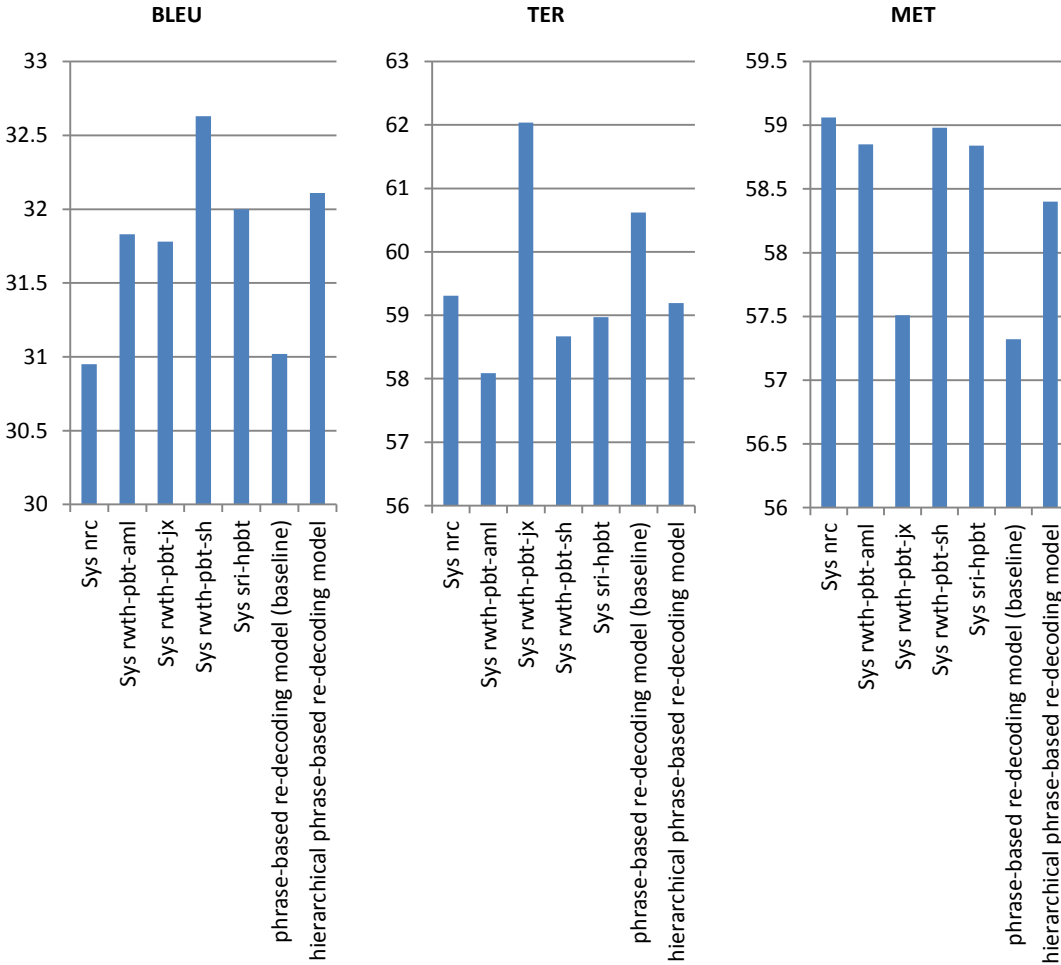


Figure 3.11: Comparing the performance of *hierarchical phrase-based re-decoding model* with all other systems

From Table 3.3, we see that the *hierarchical phrase-based re-decoding model* performs better than the *phrase-based re-decoding model*, showing that hierarchical phrases do bring some benefits by better modeling long-distance phrase reordering and the occurrences of discontinuous phrases. However, *hierarchical phrase-based re-decoding model* does not beat the best MT system.

3.3 Conclusions

In this chapter, we propose the *hierarchical phrase-based re-decoding model*, which outperforms one of the baseline combination systems – the *phrase-based re-decoding model*. It features the use of hierarchical phrases and the use of a synchronous context-free grammar dynamically learned from source sentence and target hypotheses to represent the translation information. Through re-decoding the source sentence with the hierarchical phrases, it is able to obtain the combination result with stronger abilities of word re-ordering and consensus among the multiple MT systems' translations compared with *phrase-based re-decoding model*.

For re-decoding framework, although our current model do not outperform the best MT system, there exists much potential to improve our approach because there are relatively more resources available to improve the performance in comparison with paraphrasing framework, such as bilingual corpora. So the future work for our re-decoding framework involves the integration of the existing translation probabilities trained from a bilingual corpus to the combination model.

Chapter 4

Phrase-level Combination: Combination by Paraphrasing

Phrase-level combination aims to retain coherence and consistency between the words in a phrase. In the previous chapter, we presented a new re-decoding model – the *hierarchical phrase-based re-decoding model* and demonstrate it performs better than the *phrase-based re-decoding model* but does not beat the best MT system.

In this chapter, we will present a different direction of phrase-level combination: instead of using re-decoding strategies, we propose to view combination as a paraphrasing process and use paraphrasing rules. Based on this idea, we present another phrase-level combination approach, called the *paraphrasing model*, described in Section 4.2. It extracts string-to-string paraphrases from the backbone and other hypotheses, and then uses these paraphrases with a reordering model to paraphrase the backbone. In order to further capture more complicated paraphrasing phenomena between the backbone and other target hypotheses, such as longer phrase reordering or the occurrences of discontinuous phrases, in Section 4.3, we also propose the use of hierarchical phrases — phrases that contain subphrases (Chiang 2007) — for paraphrasing-based

combination. We learn hierarchical paraphrases from monolingual word alignments between a selected backbone hypothesis and other hypotheses. These hierarchical paraphrases can model more complicated paraphrasing phenomena, and thus enable more utilization of consensus among MT engines than non-hierarchical paraphrases do. We call this technique the *hierarchical paraphrasing model*.

4.1 Related Work: Lattice Decoding Model

In recent years, some phrase-level combination techniques have been presented. They rely on a *lattice decoding model* to carry out the combination (Feng et al 2009; Du and Way 2010). In a lattice, each edge is associated with a phrase (a single word or a sequence of words) rather than a single word. The construction of the lattice is based on the extraction of phrase pairs from word alignments between a selected best MT system hypothesis (the backbone) and the other translation hypotheses. The combination is carried out through decoding over the phrase lattice to search for the best path.

Feng et al (2009) designed heuristic rules to extract paraphrases from word alignments between the backbone and the set of hypotheses. The paraphrases are allowed to be discontinuous but are required to be “minimum” alignment units unless they are generated by adding null words. The lattice was then constructed by adding aligned sentence pairs incrementally. In (Du and Way 2010), a Translation Error Rate Plus (TERp) tool was employed to carry out the word alignment between the backbone and other hypotheses; a lattice is built by extracting paraphrases based on certain alignment types that TERp indicated, i.e, “stem match”, “synonym match” and paraphrases.

For the *lattice decoding model*, the word order of the backbone determines the word order of consensus outputs and thus, they are able to use existing word ordering of the backbone; however, lattice decoding models lack the ability to reorder words of the backbone.

4.2 Paraphrasing Model

In contrast to the above state-of-the-art lattice decoding techniques, we propose a novel perspective for combination: the combination process is regarded as a paraphrasing process. It extracts string-to-string paraphrases from the backbone (the selected hypothesis) and other hypotheses, and then uses these paraphrases to paraphrase the backbone. We call this technique the *paraphrasing model*. The process can be also interpreted as a post-editing process over the backbone, which system diagram is shown in Figure 4.1.

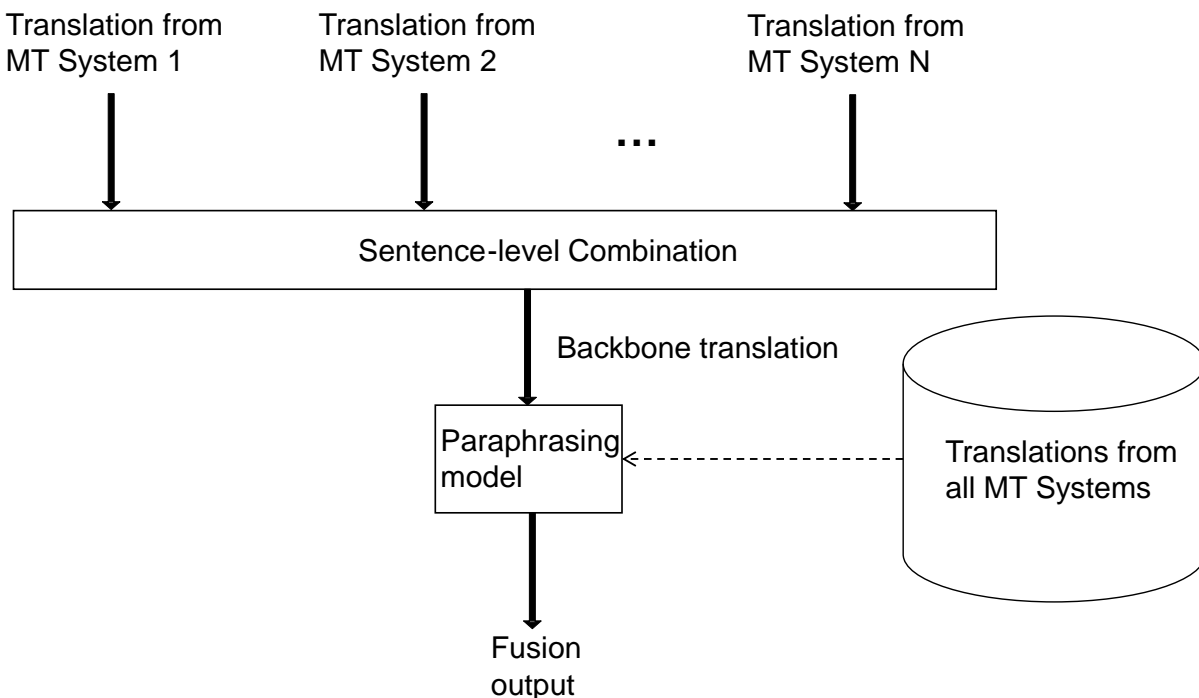


Figure 4.1: The system diagram of Paraphrasing Model

The paraphrasing perspective motivates the application of various existing phrase-based MT techniques in the combination framework. For example, bilingual phrase extraction rules (Koehn et al, 2003), which are widely used in MT, can directly map to a target-to-target version for our paraphrase extraction. The simple but efficient rules avoid the complexity of (Feng et al 2009)’s heuristic alignment-unit rules. Moreover, to extract paraphrases that are more than one word, (Feng et al 2009) and (Du and Way 2010)’s rules rely only on crossing or many-to-many word alignments that their monolingual word aligners provided, while our rules are capable of utilizing not only crossing and many-to-many word alignments but also one-to-one monolingual word alignments to form multi-word paraphrases, and this enables us to extract many more paraphrases than (Feng et al 2009) and (Du and Way 2010). For the same reason, even though our implementation uses TERp tool as the word aligner, the *paraphrasing model* actually can be applied to any kind of monolingual word aligner, including a pure one-to-one word aligner, such as Translation Error Rate (TER). Other benefits of the *paraphrasing model* include the fact that the phrase-table based lattice avoids the complexity of lattice construction in (Feng et al 2009), and decoding over the backbone enables us to integrate a reordering model into our combination model directly.

The *paraphrasing model* involves the following steps:

1. Collect the hypotheses from multiple MT systems.
2. Select the backbone sentence hypothesis. The common strategy is through Minimum Bayes Risk (MBR) decoding (Sim et al., 2007; Rosti et al., 2007a; Feng et al 2009) or system-weighted MBR (Du and Way 2010). These approaches basically only rely on the agreement of system

hypotheses. In order to utilize other information, such as a LM, we view the backbone selection as a sentence-based MT combination framework and design the following log-linear model:

$$\log p(E_i) = \sum_{s=1}^{N_s} (\lambda_s * \log(1 - TER(E_i, E_s))) + \lambda^l * \log(LM(E_i)) + \lambda^w * Length(E_i) \quad (4.1)$$

Where E is system hypothesis, N_s is system number, λ_s is system weight, λ^l is LM weight and λ^w is word penalty.

3. Get the word alignments between the backbone and all system hypotheses. The *paraphrasing model* actually can be applied to any kind of monolingual word aligner. In our implementation, we adopt TERp, one of the state-of-the-art alignment tools, to serve this purpose, described in section 4.2.1.1.

5. Given the word alignments between the backbone and all system hypotheses, we extract paraphrases as phrase table entries, described in section 4.2.1.2.

6. Assign each entry in the phrase table a paraphrase confidence score, described in section 4.2.2.

Sys2: I prefer apples
 Sys1: I feel like fruit
 Sys3: I am fond of apples

Figure 4.2(a): Example of word alignments of hypotheses. Assume Sys1 as the baseline.

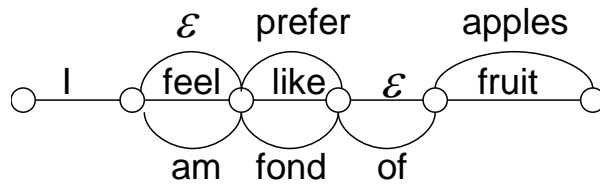


Figure 4.2(b): Confusion Network based on word alignments in Figure 4.2(a).

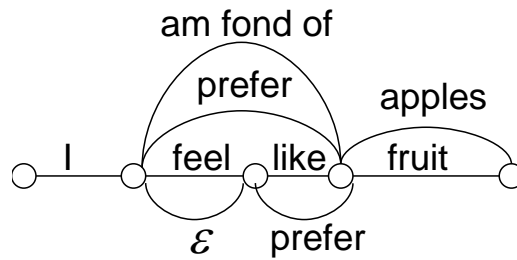


Figure 4.2(c): Lattice based on word alignments in Figure 4.2(a).

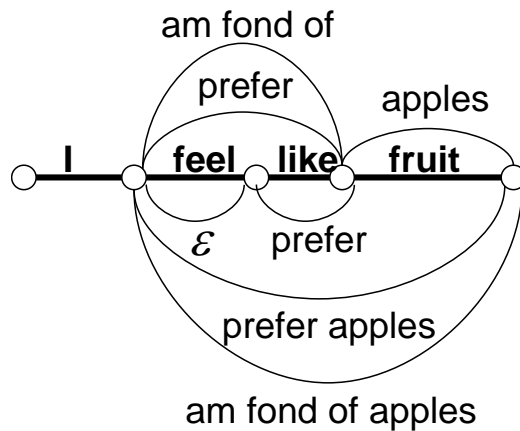


Figure 4.2(d): Search Space of the *paraphrasing model* based on word alignments in Figure 4.2(a).

Bold lines and words indicate the baseline.

The example in Figure 4.2 provides a comparison between the *paraphrasing model* and other combination approaches from the view of search space. Based on the word alignments of hypotheses from MT systems, shown in Figure 4.2(a), we construct a confusion network in Figure 4.2(b), a lattice in Figure 4.2(c), and the search space of our *paraphrasing model* in Figure 4.2(d).

From Figure 4.2(b), we see that although “am fond of ” and “feel like” have the same meaning, the confusion network-based approaches face the risk of producing degenerate translations, such as “am like of” and “feel fond of”. In Figure 4.2(c), we see that the phrases “am fond of” and “feel like” are not allowed to be mixed, but it does not consider the paraphrases - “prefer apples” and “feel like apples” and the paraphrases - “am fond of apples” and “feel like apples”. And because *lattice decoding* searches the path from left to right, the word order of the backbone completely determines the word order of consensus outputs. Thus, the *lattice decoding* search lacks the ability to reorder the words of the backbone. On the other hand, in Figure 4.2(d), we see that the *paraphrasing model* overcomes the problems of *confusion network decoding* and *lattice decoding*. It considers the paraphrases - “prefer apples” and “feel like apples” and the paraphrases - “am fond of apples” and “feel like apples”. Since the decoding object is no longer the lattice, but the backbone, it has the ability to reorder words of the backbone.

4.2.1 Paraphrase Extraction

The process of paraphrase extraction is divided into two steps. We first use a word aligner to get word alignments of hypotheses and then extract paraphrases based on these word alignments.

4.2.1.1 Monolingual Word Alignment

Our paraphrases are deduced from monolingual word alignment. Any monolingual word aligner can serve the purpose. In our implementation, we adopt TERp as our alignment tool. We briefly review it and use an abstract example to illustrate its alignment output format and how we slightly adjust the format to meet our needs.

TERp (Snover et al. 2009) is an extension of TER (Snover et al. 2006). Both TERp and TER are automatic evaluation metrics for MT, based on measuring the ratio of the number of edit operations between the reference sentence and the MT system hypothesis. TERp uses all the edit operations of TER—Matches, Insertions, Deletions, Substitutions and Shifts—as well as three new edit operations: Stem Matches, Synonym Matches and Paraphrases. TERp identifies the Stem Matches and Synonym Matches using the Porter stemming algorithm (Porter, 1980) and WordNet (Fellbaum, 1998) respectively. Sequences of words in the reference are considered to be paraphrases of a sequence of words in the hypothesis if that phrase pair occurs in the TERp’s own paraphrase database.

One valuable characteristic of TERp is that it can produce very high-quality alignments between two given input sentences and identify the alignment types including M (Exact Match), I (Insertion), D (Deletion), S (Substitution), T (Stem Match), Y (Synonym Match) and P (Paraphrase). While P is a phrase alignment, all other types are word alignments. An real alignment example using TERp is shown as Figure 4.3.

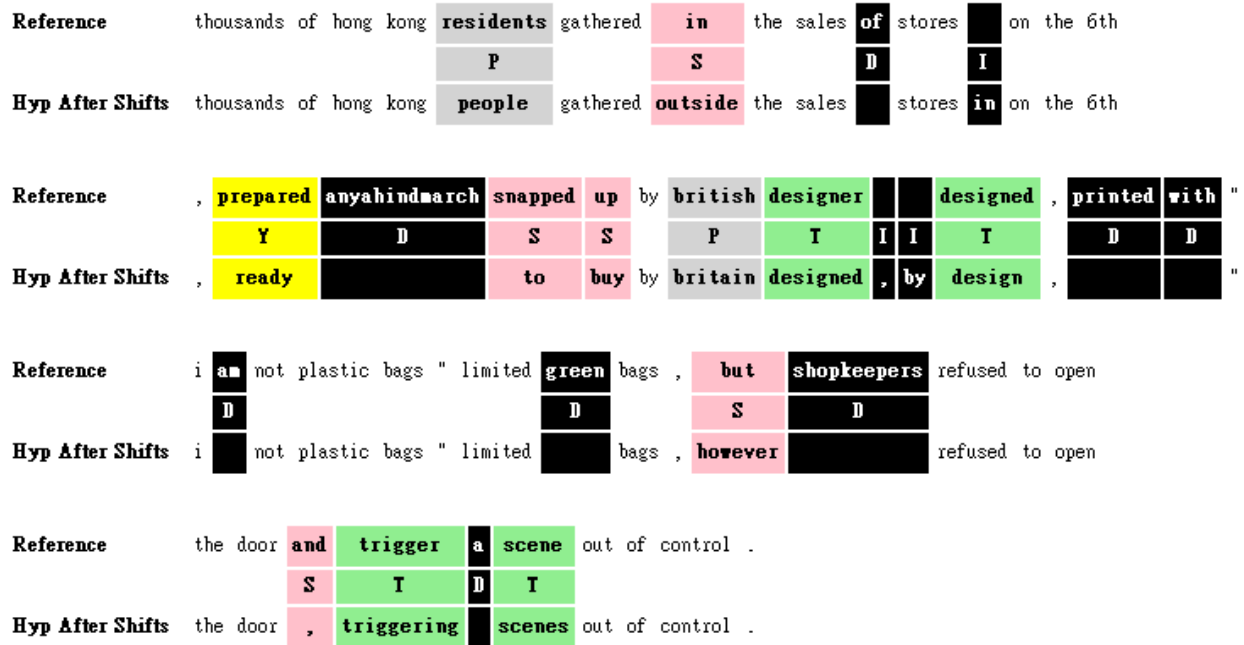


Figure 4.3: An real alignment example using TERp. P (Paraphrase) is shown in gray; S (Substitution) is shown in pink; I (Insertion) and D (Deletion) are shown in black; Y (Synonym Match) is shown in yellow; T (Stem Match) is shown in green; M (Exact Match) is shown in no color.

To better illustrate the tool, we use an abstract instance. Assume we have a backbone E_b and a system hypothesis E_h as follows:

$$E_b : w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6 \quad w_7 \quad w_8 \quad w_9 \quad w_{10} \quad w_{11}$$

$$E_h : \bar{w}_1 \quad \bar{w}_2 \quad \bar{w}_3 \quad \bar{w}_4 \quad \bar{w}_5 \quad \bar{w}_6 \quad \bar{w}_7 \quad \bar{w}_8 \quad \bar{w}_9 \quad \bar{w}_{10}$$

Figure 4.4: A backbone E_b and a system hypothesis E_h

where each w_i means a word w in position i in the sentence.

Given the sentence pair as input for the TERp tool, the alignment between E_b and E_h could be produced as follows:

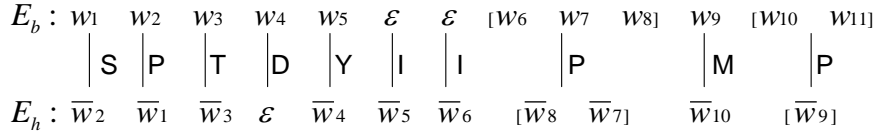


Figure 4.5: The alignment between E_b and reordered E_h

Note that in the alignment produced by TERP in Fig. 4.5, E_b 's word order remains the same but E_h 's word order is changed to fit the most reasonable alignment. To extract paraphrases using our extraction rules, we re-order it back to the original word order and keep the alignment links and types. In order to generate a pure word alignment, for each P, we link every word of E_b to every word of E_h . The adjusted format is as follows:

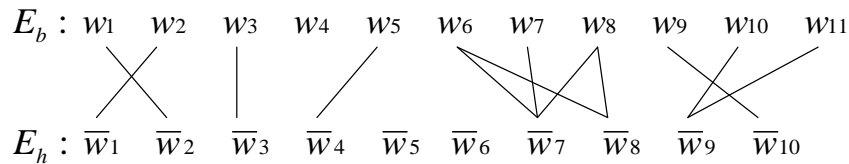


Figure 4.6: The alignment between E_b and E_h with the original word order

4.2.1.2 Algorithm for Paraphrase Extraction

Before introducing our paraphrase extraction strategy, it is worth discussing the motivation: if we compare the phrase-level combination model with a phrase-based translation model, we see their motivations are quite similar. In translation, it is very common for several words in a foreign language to translate as a whole to several words in the target language. Similarly, in combining a pair of different translation hypotheses, sometimes several words can be substituted as a whole for several other words. For example, “is sick of” and “is disgusted with” basically carry the same meaning and have similar usages. Using the word as the unit to perform combination would run the risk of producing incorrect translations, such as “is sick with” or “is disgusted of”. Since translation and combination share a similar motivation for using phrases, it is natural for us

to apply a similar phrase extraction strategy in our combination framework.

We map the standard bilingual phrase extraction rules (Koehn et al, 2003) to the following target-to-target version for our paraphrase extraction: we extract all phrases that are word-continuous and consistent with the monolingual word alignment. This means that words in a legal paraphrase are not aligned to words outside of the paraphrase, and should include at least one pair of words aligned with each other. The definition of consistency can be formally stated as follows: assume e is a phrase of a backbone and e_h is a phrase of a MT system hypothesis. A pair of phrases (e, e_h) is consistent with the monolingual word alignment matrix A if

$$\begin{aligned} & \forall w_i \in e : (w_i, x) \in A \Rightarrow x \in e_h \\ \text{and} & \quad \forall w_j \in e_h : (y, w_j) \in A \Rightarrow y \in e \\ \text{and} & \quad w_i \in e, w_j \in e_h : (w_i, w_j) \in A \end{aligned}$$

where w_i is a word of e , w_j is a word of e_h .

For a paraphrase (e, e_h) , we make word position information attach to e , while it is not necessary to do so with e_h . This results in pairs, such as (is_20 disgusted_21 with_22, is sick of), where 20-22 are the word positions in the backbone.

