Dependency Structures
for
Statistical Machine Translation

Nguyen Bach

CMU-LTI-12-001

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

<u>Thesis Committee:</u>

Alex Waibel
Stephan Vogel
Ying Zhang
Colin Cherry

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies

Abstract

Dependency structures represent a sentence as a set of dependency relations. Normally the dependency structures from a tree connect all the words in a sentence. One of the most defining characters of dependency structures is the ability to bring long distance dependency between words to local dependency structures. Another the main attraction of dependency structures has been its close correspondence to meaning. This thesis focuses on integrating dependency structures into machine translation components including decoder algorithm, reordering models, confidence measure, and sentence simplification.

First, we develop four novel **cohesive soft constraints** for a phrase-based decoder namely exhaustive interruption check, interruption count, exhaustive interruption count, and rich interruption constraints. To ensure the robustness and effectiveness of the proposed constraints, we conduct experiments on four different language pairs, including English-{Iraqi, Spanish} and {Arabic, Chinese}-English. The improvements are in between **0.4** and **1.8** BLEU points. These experiments also cover a wide range of training corpus sizes, ranging from 500K sentence pairs up to 10 million sentence pairs. Furthermore, to show the effectiveness of our proposed methods we apply them to systems using a 2.7 billion words 5-gram LM, different reordering models and dependency parsers.

Second, to go beyond cohesive soft constraints, we investigate efficient algorithms for learning and decoding with **source-side dependency tree reordering models**. We propose a novel source-tree reordering model that exploits dependency subtree *inside* / *outside* movements and cohesive soft constraints. These movements and constraints enable us to efficiently capture the subtree-to-subtree transitions observed both in the source of word-aligned training data and in the decoding time. Representing subtree movements as features allows MERT to train the corresponding weights for these features relative to others in the model. Moreover, experimental results on English-{Iraqi, Spanish} show that we obtain improvements **+0.8** BLEU and **-1.4** TER on English-Spanish and **+0.8** BLEU and **-2.3** TER on English-Iraqi.

Third, we develop *Goodness*, a novel framework to predict word and sentence level of **machine translation confidence** with dependency structures. The framework allows MT systems to inform users which words are likely translated correctly and how confident it is about the whole sentence. Experimental results show that the MT error prediction accuracy is increased from **69.1** to **72.2** in F-score. The Pearson correlation between the proposed confidence measure and the human-targeted translation edit rate (HTER) is **0.6**. Improvements between **0.4** and **0.9** TER reduction are obtained with the n-best list reranking task using the proposed confidence measure. Also, we present a visualization prototype of MT errors at the word and sentence levels with the objective to improve post-editor productivity.

Finally, inspired by study in summarization we propose *TriS*, a novel framework to simplify source sentences before translating them. We build a **statistical sentence simplification** system with log-linear models. In contrast to state-of-the-art methods that drive sentence simplification process by hand-written linguistic rules, our method used a margin-based discriminative learning algorithm operates on a feature set. The feature set is defined on statistics of dependency structures as well as surface form and syntactic structures of sentences. A stack decoding algorithm is developed in order to efficiently generate and search simplification hypotheses. Experimental results show that the simplified text produced by the proposed system reduces **1.7** Flesch-Kincaid grade level when compared with the original text. We show that a comparison of a state-of-the-art rule-based system to the proposed system demonstrates an improvement of **0.2**, **0.6**, and **4.5** points in ROUGE-2, ROUGE-4, and $AveF_{10}$, respectively. We present subjective evaluations of the simplified translation quality for an English-German MT system.

Acknowledgments

First and foremost, I thank my advisors, Alex Waibel and Stephan Vogel. Alex is responsible for allowing me freely pursuit my research interest while at the same time makes sure I am on the right track. Stephan has brought me to the fascinating world of machine translation and patiently spent so much time brainstorming with me the thesis work. Both Alex and Stephan kindly guided and supported me through countless difficulties during my PhD study.

Next, I thank my terrific Committee, consisting of Joy Ying Zhang at CMU Silicon Valley and Colin Cherry from National Research Council. Joy is an extremely energetic colleague and always enthusiastic about my work, ever since my first year, and I thank him for his encouragement. Also, I am glad that I receive lots of his technical comments on the confidence measure models. Colin is like an informal external advisor. He kindly spent so much time helping me on the extension of cohesive soft constraints, initiating the idea of the source-side dependency tree reordering models, and constructively criticizing me on the sentence simplification models.

In addition, I would like to thank Alan Black for helping me both scientifically and professionally. Alan guided me in the TransTac project which is the most valuable and practical research experience that I have ever had. I am so grateful to receive tremendous help from him, including TAing his speech processing class two times.