We use the same Chinese-to-English example of Figure 3.2 to illustrate the paraphrasing process. We assume Eh1 - “He likes you buy the book” is the selected backbone sentence hypothesis, and Eh2 – “He likes the books that you bought” is another hypothesis. Figure 4.7 shows the word alignments between the backbone and another hypothesis. Figure 4.8 shows the extracted paraphrases from the translation by MT system h1, and Figure 4.9 shows the extracted paraphrases from the translation by MT system h2.

他 喜歡 你 買 的 書
 He likes you bought s' book
 (He likes the book that you bought)

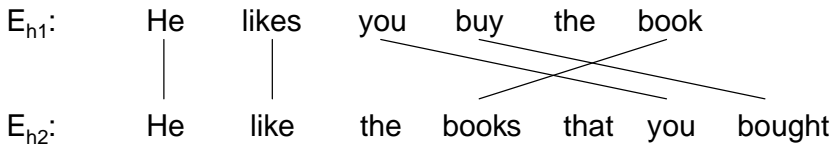


Fig 4.7: A backbone sentence (the translation Eh1), the translation Eh2 and the word alignment between the two.

- | | |
|---|---|
| < He , He > | < you , you > |
| < He likes , He likes > | < you buy , you buy > |
| < He likes you , He likes you > | < you buy the , you buy the > |
| < He likes you buy , He likes you buy > | < you buy the book , you buy the book > |
| < He likes you buy the , He likes you buy the > | < buy , buy > |
| < He likes you buy the book , He likes you buy the book > | < buy the , buy the > |
| < likes , likes > | < buy the book , buy the book > |
| < likes you , likes you > | < the , the > |
| < likes you buy , likes you buy > | < the book , the book > |
| < likes you buy the , likes you buy the > | < book , book > |
| < likes you buy the book , likes you buy the book > | |

Figure 4.8: The extracted phrases from the translation by MT system h1.

- < He , He >
- < He likes , He like >
- < He likes you buy the book , He like the books that you bought >
- < likes , like >
- < likes you buy the book , like the books that you bought >
- < you buy , you bought >
- < you buy , that you bought >
- < you buy the , you bought >
- < you buy the , that you bought >
- < you buy the book , books that you bought >
- < you buy the book , the books that you bought >

Figure 4.9: The extracted phrases from the translation by MT system h2.

Given the extracted phrases of Figure 4.8 and Figure 4.9, the *paraphrasing model* has the chance of getting the correct translation - “He likes the book that you bought” by using the rule of “<He likes, He likes>” from MT system h1, the rule of “<you buy, that you bought >” from MT system h2 and the rule of “<the book, the book >” from MT system h1, and by reordering the order of “that you bought” and “the book” to the order of “the book” and “that you bought”.

4.2.2 Model

We use the basic translation model in MT as inspiration for our combination model.

Definition 1. For the backbone of the i-th input sentence and translation of MT system h, one of the extracted paraphrasing rules j can be represented as $\langle e_j^i, \bar{e}_{h,j}^{i,h} \rangle$ and its confidence score for the system h can be represented as an indicator:

$$CS(e_j^i, \bar{e}_{h,j}^{i,h}) = \begin{cases} 1 & \text{if } e_j^i \text{ and } \bar{e}_{h,j}^{i,h} \text{ are paraphrases} \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

Definition 2. For the backbone of i-th input sentence and one of the extracted paraphrasing rules j, we can represent its overall confidence score as a weighted summarization over all MT systems’ individual confidence score toward it:

$$\sum_{h=1}^{N_s} \lambda_h * CS(e_j^i, \bar{e}_{h,j}^{i,h}) \quad (4.3)$$

Where N is the total number of MT systems, and λ_h denotes the weight of MT system h

Definition 3. For the backbone (E^i) of the i -th input sentence, we can define the confidence score for its combination result \bar{E}^i as follows:

$$\log p(\bar{E}^i | E^i) = \sum_{j=1}^J \left(\sum_{h=1}^{N_s} \lambda_h * CS(e_j^i, \bar{e}_{h,j}^i) \right) + \sum_{j=1}^J (\lambda^d * d(start_j, end_{j-1}))$$

$$\lambda_p * J + \lambda_l * \log(LM(\bar{E}^i)) + \lambda_w * length(\bar{E}^i) \quad (4.4)$$

J is the total number of phrases for the given sentence. λ_h is the weight of MT system h . λ^p is phrase penalty, which controls the preference of phrase length. λ_w is word penalty, which controls the preference of hypothesis length. d is a reordering model based on distortion cost, weighted by λ^d . LM is a general language model, weighted by λ^l . In this combination model, all weights, as well as word and phrase penalty, can be trained discriminatively for Bleu score using Minimum Error Rate Training (MERT) procedure (Och 2004).

4.2.3 Decoding

Given the backbone of an input source and the corresponding paraphrasing rules, the decoder performs a search for the single most probable path via a Viterbi approximation. The path of the search is our combination result. It can be represented as

$$\bar{E}_{best}^i = \arg \max_{\bar{E}^i} \log p(\bar{E}^i | E^i)$$

Here we mimic the combination processing using our *paraphrasing model* to combine the two hypotheses in Figure 4.7. By using the extracted phrases of Figure 4.8 and Figure 4.9. The model has the chance of getting the correct translation - “He likes the book that you bought” by using the rule “<He likes, He likes>” from MT system h1, the rule “<you buy, that you bought >” from MT system h2 and the rule “<the book, the book>” from MT system h1. Please note that because

of the reordering model, the *paraphrasing model* has the ability to put “the books” and “that you bought” in a right order. On the other hand, if we use *the lattice decoding model* of (Feng et al 2009) and (Du and Way 2010) to combine the two hypotheses, and assume the extracted phrases of Figure 4.7 and Figure 4.8 are given, the best translation we can get is “He likes that you bought the book” by using the same three rules. The only mistake of the translation is that “that you bought” and “the book” should be switched in order, because the model lacks of the ability of word reordering.

One implementation detail for the *paraphrasing model* is based on the fact that the words in the backbone are not necessarily unique within the entire sentence, so before decoding, they need to be indexed using word positions. Any standard translation decoder can be used to decode the format¹. Take a toy example to illustrate the decoding process as follows. Start with an indexed backbone:

... He_19 is_20 disgusted_21 with_22 that_23 ...

Assume there are only four entries in our phrase table:

(He_19, He)

(is_20 disgusted_21 with_22, is disgusted with)

(is_20 disgusted_21 with_22, is sick of)

(that_23, that)

Then one of the following hypotheses would be generated by the decoding:

... He is disgusted with that ...

... He is sick of that ...

¹ In our implementation, we use MOSES (<http://www.statmt.org/moses/>)

4.2.4 Experiments

Our experiments are conducted and reported on three datasets: The first dataset includes Chinese-English system translations and reference translations from DARPA GALE 2008 (GALE Chi-Eng Dataset). The second dataset includes Chinese-English system translations and reference translations and from NIST 2008 (NIST Chi-Eng Dataset). And the third dataset includes Arabic-English system translations and reference translations and from NIST 2008 (NIST Ara-Eng Dataset).

4.2.4.1 Setting

We use the GALE Chi-Eng Dataset and the NIST Chi-Eng Dataset as in Section 3.2.4.1. For the reader’s convenience, we briefly describe the two datasets here again first, followed by the introduction of the NIST Ara-Eng Dataset.

GALE Chi-Eng Dataset: The GALE Chi-Eng Dataset consists of source sentences in Chinese, corresponding machine translations of 12 MT systems and four human reference translations in English. It also provides word alignments between source and translation sentences. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 422 sentences and the test set also includes 422 sentences. Among the five systems, “rwth-pbt-sh” performs the best in BLEU, and since we are tuning toward BLEU, we regard “rwth-pbt-sh” as the top MT system.

NIST Chi-Eng Dataset: The NIST Chi-Eng Dataset also consists of source sentences in Chinese, corresponding machine translations of multiple MT systems and four human reference translations in English, but word alignments between source and translation sentences are not

included. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 524 sentences and the test set includes 788 sentences. Among the five systems, “Sys 03” performs the best in BLEU, and since we are tuning toward BLEU, we regard “Sys 03” as the top MT system.

NIST Ara-Eng Dataset: The previous datasets are Chinese-English datasets. We evaluated our models on the test set of these two datasets for every combination approach. Although we did not inspect the errors of the test set during development of a new approach, we also wanted to run our system after all approaches were finalized on a brand new dataset. We use a dataset of a different language pair as a blind test to further demonstrate our models’ robustness and consistency. The NIST Ara-Eng Dataset plays this role. It consists of source sentences in Arabic, corresponding machine translations of multiple MT systems and four human reference translations in English, but word alignments between source and translation sentences are not included. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 592 sentences and the test set includes 717 sentences.

MT System name	BLEU	TER	MET
Sys 03	45.81	48.88	69.34
Sys 07	44.67	46.70	68.00
Sys 15	45.71	46.20	70.24
Sys 26	45.83	45.35	69.42
Sys 31	48.40	45.55	70.67

Table 4.1: Techniques of top five MT of NIST Ara-Eng Dataset

From Table 4.1, we can see that “Sys 31” performs the best in BLEU, “Sys 26” performs the best in TER, and “Sys 31” performs the best in MET. Since we are tuning toward BLEU, we regard “Sys 31” as the top MT system.

4.2.4.2 Results

	BLEU	TER	MET
Sys rwth-pbt-sh	32.63	58.67	58.98
<i>phrase-based re-decoding model (baseline)</i>	31.02	60.62	57.32
<i>hierarchical phrase-based re-decoding model</i>	32.11	59.19	58.40
<i>Confusion Network (baseline)</i>	33.04	57.08	59.44
<i>paraphrasing model</i>	33.16	56.63	59.46

Table 4.2: GALE Chi-Eng Dataset : The *paraphrasing model* in comparison with baseline and previous results

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>paraphrasing model</i>	32.65	55.11	56.17

Table 4.3: NIST Chi-Eng Dataset : The *paraphrasing model* in comparison with baseline.

	BLEU	TER	MET
Sys 31	48.40	45.55	70.67
<i>Confusion Network (baseline)</i>	48.56	43.81	70.67
<i>paraphrasing model</i>	49.33	45.08	70.87

Table 4.4: NIST Ara-Eng Dataset : The *paraphrasing model* in comparison with baseline.

From Table 4.2 and 4.3, we can make the following observations: 1. For the three datasets, the *paraphrasing model* performs better than the top MT system. 2. For the three datasets, the *paraphrasing model* performs better than *confusion network decoding*, which supports our basic claim about the advantage of using phrases in combination. Especially for NIST Chi-Eng Dataset, the *paraphrasing model* enlarges the leading gap in comparison with the *confusion network decoding model*. 3. From Table 4.2, we find the *paraphrasing model* performs better than both re-decoding models. The reason could be that for the re-decoding models, we decode the source sentence, and more word reordering needs to be modeled because the input and output are in different languages. On the other hand, for the *paraphrasing model*, the backbone sentence is decoded, and less word reordering needs to be modeled because the input and output are in the same languages. In other words, the backbone has similar word reordering with the eventual combination results, lowering the chance of causing errors in word reordering.

To provide another objective evaluation, we also evaluate our *paraphrasing model* on NIST Ara-Eng Dataset as a blind test. The results are shown in Table 4.4. We see that the *paraphrasing model* still achieves the better performance in BLEU in comparison with the *confusion network decoding model*, which demonstrates the *paraphrasing model's* robustness and consistency. It shows the results are consistent across test sets and across two languages.

4.2.4.3 Analysis of Phrase Length

For our *paraphrasing model*, how long do phrases have to be to achieve high performance? Figure 4.10 displays results from experiments with different maximum phrase lengths for NIST Chi-Eng Dataset. We find that limiting the length to a maximum of five words per phrase achieves top performance in BLEU, and that limiting the length to a maximum of three words per phrase achieves top performance in MET, and that limiting the length to a maximum of seven

words per phrase achieves top performance in TER. Because we are tuning toward BLEU, we regard a maximum of five words per phrase is the best setting for the *paraphrasing model*.

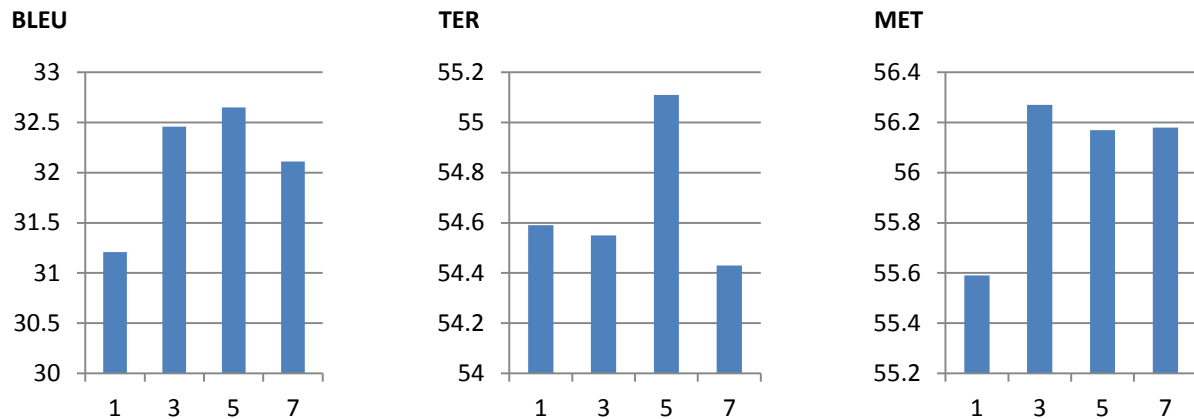


Figure 4.10: Different limits for maximum phrase length for NIST Chi-Eng Dataset.

4.2.4.4 Analysis of Syntactic Paraphrase Extraction

In section 4.2.1.2, we introduced our paraphrase extraction method: extract all phrases that are word-continuous and consistent with the monolingual word alignment, which does not consider any syntactic information or restriction. To understand the effect of syntactic paraphrases, in this section, we use the following three different extraction methods for our *paraphrasing model*.

Extraction Method A: a pair of phrases (e , e_h) is consistent with the monolingual word alignment, and only e is a constituent.

Extraction Method B: a pair of phrases (e , e_h) is consistent with the monolingual word alignment, and e and e_h are both constituents.

Extraction Method C: a pair of phrases (e , e_h) is consistent with the monolingual word alignment, and e and e_h are both constituents with the same constituent types, such as NP, VP, PP...etc.

In the three extraction methods, the constituents and their types are determined by the Stanford Parser. The combination results using these methods are shown in Table 4.5.

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>paraphrasing model</i>	32.65	55.11	56.17
<i>paraphrasing model with Extraction Method A</i>	32.11	55.07	56.18
<i>paraphrasing model with Extraction Method B</i>	31.73	54.78	56.09
<i>paraphrasing model with Extraction Method C</i>	31.66	55.30	55.78

Table 4.5: Comparing the performance of *paraphrasing model* using different extraction methods for NIST Chi-Eng Dataset.

Table 4.5 shows that syntactic paraphrases give no improvement in comparison with the basic extraction rules in section 4.2.1.2. The results might be explained by the following reason: restricting paraphrases to be syntactic paraphrases enforces the *paraphrasing model* to retain the same or similar overall syntactic structure of the backbone hypothesis. But because of these restrictions, only fewer paraphrases are extracted and many reasonable paraphrases are missing, resulting in the consequence that the backbone has a smaller chance to be paraphrased.

4.2.4.5 Analysis of the Addition of Syntactic Features

In MT, to investigate the impact of syntactic information, Koehn et. al. (2003) weighted syntactic phrases in the phrase table used in their MT experiments, and found that the consideration of syntactic phrases does not bring benefits. We adopt a similar strategy; we add the following different features individually in (4.4).

Feature A

$$\text{syn}(e_j^i, \bar{e}_{h,j}^i) = \begin{cases} 1 & \text{if } (e_j^i, \bar{e}_{h,j}^i) \text{ is consistent with the monolingual word alignment, and only } e_j^i \text{ is constituent.} \\ 0 & \text{otherwise} \end{cases}$$

Feature B

$$\text{syn}(e_j^i, \bar{e}_{h,j}^i) = \begin{cases} 1 & \text{if } (e_j^i, \bar{e}_{h,j}^i) \text{ is consistent with the monolingual word alignment, and } e_j^i \text{ and } \bar{e}_{h,j}^i \text{ are} \\ & \text{both constituents.} \\ 0 & \text{otherwise} \end{cases}$$

Feature C

$$\text{syn}(e_j^i, \bar{e}_{h,j}^i) = \begin{cases} 1 & \text{if } (e_j^i, \bar{e}_{h,j}^i) \text{ is consistent with the monolingual word alignment, and } e_j^i \text{ and } \bar{e}_{h,j}^i \text{ are} \\ & \text{both constituents with the same constituent types, such as NP, VP, PP...etc.} \\ 0 & \text{otherwise} \end{cases}$$

Each feature is attached with a weight, obtained from MERT process. In the previous section, Method A, B and C are hard constraints about syntactic paraphrases. In this section, Feature A, B and C can be regarded as soft constraints about syntactic paraphrases. In the three features, the constituents and their types are determined by Stanford Parser. The combination results using these methods are shown in Table 4.6.

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>paraphrasing model</i>	32.65	55.11	56.17
<i>paraphrasing model with Feature A</i>	32.54	54.65	56.24
<i>paraphrasing model with Feature B</i>	32.07	55.29	55.87
<i>paraphrasing model with Feature C</i>	31.91	54.82	56.02

Table 4.6: Comparing the performance of *paraphrasing model* using different features about syntactic paraphrases for NIST Chi-Eng Dataset.

From Table 4.6, we found that the features for syntactic paraphrases still gave no improvement in comparison with the basic extraction rules in section 4.2.1.2.

4.2.4.6 Analysis of the Selections of MT systems

In Section 4.2.4.2, we manually select the top five MT systems for our combination experiment. Here we investigate whether this selection based on MT systems' performances is reasonable and able to yield the best performance. We compare the performances of top three, top five, top seven, top nine MT systems, and another selection of five MT systems – 6th-10th MT systems.

		BLEU	TER	MET
The Best MT system	Sys 03	30.16	55.45	54.43
<i>Word-level combination (baseline)</i>	<i>Confusion Network</i>	31.21	54.59	55.59
<i>Phrase-level combination</i>	<i>paraphrasing model (top 3 sys)</i>	31.34	55.39	55.45
	<i>paraphrasing model (top 5 sys)</i>	32.65	55.11	56.17
	<i>paraphrasing model (top 7 sys)</i>	32.52	54.95	56.20
	<i>paraphrasing model (top 9 sys)</i>	32.48	55.02	56.17
	<i>paraphrasing model (6th-10th sys)</i>	28.44	58.53	53.11

Table 4.7: The combination performances using top 5 MT systems v.s. other choices of input MT systems on NIST Chi-Eng Dataset.

Table 4.7 shows that, the performance of top five systems provides the best performance in BLEU even compared with top seven and top nine MT systems. In other words, adding more MT systems does not always bring benefits when the added MT systems are relatively poorer. From Table 4.7, we also see that the performance of combination based on 6th-10th MT systems drops significantly, which indicates that the performance of combination strongly correlates with the individual quality of each MT system. To further support this interpretation, we compare the performances of using the selection of top three MT systems with other selections of three MT systems. The results are shown in Table 4.8.

		BLEU	TER	MET
The Best MT system	Sys 03	30.16	55.45	54.43
<i>Word-level combination (baseline)</i>	<i>Confusion Network</i>	31.21	54.59	55.59
<i>Phrase-level combination</i>	<i>paraphrasing model (top 3 sys)</i>	31.34	55.39	55.45
	<i>paraphrasing model (4th-6th sys)</i>	27.92	56.85	52.38
	<i>paraphrasing model (7th-9th sys)</i>	26.82	59.27	51.76

Table 4.8: The combination performances using top 3 MT systems and other choices of three MT systems on NIST Chi-Eng Dataset.

From Table 4.8, we see that top 3 MT systems performs the best and the lowest quality 3 MT systems performs the worst, which indicates again that the performance of combination strongly correlates with the individual quality of each MT system.

From these analyses, we can conclude that for MT combination, the selection of top N MT systems is a reasonable strategy, but larger N does not always bring benefits when N exceeds 5.

4.3 Hierarchical Paraphrasing Model

In the last section, we introduced the *paraphrasing model*, relying on string-to-string paraphrases to paraphrasing the backbone. However, these string-to-string paraphrases are not able to capture more complicated paraphrasing phenomena between the backbone and other target hypotheses, such as longer phrase reordering or the occurrences of discontinuous phrases.

In this section, we propose the use of hierarchical phrases—phrases that contain subphrases

(Chiang 2007) -- for machine translation system combination. We present a hierarchical phrase-level combination for paraphrasing by using a synchronous context-free grammar dynamically learned from bi-text without any syntactic annotations. We learn hierarchical paraphrases from monolingual word alignments between a selected backbone hypothesis and other hypotheses. These hierarchical paraphrases can model more complicated paraphrasing phenomena, and thus enable more utilization of consensus among MT engines than non-hierarchical paraphrases do. We call this technique the *hierarchical paraphrasing model*. Figure 4.11 shows the system diagram.

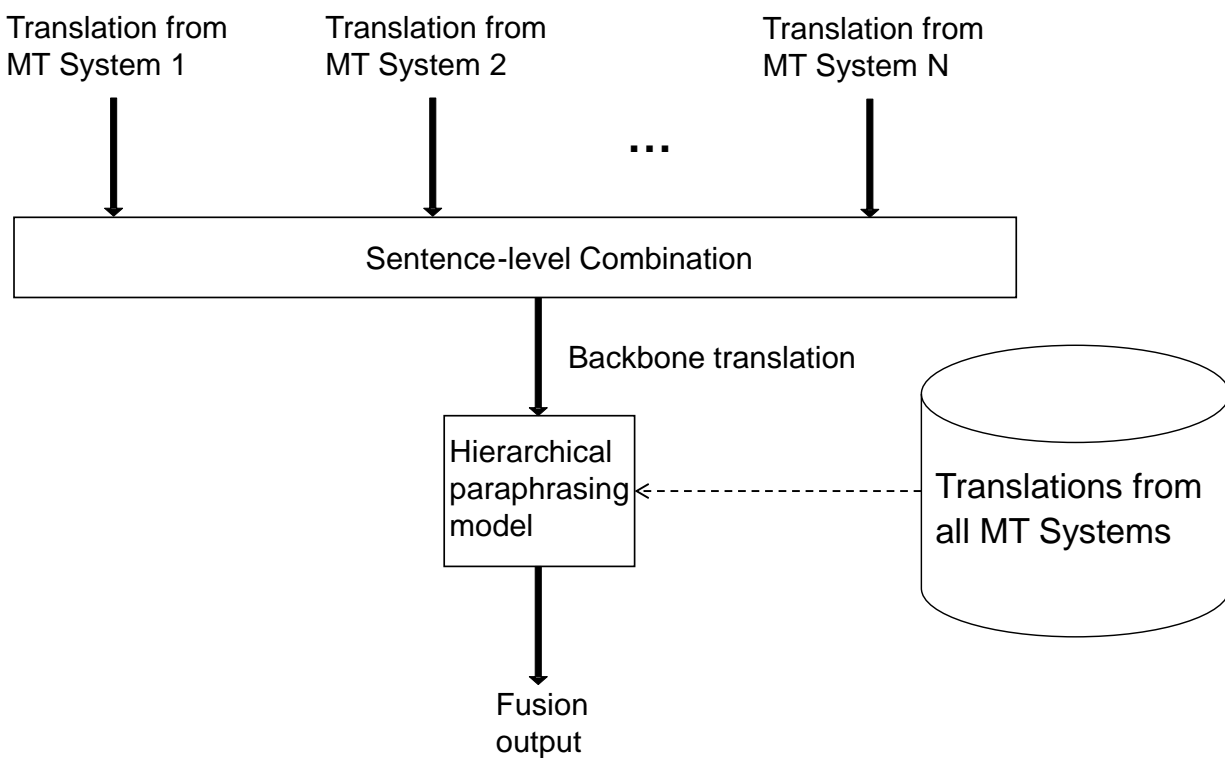


Figure 4.11: The system diagram of Hierarchical Paraphrasing Model

The combination process involves the following steps:

1. Collect the translation hypotheses from multiple MT systems.

2. Select the backbone translation hypothesis E from multiple translations for each input sentence.

We follow the common strategy of Minimum Bayes Risk (MBR) decoding (Sim et al., 2007; Rosti et al., 2007a; Feng et al 2009; Du and Way 2010) and use TER-based consensus to select the backbone. The selection method is the same as what we described in the step2 of *paraphrasing model*. For the reader's convenience, we describe it here again:

$$\log p(E_i) = \sum_{s=1}^{N_s} (\lambda_s * \log(1 - TER(E_i, E_s))) + \lambda^l * \log(LM(E_i)) + \lambda^w * Length(E_i) \quad (4.5)$$

Where E is system hypothesis, N_s is system number, λ_s is system weight, λ^l is LM weight and λ^w is word penalty.

3. Get monolingual word alignments between the backbone and all system hypotheses. We adopt TERp, one of the state-of-the-art alignment tools, to serve this purpose.

4. Extract phrases from the given monolingual word alignments. We extract all phrases that are word-continuous and consistent with the monolingual word alignment for each MT system. The extraction algorithm is the same as what we showed in section 4.2.1.

5. Extract hierarchical phrases from the given extracted phrases in step 4. The formal extraction algorithm is provided in section 4.3.1.

6. Assign each hierarchical phrase a confidence estimation, as described in section 4.3.2.
7. Re-decode the backbone using the above hierarchical phrases with confidence estimation as described in section 4.3.3.

4.3.1 Hierarchical Paraphrase Extraction

Our hierarchical phrase-level paraphrasing rules are designed as a synchronous-CFG, extracted from the backbone and the given translations of multiple MT systems. For an i -th sentence, we use E^i and e^i to represent the backbone and one of its phrases, respectively. E_h^i represents the translation of MT system h , and e_h^i is one phrase of E_h^i . We use Q_h^i to denote the set of the paraphrasing rules for sentence i and MT system h , and show how to collect Q_h^i as follows:

If $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment,

then $X \rightarrow \langle e^i, e_h^i \rangle$ is added to Q_h^i .

If $X \rightarrow \langle \gamma, \alpha \rangle$ is a rule in Q_h^i , and $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment

such that $\gamma = \gamma_1 e^i \gamma_2$ and $\alpha = \alpha_1 e_h^i \alpha_2$

then $X \rightarrow \langle \gamma_1 X_k \gamma_2, \alpha_1 X_k \alpha_2 \rangle$ is added to Q_h^i ,

where k is an index.

Two special “glue” rules are added to Q_h^i

$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$ and $S \rightarrow \langle X_1, X_1 \rangle$

Figure 4.12: Algorithm of hierarchical phrase extraction for paraphrasing

We use the same Chinese-to-English example of Figure 3.1 to illustrate the hierarchical

paraphrasing process. We assume Eh1 - “He likes you buy the book” is the selected backbone sentence hypothesis, and Eh2 - “He likes the books that you bought” is another hypothesis. Figure 4.13 shows the word alignments between the backbone and another hypothesis. Figure 4.14 shows the extracted hierarchical paraphrases from MT system h1’s translation, and Figure 4.15 shows the extracted hierarchical paraphrases from MT system h2’s translation.

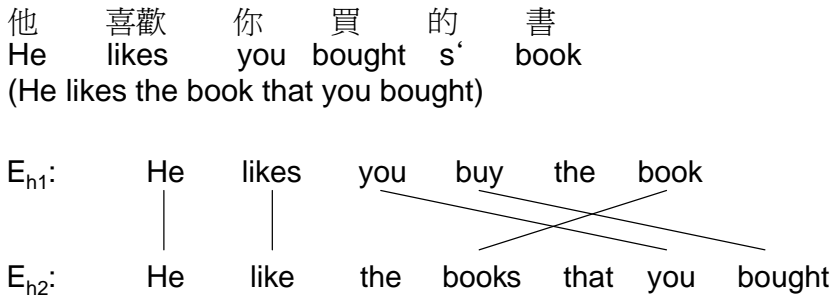


Figure 4.13: A backbone sentence (the translation Eh1), the translation Eh2 and the word alignment between the two.

- S →<S₁ X₂ , S₁ X₂ > (1)
- S →<X₁ , X₁ > (2)
- X →< He likes , He likes > (3)
-
- X →< you buy the book , you buy the book > (4)
- X →< you buy , you buy > (5)
- X →< book , book > (6)
- X →< X₁ the book , X₁ the book > (7)
- X →< you buy the X₁ , you buy the X₁ > (8)
- X →< X₁ the X₂ , X₁ the X₂ > (9)

Figure 4.14: Extracted hierarchical phrases from the source sentence and the translation by MT system h1

$$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle \quad (10)$$

$$S \rightarrow \langle X_1, X_1 \rangle \quad (11)$$

$$X \rightarrow \langle \text{He like}, \text{He likes} \rangle \quad (12)$$

.....

$$X \rightarrow \langle \text{you buy the book}, \text{the books that you bought} \rangle \quad (13)$$

$$X \rightarrow \langle \text{you buy}, \text{you bought} \rangle \quad (14)$$

$$X \rightarrow \langle \text{book}, \text{books} \rangle \quad (15)$$

$$X \rightarrow \langle X_1 \text{ the book}, \text{the books that } X_1 \rangle \quad (16)$$

$$X \rightarrow \langle \text{you buy the } X_1, \text{the } X_1 \text{ that you bought} \rangle \quad (17)$$

$$X \rightarrow \langle X_1 \text{ the } X_2, \text{the } X_2 \text{ that } X_1 \rangle \quad (18)$$

Figure 4.15: Extracted hierarchical phrases from the source sentence and the translation by MT system h2

Given the extracted hierarchical phrases of Figure 4.14 and Figure 4.15, the *hierarchical paraphrasing model* would have the chance of getting the correct translation - “He likes the book that you bought”. Figure 4.16 shows the derivation of a synchronous CFG by using rules in Figure 4.14 and Figure 4.15.

$$\begin{aligned} \langle S_1, S_1 \rangle &\Rightarrow \langle S_2 X_3, S_2 X_3 \rangle && \text{using (1) or (9)} \\ &\Rightarrow \langle X_4 X_3, X_4 X_3 \rangle && \text{using (2) or (10)} \\ &\Rightarrow \langle \text{He likes } X_3, \text{He likes } X_3 \rangle && \text{using (3)} \\ &\Rightarrow \langle \text{He likes } X_5 \text{ the } X_6, \text{He likes the } X_6 \text{ that } X_5 \rangle && \text{using (18)} \\ &\Rightarrow \langle \text{He likes you buy } X_6, \text{He likes the } X_6 \text{ that you bought} \rangle && \text{using (14)} \\ &\Rightarrow \langle \text{He likes you buy the book}, \text{He likes the book that you bought} \rangle && \text{using (6)} \end{aligned}$$

Figure 4.16: Derivation of a synchronous CFG by using rules in Figure 4.14 and Figure 4.15.

4.3.2 Model

To build our *Hierarchical Phrase-based Re-decoding Model*, we need to first provide definitions about the estimation of confidence scores.

Definition 4. For the backbone of the i -th input sentence and translation of MT system h , one of the extracted paraphrasing rules j can be represented as $X \rightarrow \langle \gamma_j^i, \alpha_{h,j}^i \rangle$ and its confidence score for the system h can be represented as an indicator:

$$CS(\gamma_j^i, \alpha_{h,j}^i) = \begin{cases} 1 & \text{if } X \rightarrow \langle \gamma_j^i, \alpha_{h,j}^i \rangle \text{ occurs in } Q_h^i \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

Definition 5. For the backbone of i -th input sentence and one of the extracted paraphrasing rules j , we can represent its overall confidence score as a weighted summarization over all MT systems' individual confidence score toward it:

$$\sum_{h=1}^{N_s} \lambda_h * CS(\gamma_j^i, \alpha_{h,j}^i) \quad (4.7)$$

Where N is the total number of MT systems, and λ_h denotes the weight of MT system h

Definition 6. For the backbone (E^i) of the i -th input sentence, we can define the confidence score for its combination result \bar{E}^i as follows:

$$\log p(\bar{E}^i | E^i) = \sum_{j=1}^J \left(\sum_{h=1}^{N_s} \lambda_h * CS(\gamma_j^i, \alpha_{h,j}^i) \right) + \lambda_p * J + \lambda_l * \log(LM(\bar{E}^i)) + \lambda_w * length(\bar{E}^i) \quad (4.8)$$

J is the total number of phrases for the given sentence. λ_h is the weight of MT system h . λ^p is phrase penalty, which controls the preference of phrase length. λ_w is word penalty, which controls the preference of hypothesis length. LM is a general language model, weighted by λ^l . In this combination model, all weights as well as word and phrase penalty can be trained discriminatively for Bleu score using Minimum Error Rate Training (MERT) procedure (Och 2004).

4.3.3 Decoding

Given the backbone of an input source and the corresponding paraphrasing rules, the decoder performs a search for the single most probable derivation via the CKY algorithm with a Viterbi approximation. The path of the search is our combination result. The single most probable derivation can be represented as

$$\bar{E}_{best}^i = \arg \max_{\bar{E}^i} \log p(\bar{E}^i | E^i)$$

4.3.4 Experiment

The experiments are conducted and reported on two datasets: One dataset includes Chinese-English system translations and references from DARPA GALE 2008 (GALE Chi-Eng Dataset). The other one includes Chinese-English system translations and references and from NIST 2008 (NIST Chi-Eng Dataset).

4.3.4.1 Setting

We use the same setting as in Section 4.2.4.1. For the reader’s convenience, we describe it here again:

GALE Chi-Eng Dataset: The GALE Chi-Eng Dataset consists of source sentences in Chinese, corresponding machine translations of 12 MT systems and four human reference translations in English. It also provides word alignments between source and translation sentences. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 422 sentences and the test set also includes 422 sentences. Among the five systems, “rwth-pbt-sh” performs the best in BLEU, and since we are tuning toward BLEU, we regard “rwth-pbt-sh” as the top MT system.

NIST Chi-Eng Dataset: The NIST Chi-Eng Dataset also consists of source sentences in Chinese, corresponding machine translations of multiple MT systems and four human reference translations in English, but word alignments between source and translation sentences are not included. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 524 sentences and the test set includes 788 sentences. Among the five systems, “Sys 03” performs the best in BLEU, and since we are tuning toward BLEU, we regard “Sys 03” as the top MT system.

4.3.4.2 Results

	BLEU	TER	MET
Sys rwth-pbt-sh	32.63	58.67	58.98
<i>phrase-based re-decoding model (baseline)</i>	31.02	60.62	57.32
<i>hierarchical phrase-based re-decoding model</i>	32.11	59.19	58.40
<i>Confusion Network (baseline)</i>	33.04	57.08	59.44
<i>paraphrasing model</i>	33.16	56.63	59.46
<i>hierarchical paraphrasing model</i>	33.09	56.68	59.34

Table 4.9: Comparing the performance of the *hierarchical paraphrasing model* with top MT system, re-decoding models, the *paraphrasing model* and *confusion network decoding* for GALE Chi-Eng Dataset.

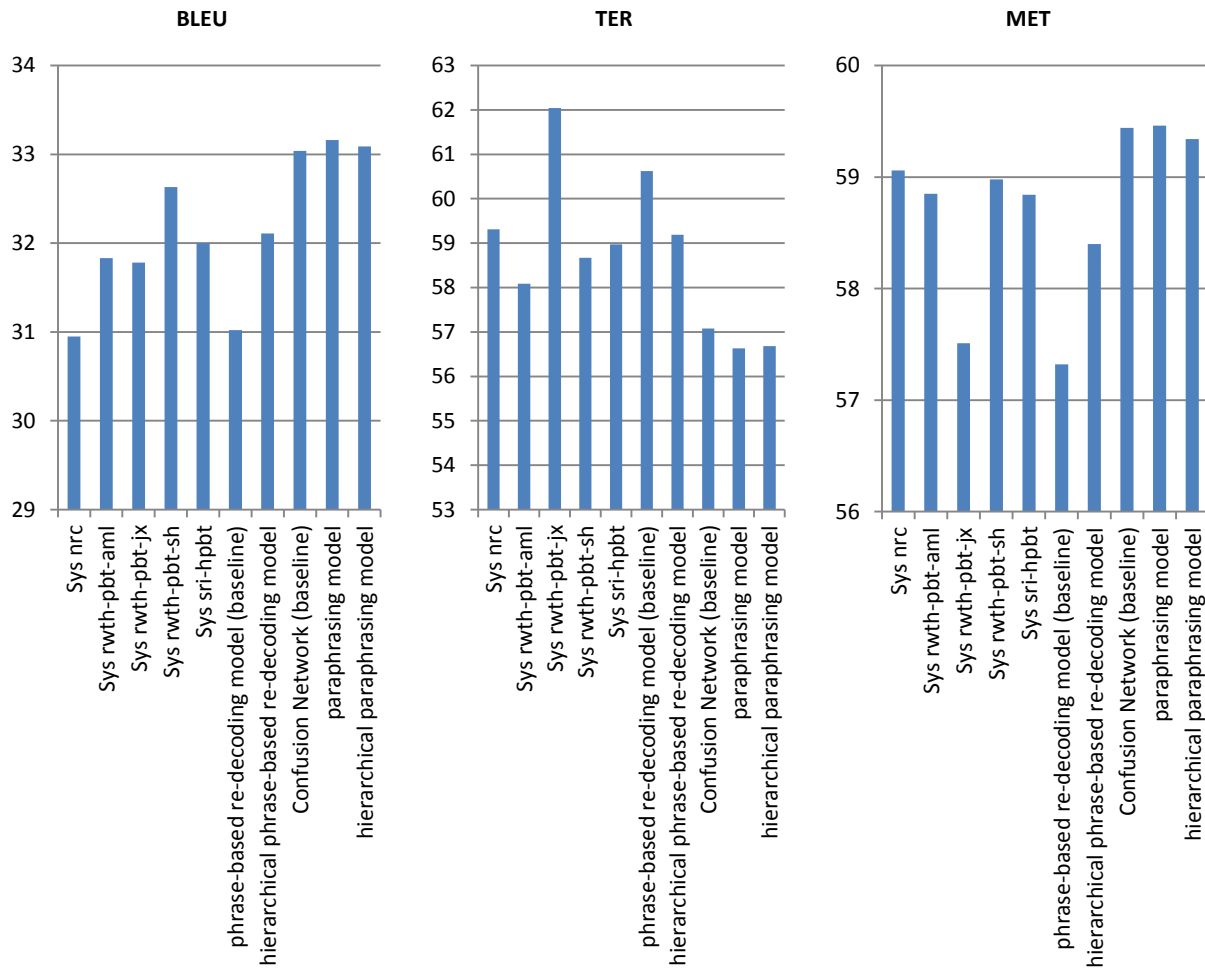


Figure 4.17: Comparing the performance of *hierarchical paraphrasing model* with all other systems for GALE Chi-Eng Dataset.

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>paraphrasing model</i>	32.65	55.11	56.17
<i>hierarchical paraphrasing model</i>	32.59	55.06	56.19

Table 4.10: Comparing the performance of the *hierarchical paraphrasing model* with top MT system, the *paraphrasing model* and *confusion network decoding* for NIST Chi-Eng Dataset.

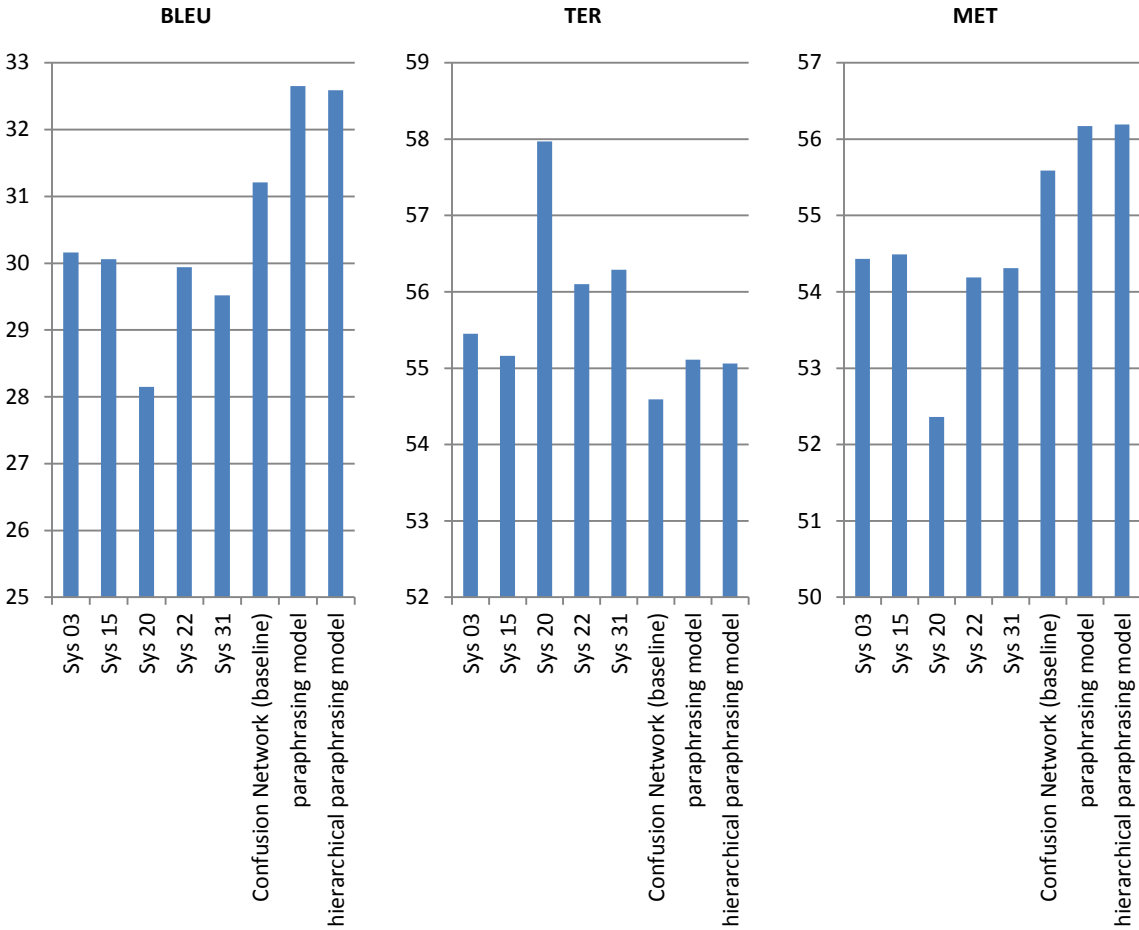


Figure 4.18: Comparing the performance of *hierarchical paraphrasing model* with all other systems for NIST Chi-Eng Dataset.

	BLEU	TER	MET
Sys 31	48.40	45.55	70.67
<i>Confusion Network (baseline)</i>	48.56	43.81	70.67
<i>paraphrasing model</i>	49.33	45.08	70.87
<i>hierarchical paraphrasing model</i>	49.46	44.84	70.99

Table 4.11: Comparing the performance of the *hierarchical paraphrasing model* with top MT system, the *paraphrasing model* and *confusion network decoding* for NIST Ara-Eng Dataset.

From Table 4.9 and 4.10, we see that the *hierarchical paraphrasing model* outperforms the best MT system and *confusion network decoding model* for the two datasets. And from Table 4.9, similar to the *paraphrasing model*, we also find that the *hierarchical paraphrasing model* performs better than both re-decoding models.

Table 4.9 and 4.10 are Chinese-English datasets, which we evaluate our *hierarchical paraphrasing model* during and after the development process. To provide a more objective evaluation, we evaluate our *hierarchical paraphrasing model* on NIST Ara-Eng Dataset as a blind test. The results are shown in Table 4.11. we see that the *hierarchical paraphrasing model* still achieves the better performance in BLEU in comparison with the best MT system and the *confusion network decoding model*, which demonstrates the *hierarchical paraphrasing model*'s robustness and consistency.

Although the experimental results show that *hierarchical paraphrasing model* performs well, there is almost no difference in performance in any of the three metrics when compared against the *paraphrasing model*. However, in Chapter 3, we did see that the *Hierarchical Phrase-based Re-decoding Model* outperforms *Phrase-based Re-decoding Model* in all three metrics. So there are two questions emerging:

1. What is the reason that the hierarchical phrase-based technique works better for the re-decoding strategy than the paraphrasing strategy?
2. Is it possible that the *hierarchical paraphrasing model* can compensate for the *paraphrasing models* for some sentences, still bringing some benefits to the overall performance?

We try to answer the first question in this section, and leave the second question to Chapter 6 to answer when hybrid combination strategy is introduced. In fact, because the two questions are relevant, our observations shown in this section for answering the first question help answer the second question in Chapter 6.

To answer the first question, we note that the decoding targets for the re-decoding strategy and for the paraphrasing strategy are actually different. For the re-decoding strategy, the source sentence is decoded, and for the paraphrasing strategy, the backbone sentence is decoded. For the source sentence, more word reordering needs to be modeled because of the big difference of word ordering between the source language and the target language. On the other hand, for the backbone sentence, it has similar word ordering with the eventual combination results, because the MT systems already tried their best to model word reordering. And since a major strength of the hierarchical phrase-based technique is that it has a stronger ability to address word reordering, we hypothesis that when more word reordering is needed, hierarchical phrase-based techniques can bring more benefits in comparison with its counterpart using the non-hierarchical phrase-based technique. In other words, when less word reordering is necessary, the hierarchical phrase-based technique seems unlikely to bring significant improvement over its counterpart using the non-hierarchical phrase-based technique. The observation that the hierarchical phrase-based technique works better for the re-decoding strategy than the paraphrasing strategy

can support this hypothesis. In order to obtain more evidence to prove this hypothesis, we carried out the following experiment in the next section, Section 4.3.4.3.

4.3.4.3 Analysis of Paraphrasing Different Backbones

The quality of a given translation hypothesis is related to word choices and their orders. Based on this fact, we make an assumption that if a given hypothesis for paraphrasing is well translated, it is more likely to have relatively correct word order, so less word reordering is needed. On the other hand, if a given hypothesis for paraphrasing is poorly translated, it is more likely to have relatively incorrect word order, so more word reordering needs to be done.

Based on this assumption, it can be expected that when a well-translated hypothesis is paraphrased, hierarchical phrase-based techniques would be less likely to bring significant improvement than its counterpart using non-hierarchical phrase-based techniques, but when a poorly translated hypothesis is paraphrased, hierarchical phrase-based techniques can bring more benefits in comparison with its counterpart using the non-hierarchical phrase-based technique.

From Table 3.2, we observe that although the five MT systems are the selected top 5 systems in the NIST Chi-Eng Dataset, their performances are still quite different. For each MT system, we paraphrase its translations using the paraphrasing model and the hierarchical paraphrasing model separately, aiming to compare the performances of the two models on each MT system. In other words, we do not first do backbone selection. Every MT system's translation is regarded as a backbone. The results are shown in Figure 4.19 - 4.21.

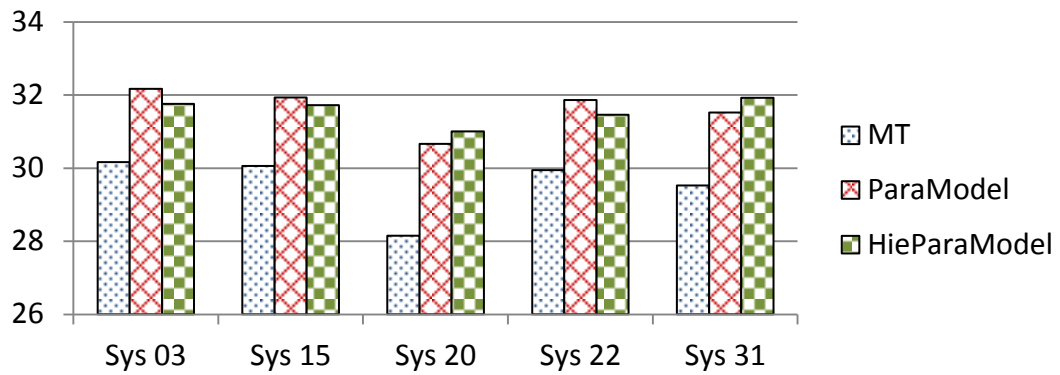


Figure 4.19: Comparing the performance using BLEU of the MT systems, the *paraphrasing model* and the *hierarchical paraphrasing model* on the NIST Chi-Eng Dataset.