I am also indebted to many collaborators during my PhD study, especially Qin Gao, Matthias Eck, Fei Huang, and Yaser Al-Onaizan. Qin is a fantastic officemate and collaborated with me on many projects including the source-side dependency tree reordering models and the statistical sentence simplification which laid down the foundation of the thesis. Matthias gave many important suggestions on the initial draft of the thesis such as the skeleton and structure of Chapter 1. I was also very lucky to have a 6-month internship with the IBM machine translation group hosted by Fei and Yaser. It was at IBM where I collaborated with Fei and Yaser to develop the confidence measure models.

Furthermore, I want to thank people at CMU. My peers not only helped me much in research but also buoyed my spirits, especially during difficult times. They include Vamshi Ambati, Sameer Badaskar, Susi Burger, Paisarn Charoenpornsawat, Sanjika Hewavitharana, Roger Hsiao, Silja Hildebrand, Isaac Harris, Qin Jin, Stan Jou, Thilo Kohler, Kornel Laskowski, Ian Lane, Florian Metze, Mohamed Noamany, Udhyakumar Nallasamy, ThuyLinh Nguyen, Avneesh Saluja, Wilson Tam, Ashish Venugopal, Bing Zhao, and Andreas Zollmann at InterACT, and Shilpa Arora, Shay Cohen, David-Huggins Daines, Kevin Gimpel, Greg Hanneman, Michael Heilman, Rohit Kumar, Jae Dong Kim, Abhimanyu Lad, Julianne Mentzer, Aaron Phillips, Alok Parlikar, Hideki Shima, Nico Schlaefer, Narges SharifRazavian, and Konstantin Salomatin at LTI.

I also thank our faculty members that I interacted with: Tanja Schultz, Lori Levin, Bob Frederking, Alon Lavie, William Cohen, and Noah Smith. Tanja, whose expertise on speech processing and warm-heartedness really impressed me, supervised me on a speech recognition project for Arabic. Lori helped me to understand lots of linguistic theoretical concepts. Bob and Alon provided an active MT research atmosphere at LTI that I am so lucky to be part of it. William's class inspired me the connection between information extraction and machine translation. Noah is the best instructor I ever had in NLP.

Outside academics, I was fortunate enough to be surrounded by many good Vietnamese friends in Pittsburgh including Minh Duong, Son Le, Quang Nguyen, MinhHoai Nguyen, Thao Pham, Tien Tran, and many others.

Last but also the most, I thank my family for their unconditional love and endless support throughout all these years, without which I could never have gone this far. Back to the old days, my father led me into the fascinating world of language technologies when I was a junior. My mother made sure that I will receive a solid support in whichever condition. My elder sister took the responsibility to take care of my parents when I was away from home. My wife supports me at the hardest time during the study. I dedicate this thesis to them.

Abbreviations

BLEU	Bilingual Evaluation Understudy
CFG	Contex-free Grammar
DG	Dependency Grammar
DP	Dynamic Programming
HPSG	Head-driven Phrase-structure Grammar
HTER	Human-targeted Translation Edit Rate
MERT	Minimum Error Rate Training
MIRA	Margin Infused Relax Algorithm
PBMT	Phrase-based Machine Translation
POS	Parts-of-Speech
SMT	Statistical Machine Translation
SAMT	Syntax-augmented Machine Translation
SBMT	Syntax-based Machine Translation
SPSG	Synchronous Phrase-structure Grammars
STSG	Synchronous Tree-substitution Grammars
TER	Translation Edit Rate

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Statistical Machine Translation (SMT) have been evolving rapidly during the last two decades of research and development. SMT paradigms have emerged from word-based translation (Brown et al., 1993) to phrase-based translation (Koehn et al., 2003) and syntax-based translation (Galley et al., 2004; Quirk et al., 2005; Liu et al., 2006; Chiang, 2005; Shen et al., 2008). In addition to standalone machine translation systems which are already being used in everyday life, SMT also plays a vital role in other applications such as speech translation, cross-lingual information retrieval/extraction, distillation systems and virtual world communication (Bach et al., 2007; Shima et al., 2008; Sudo et al., 2004; Zhang and Bach, 2009).

Considerable improvements have been made and high-quality machine translation can be obtained for language pairs, such as Spanish-English and French-English, which have overlaps in linguistic properties and phenomena such as vocabulary, cognate, and grammar . However, we still see unsatisfactory translations when translating from languages which have complicated structures such as from German, Chinese, and Japanese to English, or from English to rich morphological languages such as Arabic, Farsi, and Pashto. According to Vilar et al. (2006), incorrect translations occur mainly due to the following reasons:

- Word order: Word order errors occur when the translated words are in an incorrect order. This problem produces grammatically ill-formed sentences that might be hard to understand or even misleading.
- Missing words: Content words are omitted during the translation process and as a result machine translation output will lose key information.
- Incorrect words: Single words can have different senses depending on the context in which they are used. Those words are often translated using an incorrect word sense.