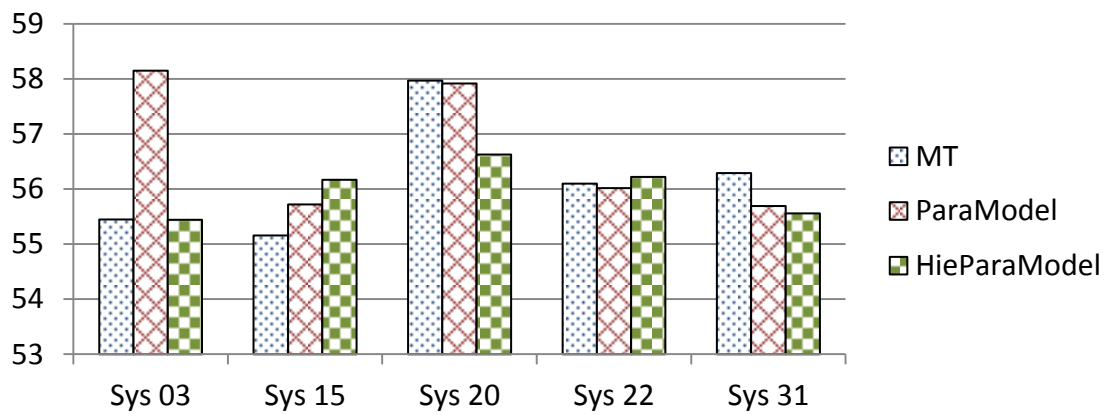


Figure 4.20: Comparing the performance using TER of the MT systems, the *paraphrasing model* and the *hierarchical paraphrasing model* on the NIST Chi-Eng Dataset.

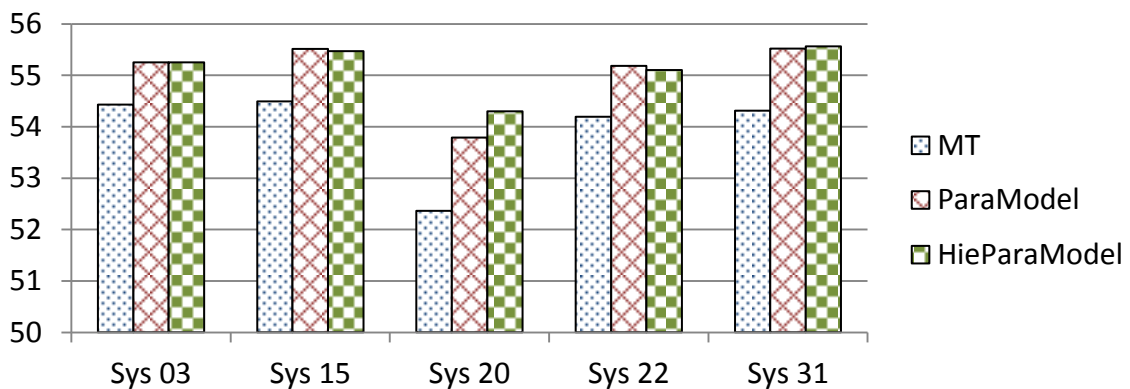


Figure 4.21: Comparing the performance using MET of the MT systems, the *paraphrasing model* and the *hierarchical paraphrasing model* on the NIST Chi-Eng Dataset.

Among the five MT systems, “Sys 20” and “Sys 31” perform poorer than the other three MT systems. When we paraphrase the two systems, we find that the *hierarchical paraphrasing model* outperforms the *paraphrasing model* in all three metrics. Based on these results, we show that when more word reordering is needed, hierarchical phrase-based techniques can bring more benefit in comparison with non-hierarchical phrase-based techniques.

In fact, this finding motivates us to develop a hybrid combination structure to integrate these various paraphrasing results using a hypothesis selection procedure, which will be introduced in detail in Chapter 6.

4.3.4.4 Analysis of Syntactic Paraphrase Extraction

In Section 4.3.1, we introduced our algorithm of hierarchical phrase extraction for paraphrasing - a synchronous CFG with the form $X \rightarrow \langle \gamma, \alpha \rangle$, where X is any non-terminal in the grammar; γ and α are strings of terminals and non-terminals, which do not consider any syntactic information or restriction. Similar to the analysis of syntactic paraphrases used in the *paraphrasing model*, we investigate the effect of syntactic paraphrases used in the *hierarchical paraphrasing model* by using three different extraction methods.

We use the same notation as we used in Section 4.3.1: for an i -th sentence, we use E^i and e^i to represent the backbone and one of its phrases, respectively. E_h^i represents the translation of MT system h , and e_h^i is one phrase of E_h^i . We use Q_h^i to denote the set of the paraphrasing rules for sentence i and MT system h . Our three different extraction methods – D, E and F are shown in Figures 4.22, 4.23 and 4.24, respectively. The differences with our algorithm of hierarchical phrase extraction shown in Section 4.3.1 are highlighted.

Extraction Method D:

If $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment, and e^i is a constituent

then $X \rightarrow \langle e^i, e_h^i \rangle$ is added to Q_h^i .

If $X \rightarrow \langle \gamma, \alpha \rangle$ is a rule in Q_h^i , and $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment,

and e^i is a constituent such that $\gamma = \gamma_1 e^i \gamma_2$ and $\alpha = \alpha_1 e_h^i \alpha_2$

then $X \rightarrow \langle \gamma_1 X_k \gamma_2, \alpha_1 X_k \alpha_2 \rangle$ is added to Q_h^i ,

where k is an index.

Two special “glue” rules are added to Q_h^i

$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$ and $S \rightarrow \langle X_1, X_1 \rangle$

Figure 4.22: Hierarchical phrase extraction method D

Extraction Method E:

If $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment, and e^i and e_h^i are both constituents

then $X \rightarrow \langle e^i, e_h^i \rangle$ is added to Q_h^i .

If $X \rightarrow \langle \gamma, \alpha \rangle$ is a rule in Q_h^i , and $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment,

and e^i and e_h^i are both constituents such that $\gamma = \gamma_1 e^i \gamma_2$ and $\alpha = \alpha_1 e_h^i \alpha_2$

then $X \rightarrow \langle \gamma_1 X_k \gamma_2, \alpha_1 X_k \alpha_2 \rangle$ is added to Q_h^i ,

where k is an index.

Two special “glue” rules are added to Q_h^i

$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$ and $S \rightarrow \langle X_1, X_1 \rangle$

Figure 4.23: Hierarchical phrase extraction method E

Extraction Method F:

If $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment, and e^i and e_h^i are both constituents with the same constituent types, such as NP, VP, PP... etc.

then $X \rightarrow \langle e^i, e_h^i \rangle$ is added to Q_h^i .

If $X \rightarrow \langle \gamma, \alpha \rangle$ is a rule in Q_h^i , and $\langle e^i, e_h^i \rangle$ is consistent with monolingual word alignment, and e^i and e_h^i are both constituents with the same constituent types, such as NP, VP, PP... etc.

such that $\gamma = \gamma_1 e^i \gamma_2$ and $\alpha = \alpha_1 e_h^i \alpha_2$

then $X \rightarrow \langle \gamma_1 X_k \gamma_2, \alpha_1 X_k \alpha_2 \rangle$ is added to Q_h^i ,

where k is an index.

Two special “glue” rules are added to Q_h^i

$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$ and $S \rightarrow \langle X_1, X_1 \rangle$

Figure 4.24: Hierarchical phrase extraction method F

In the three extraction methods, the constituents and their types are determined by the Stanford Parser. The combination results using these methods are shown in Table 4.12.

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>hierarchical paraphrasing model</i>	32.59	55.06	56.19
<i>hierarchical paraphrasing model with Extraction Method D</i>	32.12	55.11	56.21
<i>hierarchical paraphrasing model with Extraction Method E</i>	32.00	54.81	56.12
<i>hierarchical paraphrasing model with Extraction Method F</i>	31.77	55.24	55.83

Table 4.12: Comparing the performance of *hierarchical paraphrasing model* using different extraction methods for NIST Chi-Eng Dataset.

Table 4.12 shows that syntactic paraphrases give no improvement in comparison with the basic extraction rules in section 4.3.1. The results might be explained by the same reason we mentioned for the *paraphrasing model* in Section 4.2.4.4; restricting paraphrases to be syntactic paraphrases makes the paraphrasing model to retain the same or similar overall syntactic structure of the backbone hypothesis. As a result, only fewer paraphrases are extracted and thus the backbone has less chance to be paraphrased.

4.3.4.5 Analysis of the Addition of Syntactic Features

In Section 4.2.4.5, to investigate the impact of syntactic information, we weighted syntactic phrases in the paraphrase table used in our *paraphrasing model*, and found that the consideration of syntactic phrases does not bring benefits. Here for *hierarchical paraphrasing model*, we adopt a similar strategy; we add the following different features individually in (4.8).

Feature D

$$\text{syn}(\gamma_j^i, \alpha_{h,j}^i) = \begin{cases} 1 & \text{if } X \rightarrow \langle \gamma_j^i, \alpha_{h,j}^i \rangle \text{ is in } Q_h^i \text{ obtained by Extraction Method D} \\ 0 & \text{otherwise} \end{cases}$$

Feature E

$$\text{syn}(\gamma_j^i, \alpha_{h,j}^i) = \begin{cases} 1 & \text{if } X \rightarrow \langle \gamma_j^i, \alpha_{h,j}^i \rangle \text{ is in } Q_h^i \text{ obtained by Extraction Method E} \\ 0 & \text{otherwise} \end{cases}$$

Feature F

$$\text{syn}(\gamma_j^i, \alpha_{h,j}^i) = \begin{cases} 1 & \text{if } X \rightarrow \langle \gamma_j^i, \alpha_{h,j}^i \rangle \text{ is in } Q_h^i \text{ obtained by Extraction Method F} \\ 0 & \text{otherwise} \end{cases}$$

Each feature is attached with a weight, obtained from MERT process. In the previous section, Method D, E and F are hard constraints about syntactic paraphrases. In this section, Feature D, E and F are soft constraints on syntactic paraphrases.

In the three features, the constituents and their types are determined by the Stanford Parser.

The combination results using these methods are shown in Table 4.13.

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>hierarchical paraphrasing model</i>	32.59	55.06	56.19
<i>hierarchical paraphrasing model with Feature D</i>	32.54	55.03	56.25
<i>hierarchical paraphrasing model with Feature E</i>	31.91	56.19	55.60
<i>hierarchical paraphrasing model with Feature F</i>	31.81	54.58	55.92

Table 4.13: Comparing the performance of *hierarchical paraphrasing model* using different features about syntactic paraphrases for NIST Chi-Eng Dataset.

Table 4.13 shows that the features for syntactic paraphrases still give no improvement over the basic extraction rules in section 4.3.1.

4.4 Conclusions

In this chapter, we propose two models, which view combination as a paraphrasing process with the use of a set of paraphrases, learned from monolingual word alignments between a selected best translation hypothesis and other hypotheses. *The paraphrasing model* relies on string-to-string paraphrases to paraphrase the backbone translation hypothesis while *the hierarchical paraphrasing model* uses hierarchical paraphrases to paraphrase the backbone translation hypothesis. Our experimental results show that they have similar performances, and both of them give superior performance compared with the best single translation engine and outperform the re-decoding model and *confusion network decoding*.

Our experiments show that the addition of simple syntactic constraints in both models does not yield improvement. Moreover, we also carried out some investigational experiments and found out that if a given hypothesis for paraphrasing is well translated, *the hierarchical paraphrasing model* would not bring benefits to *paraphrasing model*. But on the other hand, if a given hypothesis for paraphrasing is poorly translated, *the hierarchical paraphrasing model* is more likely to improve than the *paraphrasing model*.

Chapter 5

Sentence-level Combination

In Chapter 3 and Chapter 4, we introduced our phrase-level combination techniques. In this chapter, we present a sentence-level combination model using a log linear model with some novel features to select the best translation hypothesis among multiple candidates of translation hypotheses. In comparison with phrase-level combination, the advantage of sentence-level combination is that because the whole sentence can be used to evaluate the translation quality, it allows for easy integration of complex syntactic features that would be too expensive to use during the decoding process of phrase-level combination techniques. That enables us to do relatively deeper analysis to evaluate the translation quality and represent syntactic and semantic features in addition to consensus in a log linear model. On the other hand, the limit of sentence-level combination is that it not generate any new fused hypothesis from the given multiple translation hypotheses.

In order to identify ungrammatical hypotheses from a set of candidate translations, we utilize grammatical knowledge in the target language, including using a supertag-based structural language model that expresses syntactic dependencies between words, described in Section 5.2, and a syntactic error detector based on a feature-based lexicalized tree adjoining grammar

(FB-LTAG) to recognize ungrammatical translations, described in Section 5.3. In addition, we hypothesize that, for a good translation, most of the predicate-argument structures from the source language should be retained in order to preserve the semantics. That is, predicate-argument structures and argument types in source and target should be the same in most cases. Based on this assumption, we develop a measure of how likely arguments should be aligned, shown in Section 5.4.

In this chapter, our experimental goal is to use our sentence-level model to select a translated sentence from multiple MT systems. In chapter 6, we will propose several hybrid combination structures to integrate our phrase-level combination models and the sentence-level combination model, in which the sentence-level combination model makes the final decision among all fused translations generated by the phrase-level models.

5.1 Related Work

In recent years, there has been a burgeoning interest in incorporating syntactic structure into statistical machine translation (SMT) models (e.g., Galley et al., 2006; DeNeeffe and Knight 2009; Quirk et al., 2005). In addition to modeling syntactic structure in the decoding process, a methodology for candidate translation selection has also emerged. This methodology first generates multiple candidate translations followed by rescoring using global sentence-level syntactic features to select the final translation.

Candidate translation selection is usually applied in two scenarios: one scenario is as part of an n-best reranking (Och et al., 2004; Hasan et al., 2006), where n-best candidate translations are generated through a decoding process. Hasan et al., (2006) focused on monolingual syntax and investigated the effect of directly using the log-likelihood of the output of a HMM-based

supertagger, and found it did not improve performance significantly. It is worth noticing that this log-likelihood is based on supertagged n-gram LM, which is one type of class-based n-gram LM, so it does not model explicit syntactic dependencies between words in contrast to the work we describe in this thesis. Hardmeier et al., (2012) use tree kernels over constituency and dependency parse trees for either the input or output sentences to identify constructions that are difficult to translate in the source language, and doubtful syntactic structures in the output language. The tree fragments extracted by their tree kernels are similar to our elementary trees but they only regard them as the individual inputs of support vector machine regression while binary relations of our elementary trees are considered in a formulation of a structural language model. Och et al., (2004) investigated various syntactic feature functions to rerank the n-best candidate translations. Most features are syntactically motivated and based on alignment information between the source sentence and the target translation. The results are rather disappointing. Only the non-syntactic IBM model 1 yielded significant improvement. All other tree-based feature functions had only a very small effect on the performance.

The other scenario for candidate translation selection is translation selection or reranking (Hildebrand and Vogel 2008; Callison-Burch et al., 2012), where candidate translations are generated by different decoding processes or different decoders. Our approaches in Section 5.2 and 5.4 belong to this scenario.

As for the identification of ungrammatical hypotheses, researchers developed a variety of methods used for grammar checking, including statistic-based approaches, rule-based approaches and the mix of both. For example, Alam et al., (2006) and Wu et al., (2006) rely on N-gram language model to consider if a given sentence has grammatical problems: if a sentence has grammatical problems, it is likely to have uncommon word sequences, result in lower score of language model. Huang et al. (2010) extracted erroneous and correct patterns of consecutive

words from the data of an online-editing diary website. Some researchers use a set of hand crafted rules out of words and POS tags (Naber, 2003), or out of parsing results (Heidorn, 2000) to detect errors. Jensen et al. (1993) utilize a parsing procedure to detect errors: each sentence must be syntactically parsed; a sentence is considered incorrect if parsing does not succeed. Stymne and Ahrenberg (2010) utilized an existing rule-based Swedish grammar checker, as a post-processing tool for their English-Swedish translation system. They tried to fix the ungrammatical translation phrases by applying the grammar checker’s correction suggestions. In contrast to their using an existing grammar checker, we developed our own novel grammar checker for translated English in order to better controlling the quality of error detection and have more insights about how to correct errors in translation context.

5.2 Supertagged Dependency Language Model

In this section, we present a novel, structured language model - Supertagged Dependency Language Model to model the syntactic dependencies between (Ma and McKeown, 2013). The goal is to identify ungrammatical hypotheses from given candidate translations using grammatical knowledge in the target language that expresses syntactic dependencies between words. To achieve that, we propose a novel Structured Language Model (SLM) - Supertagged Dependency Language Model (SDLM) to model the syntactic dependencies between words. Supertag (Bangalore and Joshi, 1999) is an elementary syntactic structure based on Lexicalized Tree Adjoining Grammar (LTAG). Traditional supertagged n-gram LM predicts the next supertag based on the immediate words to the left with supertags, so it can not explicitly model long-distance dependency relations. In contrast, SDLM predicts the next supertag using the words with supertags on which it syntactically depend, and these words could be anywhere and arbitrarily far apart in a sentence. A candidate translation’s grammatical degree or “fluency” can

be measured by simply calculating the SDLM likelihood of the supertagged dependency structure that spans the entire sentence.

To obtain the supertagged dependency structure, the most intuitive way is through a LTAG parser (Schabes et al., 1988). However, this could be very slow as it has time complexity of $O(n^6)$. Another possibility is to follow the procedure in (Joshi and Srinivas 1994, Bangalore and Joshi, 1999): use a HMM-based supertagger to assign words with supertags, followed by derivation of a shallow parse in linear time based on only the supertags to obtain the dependencies. But since this approach uses only the local context, in (Joshi and Srinivas 1994), they also proposed another greedy algorithm based on supertagged dependency probabilities to gradually select the path with the maximum path probability to extend to the remaining directions in the dependency list.

In contrast to the LTAG parsing and supertagging-based approaches, we propose an alternative mechanism: first we use a state-of-the-art constituent parser to obtain the parse of a sentence, and then we extract elementary trees with dependencies from the parse to assign each word with an elementary tree. The second step is similar to the approach used in extracting elementary trees from the TreeBank (Xia, 1999; Chen and Vijay-Shanker, 2000).

Aside from the consideration of time complexity, another motivation of this two-step mechanism is that, compared with LTAG parsing, the mechanism is more flexible for defining syntactic structures of elementary trees for our needs. Because those structures are defined only within the elementary tree extractor, we can easily adjust the definition of those structures within the extractor and avoid redesigning or retraining our constituent parser.

We experiment with sentence-level translation combination of five different translation systems of the NIST Chi-Eng Dataset; the goal is for the sentence-level combination system to

select the best translation for each input source sentence among the translations provided by the five systems.

5.2.1 LTAG and Supertag

LTAG (Joshi et al., 1975; Schabes et al., 1988) is a formal tree rewriting formalism, which consists of a set of elementary trees, corresponding to minimal linguistic structures that localize dependencies, including long-distance dependencies, such as predicate-argument structure. Each elementary tree is associated with at least one lexical item on its frontier. The lexical item associated with an elementary tree is called the anchor in that tree; an elementary tree thus serves as a description of syntactic constraints of the anchor. The elementary syntactic structures of elementary trees are called supertags (Bangalore and Joshi, 1999), in order to distinguish them from the standard part-of-speech tags.

Elementary trees are divided into initial and auxiliary trees. Initial trees are those for which all non-terminal nodes on the frontier are substitutable. Auxiliary trees are defined as initial trees, except that exactly one frontier, non-terminal node must be a foot node, with the same label as the root node. Two operations - substitution and adjunction - are provided in LTAG to combine elementary trees into a derived tree.

5.2.2 Elementary Tree Extraction

We use an elementary tree extractor, a modification of (Chen and Vijay-Shanker, 2000), to serve our purpose. Heuristic rules were used to distinguish arguments from adjuncts, and the extraction process can be regarded as a process that gradually decomposes a constituent parse to multiple elementary trees and records substitutions and adjunctions. From elementary trees, we can obtain supertags by only considering syntactic structure and ignoring anchor words. Take the sentence –

“The hungry boys ate dinner” as an example; the constituent parse is shown in Figure 5.1, and extracted supertags are shown in Figure 5.2.

In Figure 5.2, dotted lines represent the operations of substitution and adjunction. Note that each word in a translated sentence would be assigned exactly one elementary syntactic structure which is associated with a unique supertag id for the whole corpus. Different anchor words could own the same elementary syntactic structure and would be assigned the same supertag id, such as “*a1*” for “boys” and “dinner”.

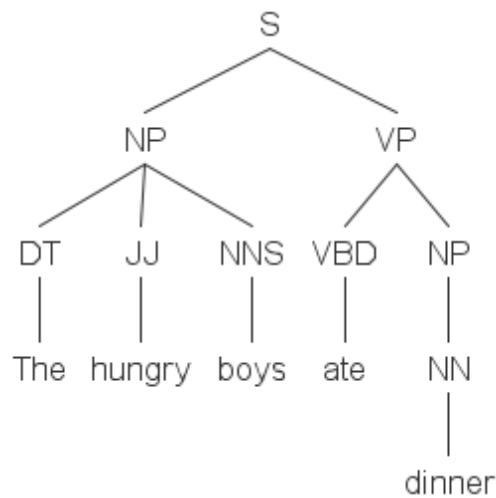


Figure 5.1: Parse of “The hungry boys ate dinner”

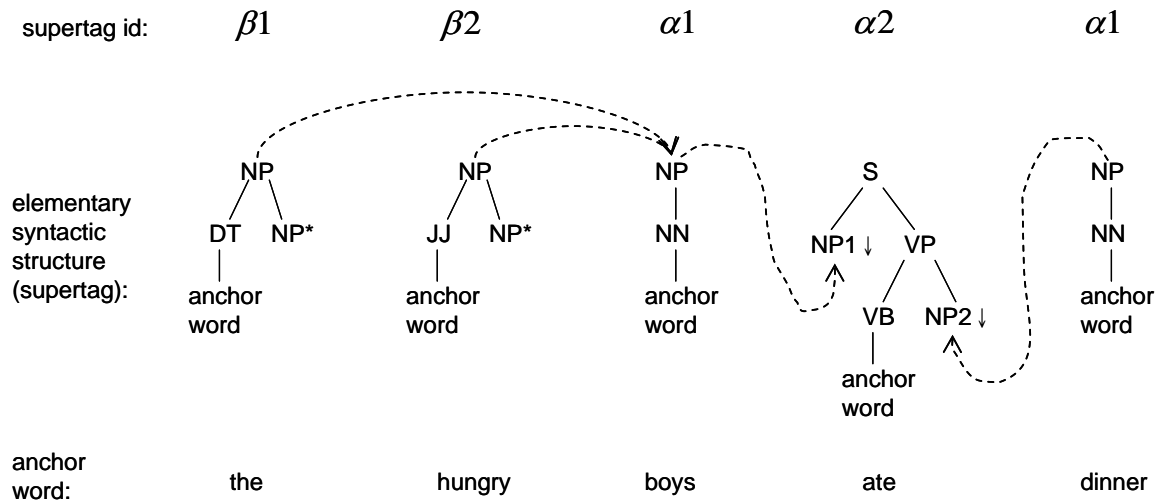


Figure 5.2: Extracted elementary trees of “The hungry boys ate dinner”

5.2.3 Model

Bangalore and Joshi (1999) gave a concise description for dependencies between supertags: “A supertag is dependent on another supertag if the former substitutes or adjoins into the latter”. Following this description, for the example in Figure 1 (b), supertags of “the” and “hungry” are dependent on the supertag of “boys”, and supertags of “boys” and “dinner” are dependent on the supertag of “ate”. These dependencies between supertags also provide the dependencies between anchor words.

Since the syntactic constraints for each word in its context are decided and described through its supertag, the likelihood of SDLM for a sentence could also be regarded as the degree of violations of the syntactic constraints on all words in the sentence. Consider a sentence $S = w_1 w_2 \dots w_n$ with corresponding supertags $T = t_1 t_2 \dots t_n$. We use $d_i=j$ to represent the dependency relations for words or supertags. For example, $d_3 = 5$ means that w_3 depends on w_5 or t_3 depends on t_5 . We propose five different bigram SDLM as follows and evaluate their effects in the following.

$$\prod_i P(w_i t_i | w_{d_i} t_{d_i}) \quad \text{SDLM model(1)}$$

$$\prod_i P(w_i t_i | w_{d_i} t_{d_i}) \approx \prod_i P(t_i | t_{d_i}) P(w_i | t_i) \quad \text{SDLM model(2)}$$

$$\prod_i P(t_i | t_{d_i}) \quad \text{SDLM model(3)}$$

$$\prod_i P(w_i | t_i) \quad \text{SDLM model(4)}$$

$$\prod_i P(w_i | w_{d_i}) \quad \text{SDLM model(5)}$$

SDLM model (2) is the approximation form of model (1); models (3) and (4) are individual terms of model (2); model (5) models word dependencies based on elementary tree dependencies. The estimation of the probabilities is done using maximum likelihood estimations with Laplace smoothing. Take Figure 5.2 as an example; using model (1), the SDLM likelihood of “The hungry boys ate dinner” is

$$P(\text{the}, \beta_1 | \text{boys}, \alpha_1) * P(\text{hungry}, \beta_2 | \text{boys}, \alpha_1) * P(\text{boys}, \alpha_1 | \text{ate}, \alpha_2) * \\ P(\text{dinner}, \alpha_1 | \text{ate}, \alpha_2) * P(\text{ate}, \alpha_2 | \text{root})$$

In our experiment on sentence-level translation combination, we use a log-linear model to integrate all features including SDLM models. The corresponding weights are trained discriminatively for Bleu score using Minimum Error Rate Training (MERT).

5.2.4 Experiment

The experiments are conducted and reported on Chinese-English system translations and references and from NIST 2008 (NIST Chi-Eng Dataset).

5.2.4.1 Setting

We use the same setting of NIST Chi-Eng Dataset as in Section 3.2.4.1. The NIST Chi-Eng Dataset consists of source sentences in Chinese, corresponding machine translations of multiple MT systems and four human reference translations in English, but word alignments between source and translation sentences are not included. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 524 sentences and the test set includes 788 sentences. Among the five systems, “Sys 03” performs the best in BLEU, and since we are tuning toward BLEU, we regard “Sys 03” as the top MT system.

In terms of SDLM training, we extract elementary trees from automatically-generated parses of part of the Gigaword corpus (around one year of newswire of “`afp_eng`” in Gigaword 4) in addition to TreeBank-extracted elementary trees. In total, 17053 different elementary syntactic structures (17053 supertag ids) are extracted.

For the baseline combination system, we use the following feature functions in the log-linear model to calculate the score of a system translation.

- Sentence consensus toward MT systems’ translations based on TER
- Gigaword-trained 3-gram LM

$$\log p(E_i) = \sum_{s=1}^{N_s} (\lambda_s * \log(1 - TER(E_i, E_s))) + \lambda^l * \log(LM(E_i)) + \lambda^w * Length(E_i) \quad (5.1)$$

Where E is system hypothesis, N_s is system number, λ_s is system weight, λ^l is LM weight and λ^w is word penalty.

For testing SDLM, in addition to all features that the baseline combination system uses, we add single or multiple SDLM models in the log-linear model, and each SDLM model has its own weight.

5.2.4.2 Results

From Table 5.1, we see that the combination of SDLM model 3, 4 and 5 yields the best performance, which is better than the best MT system by a difference in Bleu score of 1.45, TER of 0.67 and METEOR of 1.25, and also better than the baseline combination system by a difference in Bleu score of 0.72, TER of 0.25 and METEOR of 0.44. Compared with SDLM model 5, which represents a type of word dependency LM without labels, the results show that adding appropriate syntactic “labels” (here, they are “supertags”) on word dependencies brings benefits.

	Bleu	TER	METEOR
Sys 03	30.16	55.45	54.43
Sys 15	30.06	55.16	54.49
Sys 20	28.15	57.97	52.36
Sys 22	29.94	56.10	54.19
Sys 31	29.52	56.29	54.31
LM+consensus (baseline)	30.89	55.03	55.24
LM+consensus + model 1	31.29	54.99	55.63
LM+consensus + model 2	31.25	55.23	55.37
LM+consensus + model 3	31.25	55.06	55.40
LM+consensus + model 4	31.44	54.70	55.54
LM+consensus + model 5	31.39	55.15	55.68
LM+consensus + model 3+model 4+ model 5	31.61	54.78	55.68

Table 5.1: Result of sentence-level translation combination using SDLM

In addition to automatic metrics, we also carry out a human evaluation task on Amazon Mechanical Turk (AMT) to compare the translation sentences produced by the *baseline* of using feature sets of *LM+consensus* and the combination model of using feature sets of *LM+consensus+SDLM (model3+model4+model5)*. We call the former *baseline* and the latter *CombUsingSDLM*.

208 sentences out of 788 sentences of the testing dataset of NIST Chi-Eng Dataset produced by *baseline* and *CombUsingSDLM* are different. So we asked native English speakers on AMT to compare only those translation pairs. The judgment is based on two dimensions separately: *fluency* and *adequacy*. The *fluency* evaluation asked Turk users to judge which translation between the two is more fluent, regardless of the correct meaning of the source, while the *adequacy* evaluation measures which translation between the two conveys the more correct meaning in the source sentence in comparison to the reference, even if the translation is not fully fluent. For *adequacy*, each comparison (hit) consists of one correct translation reference and the translation pair. For *fluency*, only the translation pair is provided. Each comparison for either *adequacy* or *fluency* task is done by 5 different native English speakers and the translation with more votes wins.

	better fluency	better adequacy
<i>baseline</i>	37.50	43.75
<i>CombUsingSDLM</i>	62.50	56.25

Table 5.2: Experimental results of human evaluation on 208 different combination results

The results in Table 5.2 show that the performance of *CombUsingSDLM* is better than *baseline* from both the adequacy and the fluency perspectives, demonstrating the effective of SDLM. In

Table 5.1, although *CombUsingSDLM* yields better performance than the baseline by a difference in Bleu score of 0.72, TER of 0.25 and METEOR of 0.44, the differences, while significant, are small, because these automatic metrics are not focusing on the syntactic quality of translations, which SDLM tries to improve. Syntactic problems in particular are sometimes caused by very few words yet they can result in misunderstanding of the entire sentence; those mistakes are not easily reflected through automatic metrics. On the other hand, we see that human evaluation is able to reflect a greater effect of SDLM: the difference in fluency of *CombUsingSDLM* is 25% and the difference in adequacy of *CombUsingSDLM* is 12.5%. These results show that the syntactic quality would not only influence translations' fluency but also play a crucial role in the understanding of translations.

5.3 Syntactic Error Detector

In the last section, we use SDLM to evaluate the syntactic correctness of a given translation but do not use any existing linguistic resources to evaluate the given translation's grammar. As illustrative examples, consider the following three ungrammatical English sentences:

1. Many young student play basketball.
2. John play basketball and Tom also play basketball.
3. John thinks to play basketball.

In 1 and 2 above, number agreement errors between the subjects and verbs (and quantifier) cause the sentences to be ungrammatical, while in 3, the infinitive following the main verb makes it ungrammatical. One could argue that an existing grammar checker could do the error detection for us, but if we use Microsoft Word 2010 (MS Word)'s grammar checker (Heidorn,

2000) to check the three sentences, the entire first sentence will be underlined with green wavy lines without any indication of what should be corrected, while no errors are detected in 2 and 3. However, an ideal grammatical detection should detect multiple errors, identify their types, and track the words in which they occur, such as Table 5.3.

Sentence	Error Types	Words
Many young student play basketball.	argeement	Many, student
John play basketball and Tom also play basketball.	argeement	John, play
	argeement	Tom, play
John thinks to play basketball.	mode	thinks

Table 5.3: Examples of ideal grammatical detection

To achieve this goal, we use XTAG English grammar (XTAG group, 2001), a feature-based lexicalized tree adjoining grammar (FB-LTAG), to serve this mission. In FB-LTAG, each lexical item is associated with a syntactic elementary tree, in which each node is associated with a set of feature-value pairs, called Attribute Value Matrices (AVMs). AVMs define the lexical item’s syntactic usage. Our syntactic error detection works by checking the AVM values of all lexical items within a sentence using a unification framework. Thus, we use the feature structures in the AVMs to detect multiple errors, identify their types, and track the words in which they occur (Ma and McKeown, 2012b; 2012c). In order to simultaneously detect multiple error types and track their corresponding words, we propose a new unification method which allows the unification procedure to continue when unification fails and also to propagate the failure information to relevant words. We call the modified unification a *fail propagation unification*.

Through the *fail propagation unification*, one is able to correct errors based on a unified consideration of all related words under the same error types. We present a simple mechanism to

correct part of the detected situations. In the experiment described in this section, we use our approach to detect and correct translations of five single statistical machine translation systems run on the GALE Chi-Eng Dataset. The results show that most of the corrected translations are improved.

5.3.1 Background

We briefly introduce the FB-LTAG formalism and XTAG grammar in this section.

5.3.1.1 Feature-Based Lexicalized Tree Adjoining Grammars

FB-LTAG is based on tree adjoining grammar (TAG) proposed in (Joshi et al., 1975). The TAG formalism is a formal tree rewriting system, which consists of a set of elementary trees, corresponding to minimal linguistic structures that localize the dependencies, such as specifying the predicate-argument structure of a lexeme. Elementary trees are divided into initial and auxiliary trees. Initial trees are those for which all non-terminal nodes on the frontier are substitutable, marked with “ \downarrow ”. Auxiliary trees are defined as initial trees, except that exactly one frontier, nonterminal node must be a foot node, marked with “*”, with the same label with the root node. Two operations - substitution and adjunction are provided in TAG to adjoin elementary trees.

FB-LTAG has two important characteristics: First, it is a lexicalized TAG (Schabes, 1988). Thus each elementary tree is associated with at least one lexical item. Second, it is a feature-based lexicalized TAG (Vijay-Shanker & Joshi, 1988). Each node in an elementary tree is constrained by two sets of feature-value pairs (two AVMs). One AVM (top AVM) defines the relation of the node to its super-tree, and the other AVM (bottom AVM) defines the relation of the

node to its descendants. We use Figure 5.3 and Figure 5.4 to illustrate the substitution and adjunction operations with the unification framework respectively.

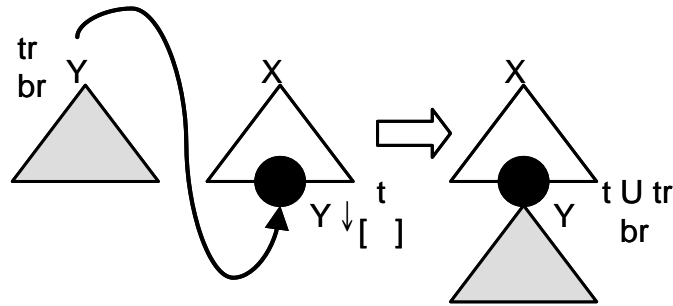


Figure 5.3: Substitution of FB-LTAG

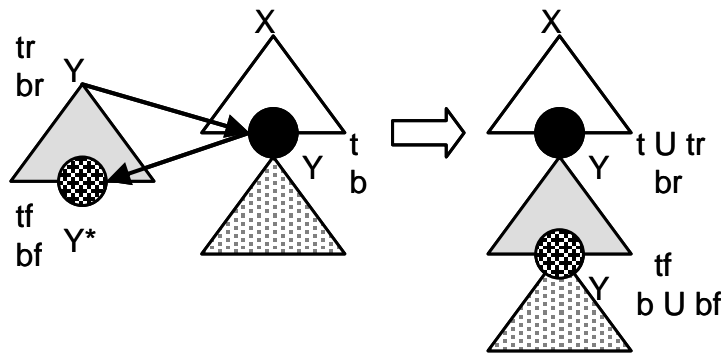


Figure 5.4: Adjunction of FB-LTAG

In Figure 5.3, we can see that the feature structure of a new node created by substitution inherits the union of the features of the original nodes. The top feature of the new node is the union of the top features of the two original nodes, while the bottom feature of the new node is simply the bottom feature of the top node of the substituting tree. In Figure 5.4, we can see that the node undergoing adjunction splits, and its top features unify with the top features of the root adjoining node, while its bottom features unify with the bottom features of the foot adjoining node.

5.3.1.2 XTAG English Grammar

XTAG English grammar (XTAG group, 2001) is designed using the FB-LTAG formalism released by UPENN in 2001. The range of syntactic phenomena that can be handled is large. It defines 57 major elementary trees (tree families) and 50 feature types, such as agreement, case, mode (mood), tense, passive, etc, for its 20027 lexical entries. Each lexical entry is associated with at least one elementary tree, and each elementary tree is associated with at least one AVM. For example, Figure 5.5 shows the simplified elementary tree of “saw”. “<number>” indicates the same feature value. For example, the feature – “arg_3rdsing” in the bottom AVM of root S should have the same feature value of “arg_3rdsing” in the top AVM of VP. In our implementation, it is coded using the same object in an object-oriented programming language. Since the feature value of mode in the top AVM of “S ↓” is “base”, we know that “saw” can only be followed by a sentence with a base verb. For example, “He saw me do that” shown in Figure 5.6(a) is a grammatical sentence while “He saw me to do that” shown in Figure 5.6(b) is an ungrammatical sentence because “saw” is not allowed to be followed by an infinitive sentence.

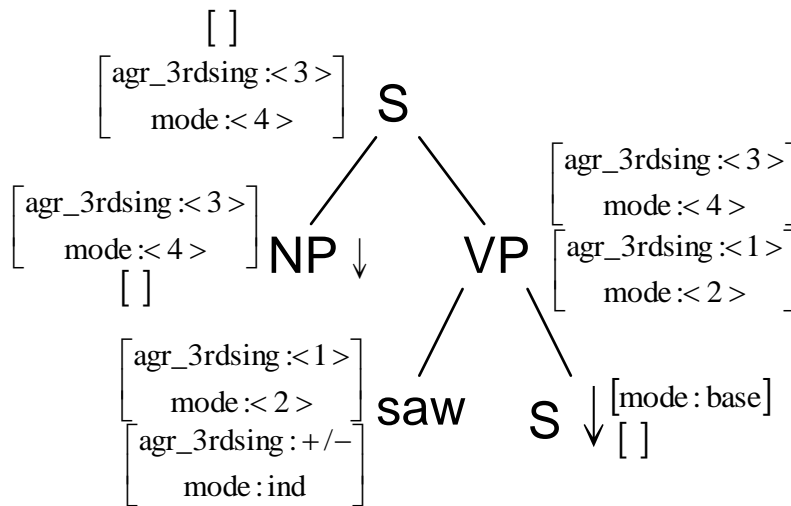


Figure 5.5: Elementary tree for “saw”

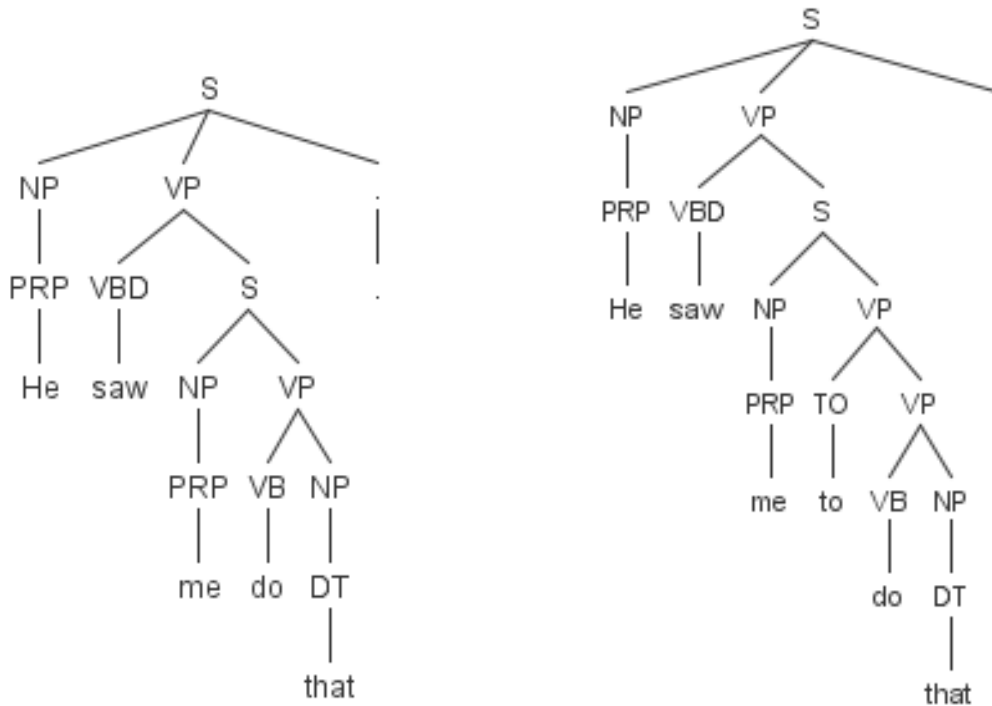


Figure 5.6(a). Grammatical sentence of “saw”

(b) Ungrammatical sentence of “saw”

But if we look at the simplified elementary tree of “asked” shown in Figure 5.7, we can find that “asked” can only be followed by a sentence with an infinitive sentence (inf). For example, “He asked me to do that” shown in Figure 5.8(a) is a grammatical sentence while “He asked me do that” shown in Figure 5.8(b) is an ungrammatical sentence because “asked” is not allowed to be followed by a sentence with a base verb.

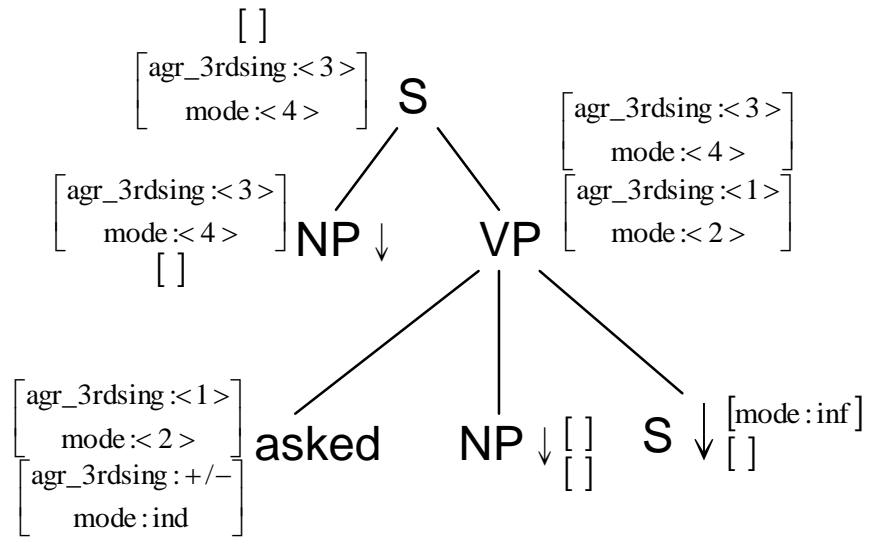


Figure 5.7: Elementary tree for “ask”

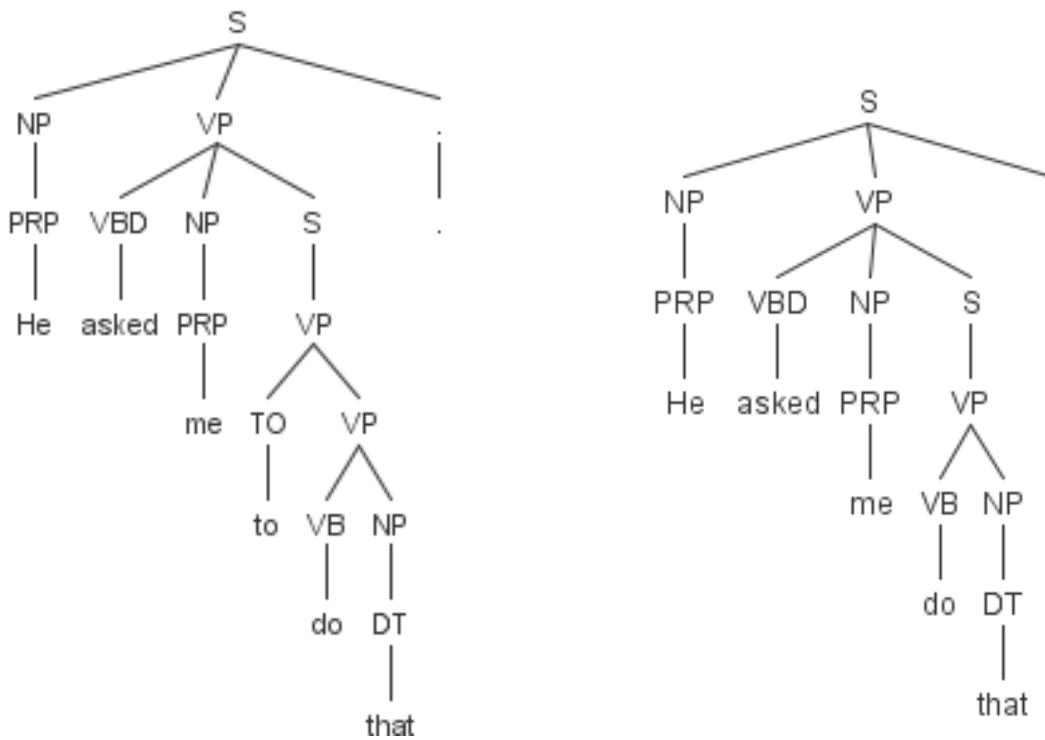


Figure 5.8(a). Grammatical sentence of “ask”

(b) Ungrammatical sentence of “ask”

5.3.2 Syntactic Error Detection

Our procedure for syntactic error detection includes 1. decomposing each sentence hypothesis parse tree into elementary trees, 2. associating each elementary tree with AVMs through look-up in the XTAG grammar, and 3. reconstructing the original parse tree out of the elementary trees using substitution and adjunction operations along with AVM unifications.

When unification of the AVMs fails, a grammatical error has been detected and its error type is also identified by the corresponding feature in the AVM. In order to simultaneously detect multiple error types and their corresponding words, we adjust the traditional unification definition to allow the unification procedure to continue after an AVM failure occurs and also propagate the failure information to relevant words. We call the modified unification fail propagation unification. Each step is illustrated in this section.

5.3.2.1 Decomposing to Elementary trees

Given a translation sentence, we first get its syntactic parse using the Stanford parser (Klein & Manning, 2003) and then decompose the parse to multiple elementary trees by using an elementary tree extractor, a modification of (Chen & Vijay-Shanker, 2000). After that, each lexical item in the sentence will be assigned one elementary tree. Taking the sentence – “Many young student play basketball” as an example, its parse and extracted elementary trees are shown in Figure 5.9 and Figure 5.10, respectively. In Figure 5.9, the arrows represent relations among the elementary trees and the relations are either substitution or adjunction. In this example, the two upper arrows are substitutions and the two bottom arrows are adjunctions.

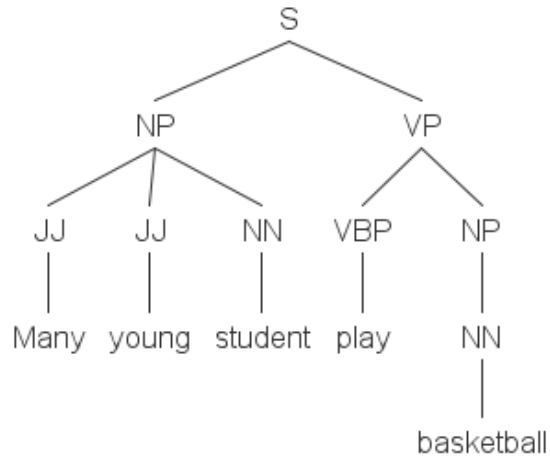


Figure 5.9: Parse of “Many young student play basketball”

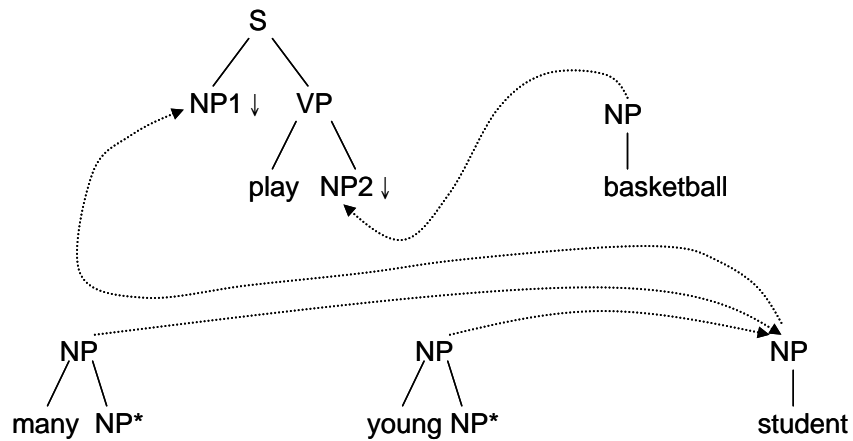


Figure 5.10: The elementary trees of ‘Many young student play basketball’ and their relations

5.3.2.2 Associating AVMs to Elementary trees

Each elementary tree is associated with AVMs through look-up in the XTAG English grammar. Using the same example of the sentence – “Many young student play basketball”, its elementary trees, relations and one set of AVMs (simplified version) are shown in Figure 5.11. To keep tracing what word(s) that a feature value relates to for the next step of reconstruction, we design a new data structure of word set, named “word trace”. It is represented by “{...}” and attached

with each feature value except the value of “null”, such as “agr_num:pl{play}” in Figure 5.11.

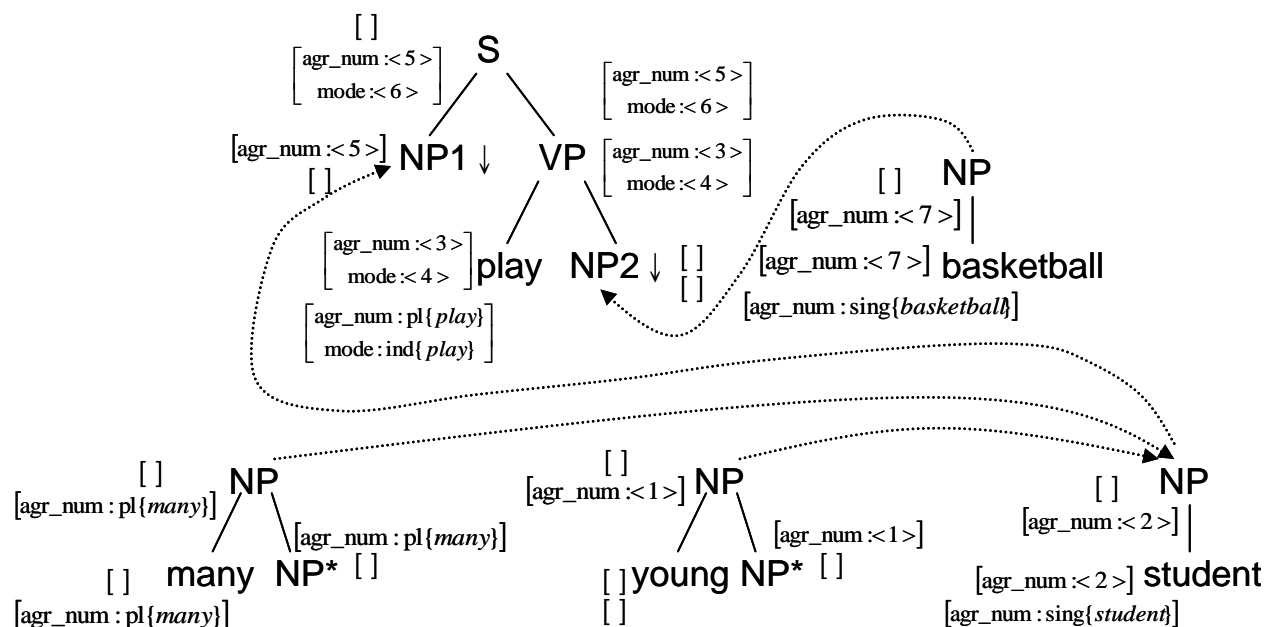


Figure 5.11: The elementary trees of ‘Many young student play basketball’, their relations and AVMs (simplified version).

In XTAG English Grammar, sometimes one elementary tree could have multiple possible AVM associations. For example, for the verb “are”, one of its elementary trees is associated with three different AVMs, one for 2nd person singular, one for 2nd person plural, and one for 3rd person plural. Unless we can reference the context for “are” (e.g., its subject), we are not sure which AVM should be used in the reconstruction. So we associate each elementary tree with its all possible AVMs defined in the XTAG English Grammar.

5.3.2.3 Reconstruction Framework

Once the elementary trees are associated with AVMs, they will be used to reconstruct the original parse tree through substitution and adjunction operations which are indicated during the process of decomposing a parse tree to elementary trees. The reconstruction process is able to decide if there is any conflict with the AVMs values. When a conflict occurs, it will cause an AVM unification failure, associated with a certain grammatical error.

5.3.2.4 Fail Propagation Unification

Our system detects grammatical errors by identifying unification failures. However, traditional unification does not define how to proceed after failures occur, and also lacks an appropriate structure to record error traces. So we extend it as follows:

$$[f=x] \{t1\} \quad U \quad [f=x] \{t2\} \quad \Rightarrow \quad [f=x] \{t1\} \text{ union } \{t2\} \quad (1)$$

$$[f=x] \{t1\} \quad U \quad [f=null] \quad \Rightarrow \quad [f=x] \{t1\} \quad (2)$$

$$[f=null] \quad U \quad [f=null] \quad \Rightarrow \quad [f=null] \quad (3)$$

$$[f=x] \{t1\} \quad U \quad [f=y] \{t2\} \quad \Rightarrow \quad [f=fail] \{t1\} \text{ union } \{t2\} \quad (4)$$

$$[f=fail] \{t1\} \quad U \quad [f=null] \quad \Rightarrow \quad [f=fail] \{t1\} \quad (5)$$

$$[f=fail] \{t1\} \quad U \quad [f=y] \{t2\} \quad \Rightarrow \quad [f=fail] \{t1\} \text{ union } \{t2\} \quad (6)$$

$$[f=fail] \{t1\} \quad U \quad [f=fail] \{t2\} \quad \Rightarrow \quad [f=fail] \{t1\} \text{ union } \{t2\} \quad (7)$$

Where f is a feature type, such as “arg_num”; x and y are two different feature values; U represents the “unify” operation; $t1$ and $t2$ are word traces introduced in Section 5.3.2.2. “fail” is also a value.

(1)~(4) are traditional unification operations except that these operations are along with their

word traces' union operations. When a unification failure occurs in (4), the unification procedure does not halt but only assigns f a value of “fail” and proceeds. (5)~(7) propagate the value of “fail” to the related words' AVMs. Take the sentence of Figure 5.11 as an example, the following two fail propagation unifications occur in order during the reconstruction:

[arg_num=pl]{many} U [arg_num=sing]{student} => [arg_num =fail]{many,student}
 [arg_num=fail]{many, student} U [arg_num=pl]{play} => [arg_num =fail]{many,student,play}

After the two fail propagation unifications, we identify that there is an agr_num error related to three words – “many”, “student” and “play” by the feature value of “fail” and the word trace of “{many,student,play}”.

After going through the entire reconstruction procedure, the reconstructed parse tree with AVMs is shown in Figure 5.12.

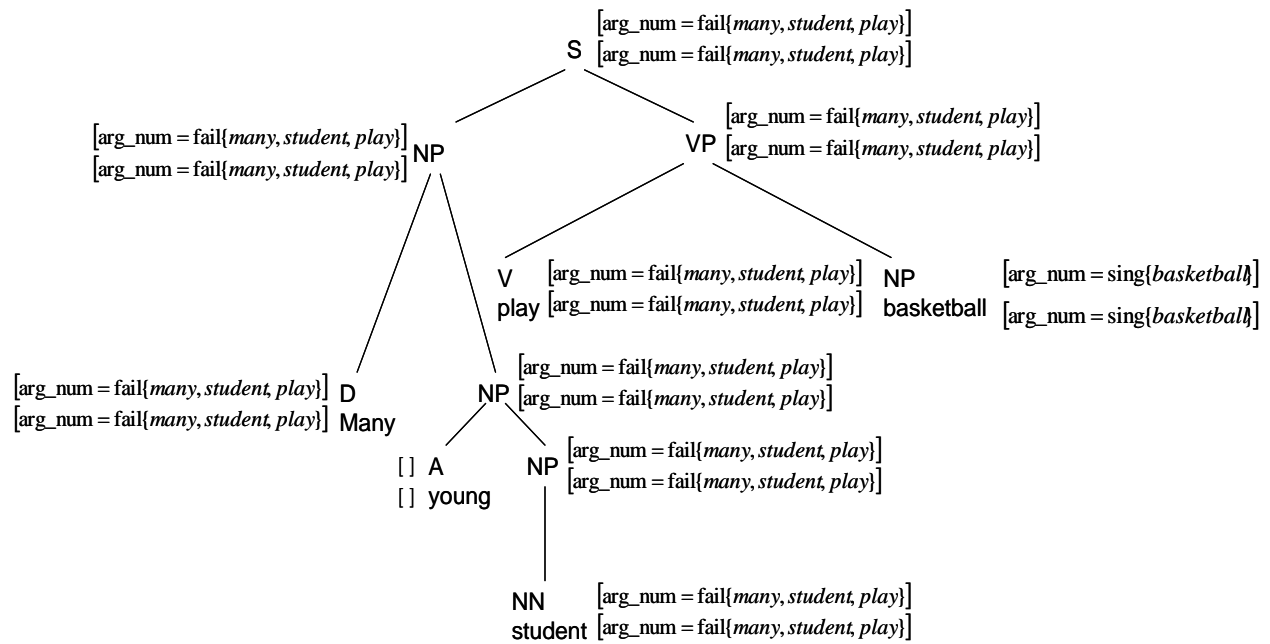


Figure 5.12: The reconstructed parse tree with AVMs of the sentence- “Many young student play basketball”

5.3.3 Syntactic Error Correction

Because in our experimental datasets, only around 10% translations are detected to have syntactic errors, it is not practical to apply the detected results as a general feature in the log-linear model of sentence-level combination. So in this section, our goal is to correct the detected translations.

When an AVM has the value of “fail”, its word trace must contain at least one ungrammatical word. The two following questions need to be answered: which words in the word trace should be corrected and how should they be corrected? To date, we have developed the following simple mechanism to correct words with the agreement problem: first, within the word trace, the words whose original feature value is in the minority compared with other words’ original feature value is decided to be corrected. We call this *feature-value voting*. Take the word trace of “{many,student,play}” in Figure 5.12 as an example, “student” should be corrected since its *agr_num* is “sing” and the other two words’ *agr_num* is “plural”.

Once the corrected words are selected, we replace them with their variations which original feature value is in the majority. For example, we replace the above “student” with “students”.

5.3.4 Experiment

Among the 57 major elementary trees and 50 feature types that XTAG defines, we have implemented 26 major elementary trees and 4 feature types – *agr_pers*, *arg_num*, *arg_3rdsing* and several cases of mode/mood at this point (The first three belong to agreement features.)

We use the same setting as in Section 3.2.4.1. For the reader’s convenience, we describe it here again:

GALE Chi-Eng Dataset: The GALE Chi-Eng Dataset consists of source sentences, corresponding machine translations of 12 MT systems and four human reference translations. It

also provides word alignments between source and translation sentences. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 422 sentences and the test set also includes 422 sentences. Among the five systems, “rwth-pbt-sh” performs the best in BLEU, and since we are tuning toward BLEU, we regard “rwth-pbt-sh” as the top MT system.

NIST Chi-Eng Dataset: The NIST Chi-Eng Dataset also consists of source sentences in Chinese, corresponding machine translations of multiple MT systems and four human reference translations in English, but word alignments between source and translation sentences are not included. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 524 sentences and the test set includes 788 sentences. Among the five systems, “Sys 03” performs the best in BLEU, and since we are tuning toward BLEU, we regard “Sys 03” as the top MT system.

We use our syntactic error detector to detect grammatical errors of a given translation. And we design a binary grammatical indicator as follows: once there is at least one error, the indicator is set to 1; otherwise, it is set to 0. In our log linear, we use this indicator along with TER-based consensus and Gigaword-trained 3-gram LM to select the best translation among all MT systems’ translations. Table 5.4 shows the results of the NIST Chi-Eng Dataset.

	Bleu	TER	METEOR
Sys 03	30.16	55.45	54.43
LM+consensus (baseline)	30.89	55.03	55.24
LM+consensus + SDLM	31.61	54.78	55.68
LM+consensus + SyntacticErrorDetection	31.41	55.03	55.62

Table 5.4: Result of sentence-level translation combination using Syntactic Error Detection on NIST Chi-Eng Dataset

From Table 5.4, we see that using syntactic error detection along with LM and consensus outperforms just using LM and consensus, which shows the effective of syntactic error detection. And we also see that the effective of syntactic error detection does not exceed SDLM.

The results of syntactic error detection for agreement and mode errors and correction for agreement errors on GALE Chi-Eng Dataset are shown in Table 5.5.

	Detected sentences (arg error + mode error)	Corrected sentences (arg error)	Bleu for corrected sentences (before)	Bleu for corrected sentences (after)
Sys nrc	23	9	26.75	27.80
Sys rwth-pbt-aml	18	7	32.13	32.67
Sys rwth-pbt-jx	25	14	31.49	32.17
Sys rwth-pbt-sh	30	11	29.31	30.61
Sys sri-hpbt	18	8	29.15	28.83

Table 5.5: The results of syntactic error detection and correction for GALE Chi-Eng Dataset

From Table 5.5, we see that the overall Bleu score for all sentences is not significantly improved. But if we take a close look at just the sentences where agreement errors were corrected and calculate their Bleu scores, we can see that the corrected translations are improved for every system except for “Sys sri-hpbt”, which shows the effectiveness and potential of our approach.

5.4 Argument Alignment

We hypothesize that for a good translation, the predicate-argument structures are retained in order to preserve the semantics, i.e., predicate-argument structures and argument types in source and target should be the same in most cases. For example, an agent for a predicate in source tends to also be an agent for that predicate in target. The hypothesis can be supported by the investigation of (Wu and Palmer 2011), who obtain argument alignments of PropBank, such as examples of Figure 5.13, using their argument aligner, and calculated the frequencies of different argument alignment type of PropBank, shown in Table 5.6.

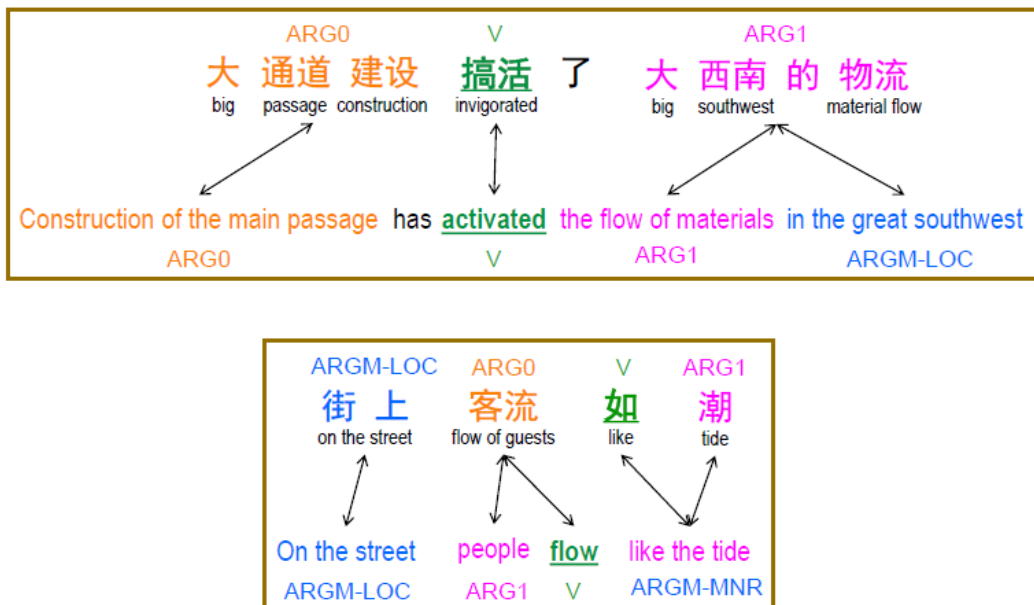


Figure 5.13: Examples of alignments between Chinese arguments and English argument

arg type	A0	A1	A2	A3	A4	ADV	BNF	DIR	DIS	EXT	LOC	MNR	PRP	TMP	TPC	V
A0	1610	79	25	0	0	28	1	0	0	0	8	5	1	11	1	9
A1	432	2665	128	11	0	83	9	12	0	0	29	12	5	21	3	142
A2	43	310	140	8	3	55	6	9	0	2	20	10	1	4	1	67
A3	2	14	21	7	0	2	4	2	0	0	1	2	1	0	1	4
A4	1	37	9	3	6	0	0	0	0	0	1	0	1	0	0	4
ADV	33	36	9	6	0	307	2	5	6	0	44	121	6	11	2	19
CAU	1	0	0	0	0	1	0	0	0	0	0	0	16	0	0	1
DIR	1	13	3	2	0	1	0	3	0	0	3	0	0	0	0	20
DIS	2	0	0	0	0	69	0	0	40	0	2	1	3	3	0	0
EXT	0	4	0	0	0	26	0	0	0	0	0	0	0	0	0	2
LOC	23	65	13	1	0	3	1	0	0	0	162	0	0	5	0	4
MNR	9	9	5	0	0	260	0	0	0	1	3	34	0	0	0	25
MOD	1	0	0	0	0	159	0	0	0	0	0	0	0	0	0	84
NEG	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	5
PNC	3	23	11	4	0	1	6	1	0	0	1	2	35	2	0	8
PRD	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	1
TMP	14	21	2	0	0	235	0	3	0	1	8	16	0	647	0	6
V	25	28	22	1	0	211	1	0	1	0	2	12	0	0	0	3278

Table 5.6: Argument Alignment Mapping Table for PropBank. For example, the cell of “1610” represents that the frequency of A0 in source and A0 in target is 1610.

In addition to this overall mapping, given an argument type and its predicate of a source sentence, Wu and Palmer (2011) also calculated the probabilities of its aligned argument types of the target sentence. Table 5.7 shows a very small part of these conditional probabilities. The first row means the probabilities of the aligned argument types of the target sentence given argument type – “A0” and its predicate - ”接受” of a source sentence. The second row means the probabilities of the aligned argument types of the target sentence given argument type – “A1” and its predicate - ”接受” of a source sentence.

	A0 in target	A1 in target	A2 in target	ADV in target	TMP in target	LOC in target
A0 of predicate - “接受” in source	0.8274	0.0952	-	0.0327	0.0119	0.0104
A1 of predicate - “接受” in source	0.0352	0.8816	0.0296	-	-	-

Table 5.7: The probabilities of the aligned argument types of the target sentence given an argument type and its predicate of a source sentence. For example, the cell of “0.8274” represents that $P(\text{A0 in target} | \text{A0 of predicate - “接受” in source}) = 0.8274$.

5.4.1 Approach

Given a source sentence, a target sentence, the word alignment between the two and the semantic role labelers for source and target sides, we can obtain the argument alignment using an argument aligner and then exploit these argument alignment probabilities learned from PropBank to evaluate the quality of the target sentence.

Our notations are described as follows:

$pred_i^s$: the i th predicate of a source sentence

$arg_{i,j}^s$: the j th argument type of $pred_i^s$

$arg_{i,j,k}^t$: the k th aligned argument of $arg_{i,j}^s$

$P(arg_{i,j,k}^t | arg_{i,j}^s, pred_i^s)$: the probability of the aligned argument type - $arg_{i,j,k}^t$ of the target sentence given an argument type - $arg_{i,j}^s$ and its predicate - $pred_i^s$ of a source sentence

We evaluate the quality of a given translation by the following formula of its *Score*, which is either used as an only measure to select the best translation or as one feature in our log-linear model, along with other features, to select the best translation.

$$Score = \sum_i \sum_j \sum_k P(\arg_{i,j,k}^t \mid \arg_{i,j}^s, pred_i^s) \quad (5.2)$$

In formula (5.2), the score for evaluating the quality of a given translation sentence is the sum of the probability of every aligned argument type of the target sentence given every argument type and its predicate of the source sentence.

5.4.2 Experiment

We use the same setting of GALE Chi-Eng Dataset as in Section 3.2.4.1. For the reader’s convenience, we describe it here again: the GALE Chi-Eng Dataset consists of source sentences in Chinese, corresponding machine translations of 12 MT systems and four human reference translations in English. It also provides word alignments between source and translation sentences. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 422 sentences and the test set also includes 422 sentences. Among the five systems, “rwth-pbt-sh” performs the best in BLEU, and since we are tuning toward BLEU, we regard “rwth-pbt-sh” as the top MT system.

5.4.2.1 Results

	Bleu	TER	METEOR
Sys nrc	30.95	59.31	59.06
Sys rwth-pbt-aml	31.83	58.09	58.85
Sys rwth-pbt-jx	31.78	62.04	57.51
Sys rwth-pbt-sh	32.63	58.67	58.98
Sys sri-hpbt	32.00	58.97	58.84
random selection	31.52	59.55	58.55
ArgumentAlignment	32.16	59.18	59.38
LM+consensus (baseline)	32.81	57.22	59.43
LM+consensus+ArgumentAlignment	32.69	57.71	59.33

Table 5.8: Results of using Argument Alignment to select the best translation

From Table 5.8, we see that when only Argument Alignment is used to select the best translation, it is among the top two in comparison with the five MT systems. And it significantly improves just random selection. This shows that Argument Alignment is helpful. However, if we add the language model and consensus along with the Argument Alignment as the features, there is no improvement in comparison with just using the consensus and LM. This observation reveals that Argument Alignment does correspond to the translation quality, but it is not as strong indicator as consensus and LM to evaluate the translation quality.

5.5 Conclusions

In this chapter, we first presented Supertagged Dependency Language Model for explicitly modeling syntactic dependencies of the words of translated sentences in Section 5.2. Its goal is to select the most grammatical translation from candidate translations. To obtain the supertagged dependency structure of a translation candidate, a two-step mechanism based on constituent

parsing and elementary tree extraction is also proposed. SDLM shows its effectiveness in the scenario of translation selection.

In Section 5.3, we also proposed a new FB-LTAG-based syntactic error detection and correction mechanism along with a novel AVM unification method to simultaneously detect multiple ungrammatical types and their corresponding words for machine translation. Our approach features: 1) the use of XTAG grammar, a rule-based grammar developed by linguists, 2) the ability to simultaneously detect multiple ungrammatical types and their corresponding words by using unification of feature structures, and 3) the ability to simultaneously correct multiple ungrammatical types based on the detection information. From the experimental results, we see that using syntactic error detection along with LM and consensus outperforms just using LM and consensus, although its effective does not exceed SDLM. We also demonstrated its utility for correcting agreement errors.

We also applied the probabilities of *argument alignment* between source and target as an indicator to evaluate the quality of the target sentence in Section 5.4. Our experimental results demonstrate that *argument alignment* is not as strong indicator as consensus and LM, but it does correspond to the translation quality.

Chapter 6

Hybrid Combination

In Chapter 3 and 4, we introduced our two phrase-level combination frameworks: one approaches combination via re-decoding the source sentence, and the other one approaches combination via paraphrasing the backbone translation hypothesis. In Chapter 5, we proposed a sentence-level model using novel syntactic and semantic features to select the best hypothesis from a pool of hypothesis candidates. Phrase-level and sentence-level combination have their own distinct advantages: the former is able to generate a whole new fused translation that never appeared in the original translations of multiple MT systems while in the latter it is easier to exploit more sophisticated syntactic and semantic information than in the phrase-level models. So, the design of a hybrid combination structure for the integration of phrase-level and sentence-level combination in order to utilize both advantages is an appealing direction.

Another motivation for a hybrid combination structure is to provide a more diverse set of plausible fused translations to consider. MT researchers have recently started to consider diversity for system combination (Macherey and Och, 2007; Devlin and Matsoukas, 2012, Xiao et al., 2013; Cer et al., 2013; Gimpel et al., 2013). Devlin and Matsoukas (2012) generate diverse

translations according to translation length and number of rules applied. Xiao et al. (2013) used bagging and boosting to get a diverse system. Cer et al. (2013) used multiple identical systems trained jointly with an objective function that encourages the systems to generate complementary translations. Gimpel et al. (2013) propose a dissimilarity function to generating diverse translations in the context of system combination, discriminative reranking and post editing.

For either the re-decoding framework or the paraphrasing framework, the decoding object is but a single object – either the source sentence or a backbone translation. Consider the paraphrasing framework as an example. Although we have shown that the quality of a backbone translation corresponds to the quality of its paraphrased outcome in Section 4.3.4.3, paraphrasing only one single translation could limit the possibility of generating more diverse fused translations. Therefore, our goal is to generate more diverse fused translations through a pipeline-based integration of our phrase-level and sentence-level combination systems. In this section, we propose two hybrid combination structures: the first one is *homogeneously hybrid combination*, where the same phrase-based techniques are used to generate fused translations for the sentence-level combination component to select the best of those, described in Section 6.1, and the other one is *heterogeneously hybrid combination*, where different phrase-based techniques are used to generate outputs for the sentence-level combination component to select the best of those, described in Section 6.2.

6.1 Homogeneously Hybrid Combination

Figure 6.1 shows one homogeneously hybrid combination architecture. The source text is translated by multiple MT systems, and each system produces the top-one translation hypothesis as well as phrase alignments between source and target. No sentence-level models are needed for

backbone selection. Instead, every MT translation has a chance to be the backbone. For each MT translation, we paraphrase it to another translation by fusing it with other MT translations using our *paraphrasing model*. Thus, this hybrid combination architecture considers more combination possibilities. Figure 6.2 shows another homogeneously hybrid combination architecture, where we use *hierarchical paraphrasing model* to provide fused translations.

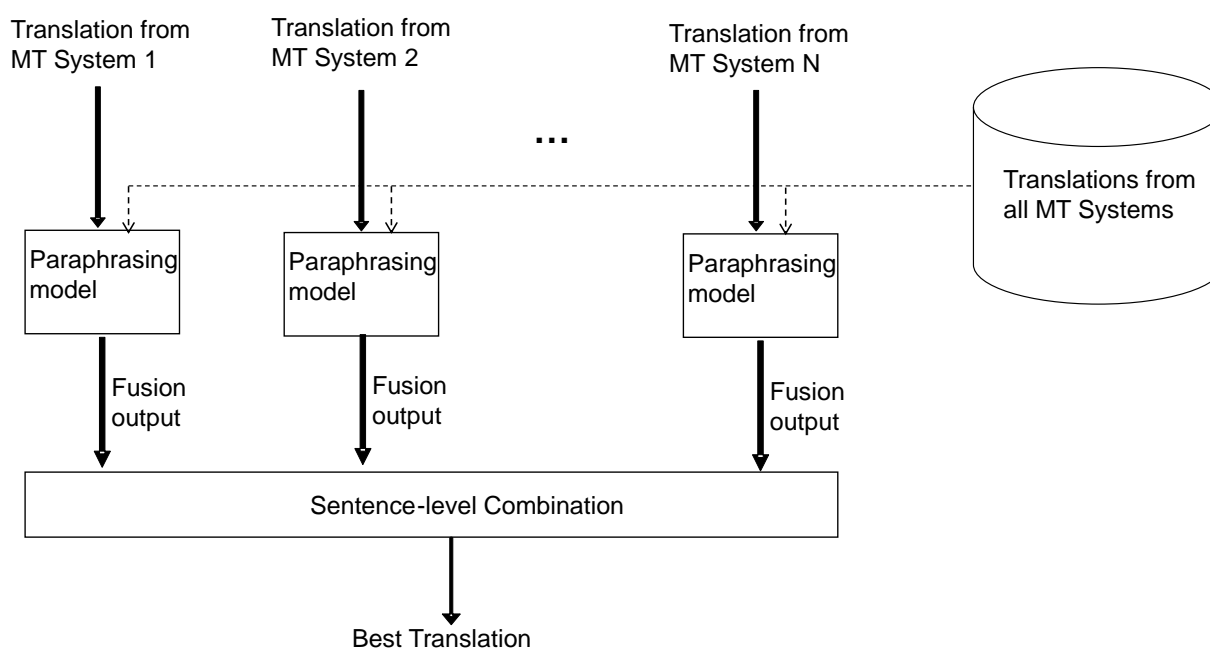


Figure 6.1: Homogeneously Hybrid Paraphrasing Model

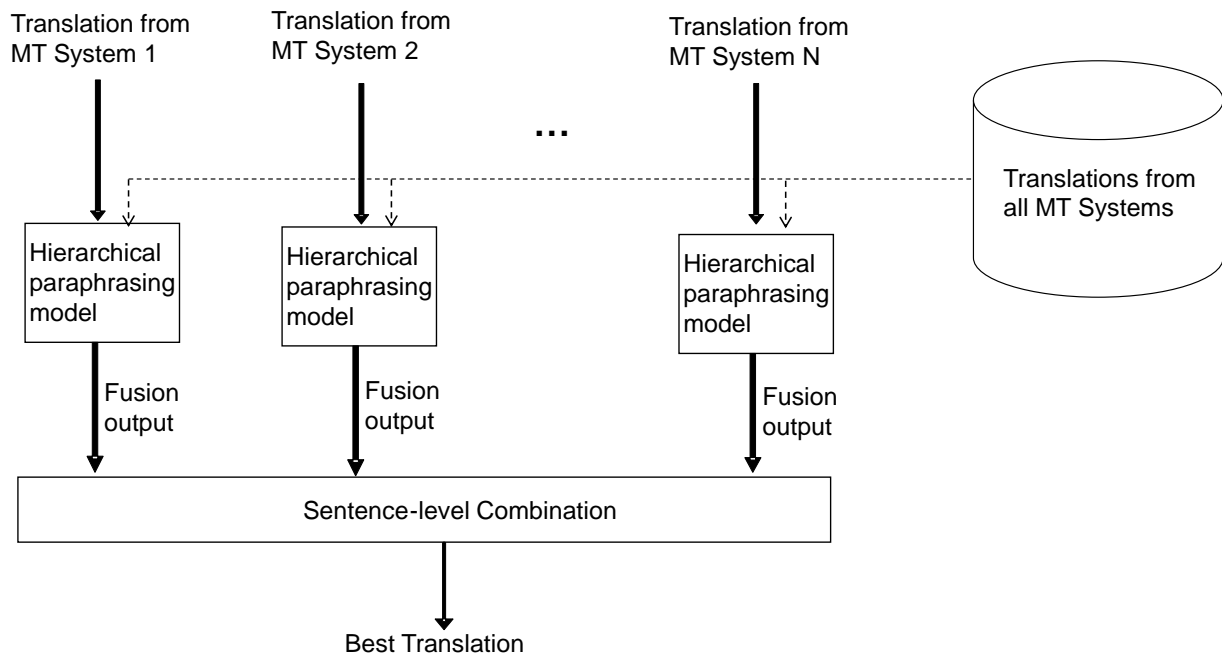


Figure 6.2: Homogeneously Hybrid Hierarchical Paraphrasing Model

6.1.1 Experiment

The experiments are conducted and reported on system translations and references from NIST Chi-Eng Dataset and NIST Ara-Eng Dataset.

6.1.1.1 Setting

We use the same setting of NIST Chi-Eng Dataset as in Section 4.2.4.1. For the reader's convenience, we describe it here again: the NIST Chi-Eng Dataset consists of source sentences in Chinese, corresponding machine translations of multiple MT systems and four human reference translations in English, but word alignments between source and translation sentences are not included. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 524 sentences and the test set includes 788 sentences. Among the five systems, "Sys 03"

performs the best in BLEU, and since we are tuning toward BLEU, we regard “Sys 03” as the top MT system.

We investigate two sets of features for our sentence-level combination. One set includes:

- Sentence consensus toward MT systems’ translations based on TER (consensus)
- Gigaword-trained 3-gram LM (LM)

And other set includes

- Sentence consensus toward MT systems’ translations based on TER (consensus)
- Gigaword-trained 3-gram LM (LM)
- Supertag-based dependency language model (SDLM)

6.1.1.2 Results

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>paraphrasing model (selected backbone)</i>	32.65	55.11	56.17
<i>paraphrasing model (Sys 03 as backbone)</i>	32.17	58.15	55.25
<i>paraphrasing model (Sys 15 as backbone)</i>	31.93	55.72	55.51
<i>paraphrasing model (Sys 20 as backbone)</i>	30.66	57.92	53.79
<i>paraphrasing model (Sys 22 as backbone)</i>	31.86	56.02	55.18
<i>paraphrasing model (Sys 31 as backbone)</i>	31.52	55.69	55.52
<i>Homogeneously hybrid paraphrasing model (LM+consensus)</i>	32.64	55.07	55.87
<i>Homogeneously hybrid paraphrasing model (LM+consensus+SDLM)</i>	32.87	55.86	56.21

Table 6.1: The results of Homogeneously Hybrid Paraphrasing Models on NIST Chi-Eng Dataset

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>hierarchical paraphrasing model (selected backbone)</i>	32.59	55.06	56.19
<i>hierarchical paraphrasing model (Sys 03 as backbone)</i>	31.76	55.44	55.25
<i>hierarchical paraphrasing model (Sys 15 as backbone)</i>	31.72	56.17	55.47
<i>hierarchical paraphrasing model (Sys 20 as backbone)</i>	31.00	56.63	54.30
<i>hierarchical paraphrasing model (Sys 22 as backbone)</i>	31.46	56.22	55.10
<i>hierarchical paraphrasing model (Sys 31 as backbone)</i>	31.92	55.56	55.56
<i>Homogeneously hybrid hierarchical paraphrasing model (LM+consensus)</i>	33.14	55.34	56.55
<i>Homogeneously hybrid hierarchical paraphrasing model (LM+consensus+SDLM)</i>	32.52	55.31	56.05

Table 6.2: The results of Homogeneously Hybrid Hierarchical Paraphrasing Models on NIST Chi-Eng Dataset

Table 6.1 shows that, in comparison with the *paraphrasing model (selected backbone)*, the *homogeneously hybrid paraphrasing model* using a feature set of *LM+consensus* does not provide improvement, but when it uses a feature set of *LM+consensus+SDLM*, it gives a little bit of improvement in BLEU and MET. Table 6.2 shows that, in comparison with the *hierarchical paraphrasing model (selected backbone)*, the *homogeneously hybrid hierarchical paraphrasing model* using a feature set of *LM+consensus* provides significant improvement but when it uses a feature set of *LM+consensus+SDLM*, it performs worse. We explain this as follows. Since SDLM aims to calculate the grammaticality of translated sentences to evaluate the quality of translation, it would be expected to be more effective on translation with poor syntactic structures. And because the *hierarchical paraphrasing model* already implicitly considers

syntactic structures via SCFG, SDLM is not able to bring the benefit.

Among all system combination models on NIST Chi-Eng Dataset, described in this thesis, *homogeneously hybrid hierarchical paraphrasing model* using a feature set of *LM+consensus* provides the best performance of Bleu score of “33.14”, which is higher than Bleu score of *Confusion Network* by “1.93” and higher than Bleu score of best MT system by “2.98”.

From Table 6.1 and 6.2, we see that under the same feature set of *LM+consensus*, *homogeneously hybrid paraphrasing model* does not provide improvement, but *homogeneously hybrid hierarchical paraphrasing model* provides significant improvement. That might stem from the hypothesis that *hierarchical paraphrasing model* is able to generate more diverse translations than the *paraphrasing model*, because the former is able to model more possible word re-orderings. To support this hypothesis, we compute TER scores for pairs of outputs of the *paraphrasing model* and compute TER scores for pairs of outputs of the *hierarchical paraphrasing model* in order to compare the diversity degree of the outputs of the two models. The results are shown in Table 6.3 and Table 6.4.

	<i>Para Sys 03</i>	<i>Para Sys 15</i>	<i>Para Sys 20</i>	<i>Para Sys 22</i>	<i>Para Sys 31</i>
<i>Para Sys 03</i>	-	31.503	31.031	29.383	29.068
<i>Para Sys 15</i>	33.676	-	36.503	33.191	33.377
<i>Para Sys 20</i>	32.442	35.689	-	23.974	30.573
<i>Para Sys 22</i>	31.430	33.293	24.506	-	27.484
<i>Para Sys 31</i>	31.021	33.203	31.172	27.383	-
<i>Average</i>	32.142	33.422	30.803	24.483	30.126
<i>Average</i>	30.995				

Table 6.3: TER-based diversity degree of the outputs of *paraphrasing model*

	<i>HiePara</i> <i>Sys 03</i>	<i>HiePara</i> <i>Sys 15</i>	<i>HiePara</i> <i>Sys 20</i>	<i>HiePara</i> <i>Sys 22</i>	<i>HiePara</i> <i>Sys 31</i>
<i>HiePara</i> <i>Sys 03</i>	-	32.945	38.687	33.403	30.232
<i>HiePara</i> <i>Sys 15</i>	31.276	-	32.479	34.227	31.551
<i>HiePara</i> <i>Sys 20</i>	37.353	32.983	-	26.029	31.282
<i>HiePara</i> <i>Sys 22</i>	32.078	34.755	25.993	-	27.092
<i>HiePara</i> <i>Sys 31</i>	29.221	32.010	31.201	27.172	-
<i>Average</i>	32.482	33.173	32.090	30.208	30.039
<i>Average</i>	31.598				

Table 6.4: TER-based diversity degree of the outputs of *hierarchical paraphrasing model*

In Table 6.3 and Table 6.4, we see that the diversity degree of the outputs of the *hierarchical paraphrasing model* (average TER score: 31.593) is higher than the diversity degree of the outputs of the *paraphrasing model* (average TER score: 30.995). Especially for the relatively poor MT systems (“Sys 20” and “Sys 22”), the *hierarchical paraphrasing model* provides much higher diversity for its outputs than the *paraphrasing model*.

In addition to NIST Chi-Eng Dataset, we also carry out the experiments of *homogeneously hybrid combination models* on NIST Ara-Eng Dataset, which plays a role of blind test to provide a more objective evaluation. The results are shown in Table 6.5 and 6.6.

	BLEU	TER	MET
Sys 03	48.40	45.55	70.67
<i>Confusion Network (baseline)</i>	48.56	43.81	70.67
<i>paraphrasing model (selected backbone)</i>	49.33	45.08	70.87
<i>Homogeneously hybrid paraphrasing model (LM+consensus)</i>	50.25	43.55	71.19

Table 6.5: The results of Homogeneously Hybrid Paraphrasing Models on NIST Ara-Eng Dataset

	BLEU	TER	MET
Sys 03	48.40	45.55	70.67
<i>Confusion Network (baseline)</i>	48.56	43.81	70.67
<i>hierarchical paraphrasing model (selected backbone)</i>	49.46	44.84	70.99
<i>homogeneously hybrid hierarchical paraphrasing model (LM+consensus)</i>	50.09	43.71	71.30

Table 6.6: The results of Homogeneously Hybrid Paraphrasing Models on NIST Ara-Eng Dataset

Table 6.5 shows that, in comparison with the *paraphrasing model (selected backbone)*, the *homogeneously hybrid paraphrasing model* using a feature set of *LM+consensus* provides significant improvement. Similarly, Table 6.6 shows that, in comparison with the *hierarchical paraphrasing model (selected backbone)*, the *homogeneously hybrid hierarchical paraphrasing model* using a feature set of *LM+consensus* yields significant improvement. These results demonstrate the *homogeneously hybrid combination model*'s robustness and consistency.

6.2 Heterogeneously Hybrid Combination

In last section, we introduced the *homogeneously hybrid combination*, where the same phrase-based technique is used to generate fused translations for the sentence-level combination component to select. In this section, we introduce the *heterogeneously hybrid combination*,

where different phrase-based techniques are used to generate outputs for the sentence-level combination component to select, shown in Figure 6.3.

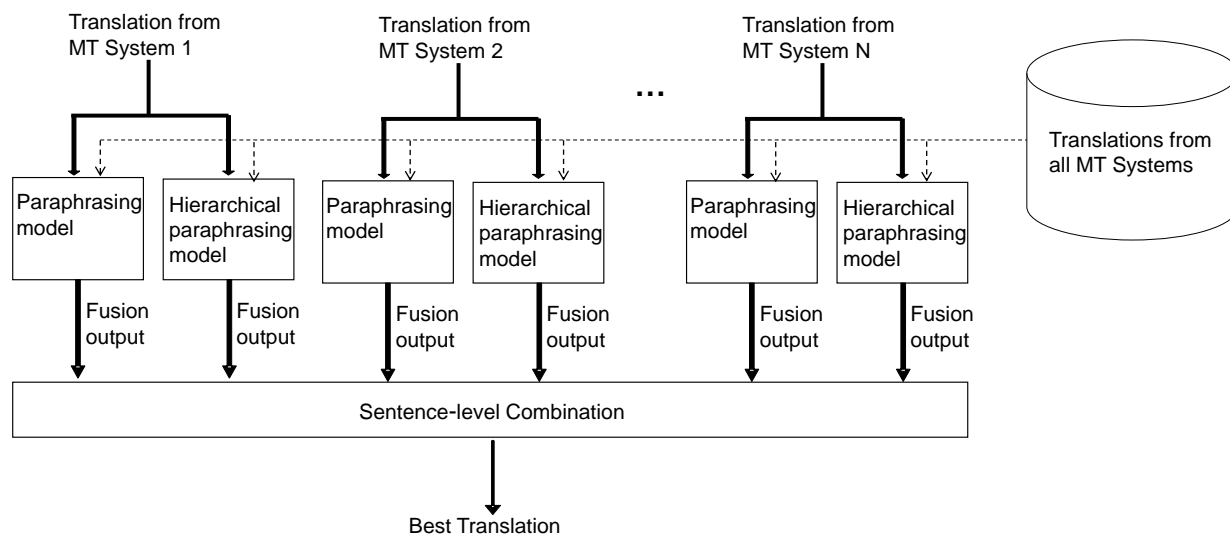


Figure 6.3: Heterogeneously Hybrid Combination

6.2.1 Experiment

The experiments are conducted and reported on NIST Chi-Eng Dataset and NIST Ara-Eng Dataset.

6.2.1.1 Setting

We use the same setting of NIST Chi-Eng Dataset as in Section 4.2.4.1. We manually select the top five MT systems for our combination experiment. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 524 sentences and the test set includes 788 sentences. Among the five systems, “Sys 03” performs the best in BLEU, and since we are tuning toward BLEU, we regard “Sys 03” as the top MT system. Besides NIST Chi-Eng Dataset, we also carried out experiments on NIST Ara-Eng Dataset. Each system provides the top one translation hypothesis for every sentence. The tuning set includes 592 sentences and the test set includes 717 sentences. Among the five systems, “Sys 31” performs the best in BLEU,

and since we are tuning toward BLEU, we regard “Sys 31” as the top MT system.

We investigate two sets of features for our sentence-level combination. One set includes:

- Sentence consensus toward MT systems’ translations based on TER (consensus)
- Gigaword-trained 3-gram LM (LM)

And other set includes

- Sentence consensus toward MT systems’ translations based on TER (consensus)
- Gigaword-trained 3-gram LM (LM)
- Supertag-based dependency language model (SDLM)

6.2.1.2 Results

	BLEU	TER	MET
Sys 03	30.16	55.45	54.43
<i>Confusion Network (baseline)</i>	31.21	54.59	55.59
<i>paraphrasing model (selected backbone)</i>	32.65	55.11	56.17
<i>hierarchical paraphrasing model (selected backbone)</i>	32.59	55.06	56.19
<i>homogeneously hybrid paraphrasing model (LM+consensus)</i>	32.64	55.07	55.87
<i>homogeneously hybrid hierarchical paraphrasing model (LM+consensus)</i>	33.14	55.34	56.55
<i>heterogeneously hybrid combination model (LM+consensus)</i>	32.82	55.52	56.66
<i>homogeneously hybrid paraphrasing model (LM+consensus+SDLM)</i>	32.87	55.86	56.21
<i>homogeneously hybrid hierarchical paraphrasing model (LM+consensus+SDLM)</i>	32.52	55.31	56.05
<i>heterogeneously hybrid combination model (LM+consensus+SDLM)</i>	32.91	55.58	56.04

Table 6.7: The results of Heterogeneously Hybrid Combination Models on NIST Chi-Eng

Dataset

Table 6.7 shows that for either feature set, the *heterogeneously hybrid combination model* outperforms both *paraphrasing model (selected backbone)* and *hierarchical paraphrasing model (selected backbone)* in BLEU, which shows the effective of the *heterogeneously hybrid combination*.

In comparison of *homogeneously hybrid combination models*, for the feature set of *LM+consensus*, the performance in BLEU of the *heterogeneously hybrid combination model* is in the middle of the performance of the two *homogeneously hybrid combination models*. And for the feature set of *LM+consensus+SDLM*, the performance in BLEU of the *heterogeneously hybrid combination model* slightly outperforms both *homogeneously hybrid combination models*.

In addition to NIST Chi-Eng Dataset, we also carry out the experiments of *heterogeneously hybrid combination model* on NIST Ara-Eng Dataset, which plays a role of blind test to provide a more objective evaluation. The results are shown in Table 6.8.

	BLEU	TER	MET
Sys 31	48.40	45.55	70.67
<i>Confusion Network (baseline)</i>	48.56	43.81	70.67
<i>paraphrasing model (selected backbone)</i>	49.33	45.08	70.87
<i>hierarchical paraphrasing model (selected backbone)</i>	49.46	44.84	70.99
<i>homogeneously hybrid paraphrasing model (LM+consensus)</i>	50.25	43.55	71.19
<i>homogeneously hybrid hierarchical paraphrasing model (LM+consensus)</i>	50.09	43.71	71.30
<i>heterogeneously hybrid combination model (LM+consensus)</i>	50.05	44.27	70.87

Table 6.8: The results of Heterogeneously Hybrid Combination Models on NIST Ara-Eng Dataset

Table 6.8 shows that, for the feature set of *LM+consensus*, the *heterogeneously hybrid combination model* still outperforms both *paraphrasing model (selected backbone)* and *hierarchical paraphrasing model (selected backbone)* in BLEU, which demonstrates the *heterogeneously hybrid combination model*'s robustness and consistency.

6.3 Conclusions

In this chapter, we proposed two hybrid combination structures for the integration of phrase-level and sentence-level combination frameworks in order to utilize the advantages of both frameworks and provide a more diverse set of plausible fused translations to consider. The first one is the *homogeneously hybrid combination*, where the same phrase-based techniques are used to generate outputs for the sentence-level combination component to select, and the other one is *heterogeneously hybrid combination*, where different phrase-based techniques are used to generate outputs for the sentence-level combination component to select. Our experiments show that both hybrid combination structures are effective, and the improvement corresponds to the diversity degree of fused translations that our phrase-level combination models provided.

Among all system combination models on NIST Chi-Eng Dataset, described in this thesis, *homogeneously hybrid hierarchical paraphrasing model* using a feature set of *LM+consensus* provides the best performance of Bleu score of “33.14”, which is higher than Bleu score of *Confusion Network* by “1.93” and higher than Bleu score of best MT system by “2.98”.

And all system combination models on NIST Ara-Eng Dataset, described in this thesis, the *homogeneously hybrid paraphrasing model* using a feature set of *LM+consensus* provides the best performance of Bleu score of “50.25”, which is higher than Bleu score of *Confusion Network* by “1.69” and higher than Bleu score of best MT system by “1.85”.

Chapter 7

Conclusions

Given the wide range of successful statistical MT approaches that have emerged recently, it would be beneficial to take advantage of their individual strengths and avoid their individual weaknesses. Multi-Engine Machine Translation attempts to do so by either fusing the output of multiple translation engines or selecting the best translation among them, aiming to improve the overall translation quality. The word-level fusion framework, such as the *confusion network decoding model*, is the most popular approach. However, using a word as the unit of fusion rather than a phrase, has a higher risk of breaking coherence and consistency between the words in a phrase and it is difficult to consider syntax and semantics.

In this thesis, we showed how to use the phrase or the sentence as our combination unit instead of the word; three new phrase-level models, three novel features for the sentence-level model and two novel pipeline-based hybrid combination structures were presented and evaluated.

7.1 Overview of Contributions

Phrase-level Combination: The goal is to fuse the given multiple MT systems' translations. We presented three different novel models to achieve this task.

- *hierarchical phrase-based re-decoding model*
 - It utilizes hierarchical phrases learned from source sentences and target translation hypotheses to re-decode the source sentences using the hierarchical phrases
- *paraphrasing model*
 - It views combination as a paraphrasing process based on a set of paraphrases, learned from monolingual word alignments between a selected best translation hypothesis and other hypotheses.
- *hierarchical paraphrasing model*
 - It views combination as a paraphrasing process with the use of a set of hierarchical paraphrases, learned from monolingual word alignments between a selected best translation hypothesis and other hypotheses.

Among the three phrase-level models, the *paraphrasing model* and the *hierarchical paraphrasing model* have similar performances, and both of them outperform the *hierarchical phrase-based re-decoding model* as well as baseline combination systems.

From our investigational experiments, we also saw that the addition of simple syntactic constraints in both models did not yield improvement. Moreover, we found out that if a given hypothesis for paraphrasing is well translated, the *hierarchical paraphrasing model* would not bring benefits to *paraphrasing model*. But, on the other hand, if a given hypothesis for paraphrasing is poorly translated, *the hierarchical paraphrasing model* is more likely to improve that translation than the *paraphrasing model*.

We also found that the performance of combination strongly correlates with the individual quality of each MT system. For MT combination, the selection of top N MT systems is a reasonable strategy, but larger N does not always bring benefits when N exceeds 5.

Sentence-level Combination: The goal is to select the best translation from the given multiple MT systems' translations. We presented three different novel features to help evaluate the quality of a given translation.

- *Supertagged Dependency Language Model (SDLM)*
 - It explicitly models syntactic dependencies of the words of translated sentences. To obtain the supertagged dependency structure of a translation candidate, a two-step mechanism based on constituent parsing and elementary tree extraction is also presented.
- *FB-LTAG-based syntactic error detector*
 - It uses XTAG grammar, a rule-based FB-LTAG developed by linguists. Our detector is able to simultaneously detect multiple ungrammatical types and their corresponding words by using a novel unification method, and we also show that it can be used to correct ungrammatical words.
- *argument alignment*
 - We applied the probabilities of *argument alignment* between source and target as an indicator to evaluate the quality of the target sentence from a semantic perspective.

Among the three features, *SDLM* is the most effective and *FB-LTAG syntactic error detector* is the second. Although *argument alignment* is not as strong indicator as *SDLM* and *FB-LTAG*

syntactic error detector, our experimental results demonstrate that *argument alignment* does correspond to the translation quality.

Hybrid Combination: the goal is to utilize the advantages of phrase-level and sentence-level combination and provide a more diverse set of plausible fused translations to consider. We presented two novel pipeline-based hybrid combination structures to achieve this task.

- *homogeneously hybrid combination*
 - The same phrase-based technique is used to generate outputs for the sentence-level combination component to select. According to different phrase-based techniques, we developed two structures:
 - *homogeneously hybrid paraphrasing model*
 - *homogeneously hybrid hierarchical paraphrasing model*
- *heterogeneously hybrid combination*
 - Different phrase-based techniques are used to generate outputs for the sentence-level combination component to select.

We found that both hybrid combination structures are effective, and the improvement corresponds to the degree of diversity of fused translations that our phrase-level combination models provided. The *homogeneously hybrid hierarchical paraphrasing model* using a feature set of *LM+consensus* provides the best performance.

Comparison of all models: we list the performances of the best models for *Phrase-level combination*, *Sentence-level combination* and *hybrid combination* on NIST Chi-Eng Dataset.

		BLEU	TER	MET
The Best MT system	Sys 03	30.16	55.45	54.43
<i>Word-level combination (baseline)</i>	<i>Confusion Network</i>	31.21	54.59	55.59
<i>Phrase-level combination</i>	<i>paraphrasing model (selected backbone)</i>	32.65	55.11	56.17
	<i>hierarchical paraphrasing model (selected backbone)</i>	32.59	55.06	56.19
<i>Sentence-level combination</i>	LM + consensus + SDLM	31.61	54.78	55.68
	LM + consensus + SyntacticErrorDetection	31.41	55.03	55.62
<i>hybrid combination</i>	<i>homogeneously hybrid hierarchical paraphrasing model (LM+consensus)</i>	33.14	55.34	56.55
	<i>heterogeneously hybrid combination model (LM+consensus+SDLM)</i>	32.91	55.58	56.04

Table 7.1: The performances of the best models for *Phrase-level combination*, *Sentence-level combination* and *hybrid combination* on NIST Chi-Eng Dataset

From Table 7.1, we see that *homogeneously hybrid hierarchical paraphrasing model* using a feature set of *LM+consensus* provides the best performance on NIST Chi-Eng Dataset through this thesis : its Bleu score of “33.14” is higher than the Bleu score of *Confusion Network* by “1.93” and higher than Bleu score of best MT system by “2.98”.

In addition to the NIST Chi-Eng Dataset, we also carried out experiments of our models on NIST Ara-Eng Dataset, which plays a role of blind test to provide a more objective evaluation.

		BLEU	TER	MET
The Best MT system	Sys 31	48.40	45.55	70.67
<i>Word-level combination (baseline)</i>	<i>Confusion Network</i>	48.56	43.81	70.67
<i>Phrase-level combination</i>	<i>paraphrasing model (selected backbone)</i>	49.33	45.08	70.87
	<i>hierarchical paraphrasing model (selected backbone)</i>	49.46	44.84	70.99
<i>hybrid combination</i>	<i>homogeneously hybrid hierarchical paraphrasing model (LM+consensus)</i>	50.25	43.55	71.19
	<i>heterogeneously hybrid combination model (LM+consensus)</i>	50.05	44.27	70.87

Table 7.2: The performances of the combination models on NIST Ara-Eng Dataset

From Table 7.2, we saw that the *homogeneously hybrid paraphrasing model* using a feature set of *LM+consensus* provides the best performance on the NIST Ara-Eng Dataset through this thesis: its Bleu score of “50.25” is higher than Bleu score of *Confusion Network* by “1.69” and higher than Bleu score of best MT system by “1.85”. This result demonstrates the *hybrid combination model’s* robustness and consistency. It shows the results are consistent across test sets and across two languages.

The reason why the *hybrid combination models* consistently provide the best performances could stem from the fact that the hybrid combination structures integrate phrase-level and sentence-level combination approaches, which fully utilize the individual advantages of the two

frameworks: phrase-level approaches are able to generate a whole new fused translation that never appeared in the original translations of multiple MT systems while sentence-level combination approaches make the final decisions using information from whole sentences. Another reason to interpret *hybrid combination models'* excelled performance is that they consider a more diverse set of plausible fused translations.

7.2 Future Work

For phrase-level combination models, the integration of grammatical knowledge, such as SDLM and XTAG English Grammar, would be an appealing future research direction. Since SDLM and XTAG English Grammar are represented in the form of *tree adjoining grammar*, it is natural to utilize a *synchronous tree adjoining grammar* (STAG) as our phrase-level combination model to integrate the grammatical knowledge in the form of *tree adjoining grammar*. The *synchronous tree adjoining grammar* could be learned either from bilingual word alignments between source sentences and target translation hypotheses, or from monolingual word alignments between a selected best translation hypothesis and other hypotheses. Semantic information, such as argument types, can also be attached in the elementary trees in STAG easily.

For the re-decoding framework, there are relatively more resources available to improve the performance in comparison with the paraphrasing framework, such as bilingual corpora. So our future work for this model involves the integration of existing translation probabilities trained from a bilingual corpus to the combination model.

For SDLM, there are several avenues for future work: we have focused on bigram dependencies in our models. The extension from bigram dependencies to more than two dependent elementary trees is straightforward. It would also be worth investigating the performance of using our sentence-level model to re-rank n-best outputs of a phrase-based

combination model.

For MEMT in general, our future research direction involves the design of a specific MEMT model, aiming to fuse outputs of semantic-based MT and statistical phrase-based MT engines, and investigate when and where to use the output of either engine. The motivation of this direction is because we believe the two kinds of engines reflect the two major brain operations a human uses to translate sentences - “understand (semantics)” and “memorize (phrase translations)”; people use the two kinds of operations to complete a translation process simultaneously. Leveraging the recent advances in semantic representation and parsing will enable the development of semantic-based MT systems; an MEMT system could integrate these semantic-based MT systems with the statistical phrase-based MT systems in order to mimic the two major brain operations for translation.

Bibliography

- [Aho and Ullman, 1969] A. V. Aho and J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3:37–56.
- [Alam et al., 2006] Md. Jahangir Alam, Naushad UzZaman and Mumit Khan. 2006. N-gram based Statistical Grammar Checker for Bangla and English. In *Proc. of ninth International Conference on Computer and Information Technology (ICCIT 2006)*, Dhaka, Bangladesh.
- [Atwell and Elliot, 1987] Eric S. Atwell and Stephen Elliot. 1987. Dealing with Ill-formed English Text. In: R. Garside, G. Leech and G. Sampson (Eds.), *The Computational Analysis of English: A Corpus-based Approach*. London: Longman.
- [Bangalore and Joshi, 1999] Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.
- [Bannard and Callison-Burch, 2005] Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*.
- [Barzilay and McKeown, 2005] Regina Barzilay and Kathleen McKeown. 2005. Sentence fusion for multi document summarization. *Computational Linguistics*, 31.
- [Callison-Burch et al., 2012] Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of WMT12*.
- [Cer et al., 2013] D. Cer, C. D. Manning, and D. Jurafsky. 2013. Positive diversity tuning for machine translation system combination. In *Proc. of WMT*.

- [Chiang, 2007] David Chiang. Hierarchical phrase-based translation. 2007. *Computational Linguistics*, 33(2):201–228.
- [Chen et al., 2007a] Boxing Chen, M. Federico and M. Cettolo. 2007a. Better N-best Translations through Generative n-gram Language Models. In *Proceeding of MT Summit XI*
- [Chen et al., 2005] Boxing Chen, Roldano Cattoni, Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. 2005. The ITC-irst SMT System for IWSLT-2005. In *Proceedings of IWSLT*
- [Chen et al., 2007b] Yu Chen, Andreas Eisele, Christian Federmann, Eva Hasler, Michael Jellinghaus, and Silke Theison. 2007b. Multi-engine machine translation with an open-source SMT decoder. In *Proceedings of WMT07*
- [Chen et al., 2009a] Boxing Chen, Min Zhang and Aiti Aw. 2009a. A Comparative Study of Hypothesis Alignment and its Improvement for Machine Translation System Combination. In: *Proceedings of ACL-IJCNLP*. pp. 1067-1074. Singapore. August.
- [Chen et al., 2009b] Yu Chen, Michael Jellinghaus, Andreas Eisele, Yi Zhang, Sabine Hunsicker, Silke Theison, Christian Federmann, Hans Uszkoreit. 2009b. Combining Multi-Engine Translations with Moses. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*
- [Chen and Vijay-Shanker, 2000] John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn treebank. In *Proceedings of the Sixth International Workshop on Parsing Technologies*
- [DeNeefe and Knight, 2009] Steve DeNeefe and Kevin Knight. 2009 Synchronous Tree Adjoining Machine Translation. In *Proceedings of EMNLP*
- [Devlin et al., 2011] Jacob Devlin, Antti-Veikko I. Rosti, Shankar Ananthakrishnan, and Spyros Matsoukas. 2011. System combination using discriminative cross-adaptation. In *Proc. IJCNLP*, pages 667–675.

- [Devlin and Matsoukas, 2012] J. Devlin and S. Matsoukas. 2012. Trait-based hypothesis selection for machine translation. In Proc. of NAACL.
- [Du et al., 2010] Jinhua Du, Pavel Pecina and Andy Way. 2010. An Augmented Three-Pass System Combination Framework: DCU Combination System for WMT 2010. In proceedings of the Fifth Workshop on Statistical Machine Translation
- [Du and Way, 2010] Jinhua Du and Andy Way. 2010. Using TERp to Augment the System Combination for SMT. In Proceedings of the Ninth Conference of the Association for Machine Translation (AMTA2010)
- [Fellbaum, 1998] Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database. MIT Press. <http://wordnet.princeton.edu/>
- [Feng et al., 2009] Yang Feng, Yang Liu, Haitao Mi, Qun Liu, and Yajuan Lu. 2009 Lattice-based System Combination for Statistical Machine Translation. In Proceedings of ACL
- [Filippova and Strube, 2008] Katja Filippova and Michael Strube. 2008. Sentence Fusion via Dependency Graph Compression. in the Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing
- [Frederking and Nirenburg, 1994] Robert Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In Proc. ANLP, pages 95–100.
- [Galley et al., 2006] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, Ignacio Thayer. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*
- [Gimpel et al., 2013] Kevin Gimpel, Dhruv Batra, Chris Dyer, and Gregory Shakhnarovich. 2013. A Systematic Exploration of Diversity in Machine Translation. In Proc. of EMNLP
- [Hardmeier et al., 2012] Christian Hardmeier, Joakim Nivre and Jörg Tiedemann. 2012. Tree Kernels for Machine Translation Quality Estimation. *In Proceedings of WMT12*

- [Hasan et al., 2006] S. Hasan, O. Bender, and H. Ney. 2006. Reranking translation hypotheses using structural properties. *In Proceedings of the EACL'06 Workshop on Learning Structured Information in Natural Language Applications*
- [Hassan et al., 2007] Hany Hassan , Khalil Sima'an and Andy Way. 2007. Supertagged Phrase-Based Statistical Machine Translation. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*
- [He et al., 2008] Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-hmm-based hypothesis alignment for computing outputs from machine translation systems. *In Proceedings of EMNLP*
- [He and Toutanova, 2009] Xiaodong He and Kristina Toutanova. 2009. Joint optimization for machine translation system combination. *In Proc. EMNLP, pages 1202–1211.*
- [Heafield and Lavie, 2010] Kenneth Heafield and Alon Lavie. 2010. Voting on N-grams for Machine Translation System Combination. *In Proceedings of Ninth Conference of the Association for Machine Translation in the Americas*
- [Heidorn, 2000] George E. Heidorn. 2000. Intelligent writing assistance. In R. Dale, H. Moisl and H. Somers (eds.), *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker, New York. pp. 181-207.
- [Huang et al., 2010] Anta Huang, Tsung-Ting Kuo, Ying-Chun Lai, Shou-De Lin. 2010. Identifying Correction Rules for Auto Editing. *In Proceedings of the 22nd Conference on Computational Linguistics and Speech Processing.*
- [Huang and Papineni, 2007] Fei Huang and Kishore Papineni. 2007. Hierarchical System Combination for Machine Translation. *In Proceedings of EMNLP-CoNLL*

- [Hildebrand and Vogel, 2008] Almut Silja Hildebrand and Stephan Vogel. 2008. Combination of machine translation systems via hypothesis selection from combined n-best lists. *In Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*
- [Huang and Papineni, 2007] Fei Huang and Kishore Papineni. 2007. Hierarchical System Combination for Machine Translation. In Proceedings of EMNLP-CoNLL
- [Jayaraman and Lavie, 2005] Shyamsundar Jayaraman and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In Proc. EAMT
- [Jensen et al., 1993] Karen Jensen, George E. Heidorn, and Stephen D. Richardson 1993 . PEG: the PLNLP English Grammar, in Jensen K., Heidorn G.E., & Richardson S.D., (Eds.), Natural Language Processing: The PLNLP Approach, Kluwer Academic Publishers, Boston, 29-45.
- [Joshi et al., 1975] Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Science*, 10:136–163.
- [Joshi and Srinivas, 1994] Aravind K. Joshi and B. Srinivas. 1994. Disambiguation of super parts of speech (or supertags): Almost parsing. *In Proceedings of the 15th International Conference on Computational Linguistics*
- [Karakos et al., 2008] Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine translation system combination using ITG-based alignments. *In Proceedings of ACL-HLT*
- [Klein and Manning, 2003] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.
- [Koehn et al., 2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In Proceedings of Human Language Technology Conference and Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)

[Koehn et al., 2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177{180, Morristown, NJ, USA, 2007. Association for Computational Linguistics.

[Leusch et al., 2011] Gregor Leusch, Markus Freitag, and Hermann Ney. The RWTH System Combination System for WMT 2011. 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*

[Leusch and Ney, 2010] Gregor Leusch and Hermann Ney. 2010. The RWTH System Combination System for WMT 2010. In *proceedings of the Fifth Workshop on Statistical Machine Translation*

[Ma and McKeown, 2011] Wei-Yun Ma and Kathleen McKeown. 2011. System Combination for Machine Translation Based on Text-to-Text Generation. In *Proceedings of Machine Translation Summit XIII*

[Ma and McKeown, 2012a] Wei-Yun Ma and Kathleen McKeown. 2012a. Phrase-level System Combination for Machine Translation Based on Target-to-Target Decoding. In *Proceedings of the 10th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, San Diego, CA.

[Ma and McKeown, 2012b] Wei-Yun Ma and Kathleen McKeown. 2012b. “Detecting and Correcting Syntactic Errors in Machine Translation Using Feature-Based Lexicalized Tree Adjoining Grammars”. In *Proceedings of Conference on Computational Linguistics and Speech Processing (ROCLING)*

[Ma and McKeown, 2012c] Wei-Yun Ma and Kathleen McKeown. 2012c. “Detecting and Correcting Syntactic Errors in Machine Translation Using Feature-Based Lexicalized Tree

Adjoining Grammars”. *International Journal of Computational Linguistics and Chinese Language Processing (IJCLCLP)*, Vol 17, No. 4, pp. 1-14.

[Ma and McKeown, 2013] Wei-Yun Ma and Kathleen McKeown. 2013. “Using a Supertagged Dependency Model to Select a Good Translation in System Combination”. In *Proceedings of NAACL-HLT*

[Macherey and Och, 2007] W. Macherey and F. J. Och. 2007. An empirical study on computing consensus translations from multiple machine translation systems. In *Proc. of EMNLPCoNLL*.

[Marsi and Kraemer, 2005] Erwin Marsi and Emiel Kraemer. 2005. Explorations in Sentence Fusion. In *Proceedings of the 10th European Workshop on Natural Language Generation*

[Matusov et al., 2006] Evgeny Matusov, Nicola Ueffing, and Hermann Ney 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. *In Proceedings of EACL*

[Matusov et al., 2008] E. Matusov, G. Leusch, R. E. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y. S. Lee, J. B. Marino, M. Paulik, S. Roukos, H. Schwenk, and H. Ney. 2008. System combination for machine translation of spoken and written language. *IEEE Transactions on Audio, Speech and Language Processing*, 16(7):1222–1237, September.

[Naber, 2003] Daniel Naber. 2003. A Rule-Based Style and Grammar Checker. Unpublished doctoral dissertation, University of Bielefeld, Germany.

[Narsale, 2010] Sushant Narsale. 2010. JHU System Combination Scheme for WMT 2010. In *Proceedings of the Fifth Workshop on Statistical Machine Translation*

[Och, 2004] Franz Josef Och. 2004. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*

[Och et al., 2004] Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain,

- Zhen Jin, and Dragomir Radev. 2004 A smorgasbord of features for statistical machine translation. *In Proceedings of the Meeting of the North American chapter of the Association for Computational Linguistics*
- [Och and Ney, 2004] Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation, *Computational Linguistics* 30(4).
- [Quirk et al., 2005] Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT, *In Proceedings of the Association for Computational Linguistics*
- [Porter, 1980] Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.
- [Rosti et al., 2007a] Antti-Veikko I. Rosti, Necip F. Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr. 2007a. Combining outputs from multiple machine translation systems. In *Proceedings of NAACL-HLT*
- [Rosti et al., 2012] Antti-Veikko I. Rosti, Xiaodong He, Damianos Karakos, Gregor Leusch, Yuan Cao, Markus Freitag, Spyros Matsoukas, Hermann Ney, Jason Smith, and Bing Zhang. 2012. Review of Hypothesis Alignment Algorithms for MT System Combination via Confusion Network Decoding , in *Proceedings of NAACL-HLT workshop on SMT (WMT)*, Association for Computational Linguistics
- [Rosti et al., 2007b] Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007b. Improved word-level system combination for machine translation. In *Proceedings of ACL*
- [Rosti et al., 2008] Antti-Veikko I. Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. Incremental Hypothesis Alignment for Building Confusion Networks with Application to Machine Translation System Combination. 2008. In *Proceedings of the Third Workshop on Statistical Machine Translation*
- [Rosti et al., 2010] Antti-Veikko I. Rosti, Bing Zhang, Spyros Matsoukas and Richard Schwartz. 2010. BBN System Description for WMT10 System Combination Task. In *proceedings of the Fifth Workshop on Statistical Machine Translation*

- [Schabes et al., 1988] Yves Schabes, Anne Abeille and Aravind K. Joshi. 1988. Parsing strategies with 'lexicalized' grammars: Application to Tree Adjoining Grammars. *In Proceedings of the 12th International Conference on Computational Linguistics*
- [Sim et al., 2007] K.C. Sim, W.J. Byrne, M.J.F. Gales, H. Sahbi and P.C. Woodland .2007. Consensus Network Decoding for Statistical Machine Translation System Combination. *In Proceedings of ICASSP*
- [Snover et al., 2006] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of Association for Machine Translation in the Americas
- [Snover et al., 2009] Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz, 2009. Fluency, Adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric. In Proceedings of the Fourth Workshop on Statistical Machine Translation
- [Stymne and Ahrenberg, 2010] Sara Stymne and Lars Ahrenberg. 2010. Using a Grammar Checker for Evaluation and Postprocessing of Statistical Machine Translation. In LREC.
- [Watanabe and Sumita, 2011] Taro Watanabe, Eiichiro Sumita. 2011. Machine Translation System Combination by Confusion Forest. In Proceedings of ACL
- [Wu and Palmer, 2011] Shumin Wu and Martha Palmer. 2011. Semantic Mapping Using Automatic Word Alignment and Semantic Role Labeling, In Proceedings of ACL11 Workshop, Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, ACL, pages 21-30., Portland, OR, June, 2011.
- [Xia, 1999] Fei Xia. 1999. Extracting Tree Adjoining Grammars from Bracketed Corpora. *In Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-1999)*
- [Xiao et al., 2013] T. Xiao, J. Zhu, and T. Liu. 2013. Bagging and boosting statistical machine translation systems. *Artif. Intell.*, 195.
- [Xu et al., 2011] Daguang Xu, Yuan Cao, Damianos Karakos. 2011. Description of the JHU System Combination Scheme for WMT 2011. *In Proceedings of the Sixth Workshop on Statistical Machine Translation*

- [Vijay-Shanker and Joshi, 1988] K. Vijay-Shanker and Aravind K. Joshi. 1988. Feature structure based tree adjoining grammar, in Proceedings of COLING-88, pp. 714-719.
- [Wu et al., 2006] Shih-Hung Wu, Chen-Yu Su, Tian-Jian Jiang, and Wen-Lian Hsu. 2006. An Evaluation of Adopting Language Model as the Checker of Preposition Usage. Proceedings from ROCLING.
- [XTAG Group, 2001] XTAG Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 01-03, University of Pennsylvania.
- [Zhao and He, 2009] Yong Zhao and Xiaodong He. 2009. Using n-gram based features for machine translation system combination. In Proceedings of the North American Chapter of the Association for Computational Linguistics
- [Zens and Ney, 2006] Richard Zens and Hermann Ney. 2006. N-Gram Posterior Probabilities for Statistical Machine Translation. In Proceedings of the NAACL Workshop on SMT