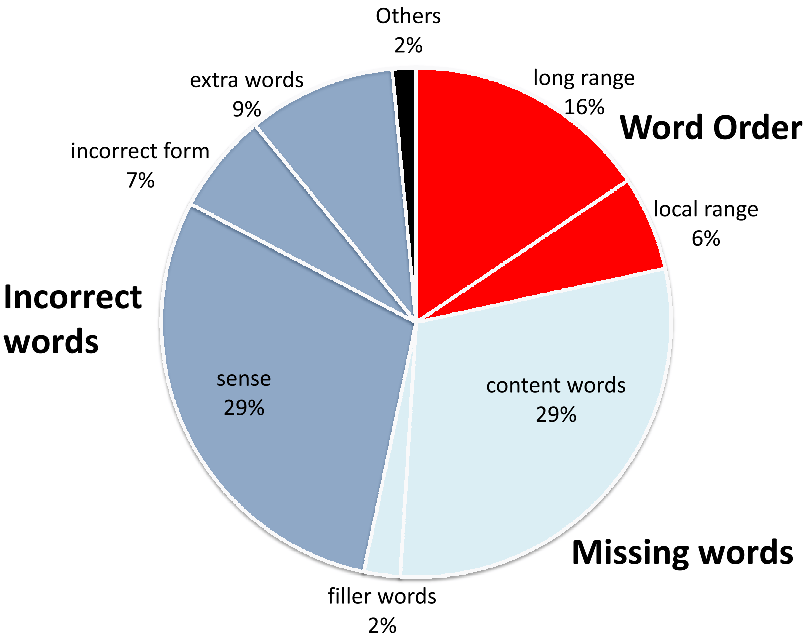


Figure 1.1: Error analysis for 311 sentences of the GALE P3.5 Chinese-English evaluation.

To get a better view of these types of errors, we performed an error analysis on the translation output of our system used for the GALE P3.5 Chinese-English evaluation. Our error classification criteria is trying to minimize the number of human edits to correct machine translation output such that it matches to the original meaning. Figure 1.1 shows that 21.6% of the errors come from the word order category, 31.7% from missing words, 45.3% from incorrect words and 1.4% from other errors. This analysis suggest that correcting word order may be helpful to alleviate the other issues because it helps the target language model to make better predictions.

Source sentence	在联想实现变革之时,柳传志也完成了一次从企业家到风险投资家的转变。
Machine translation output	lenovo to achieve change , liu chuanzhi also completed a risk investors from entrepreneurs to change .
Reference translation	when lenovo was carrying out changes , liu chuanzhi also changed his role from an entrepreneur to a venture investor .

Table 1.1: Example of errors

In addition, state-of-the-art SMT systems have made significant progress towards producing user-acceptable translation output. The example in Table 1.1 contains a Chinese source sentence, a machine translation output and an English reference. Without translation references, there is still no efficient way for MT systems to inform users which words are likely translated correctly and how confident it is about the whole sentence. Key words and phrases, such as “*lenovo*”, “*liu chuanzhi*” and “*entrepreneurs*”, are correctly translated, however, there are incorrect words for example “*venture*”-“*risk*” and ”*carries out*”-“*to achieve*”. Also incorrect word forms can be seen with “*entrepreneur*”-“*entrepreneurs*” and “*investors*”-“*investors*”.

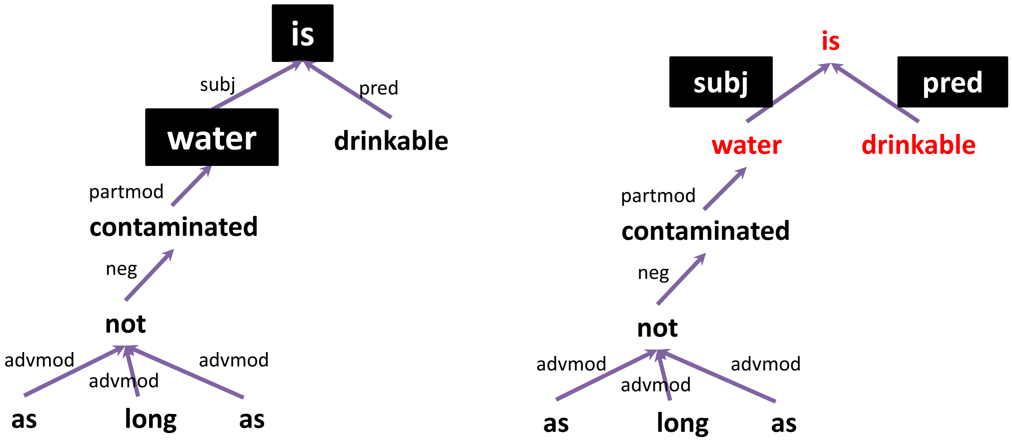
Furthermore, complicated sentences impose difficulties for translation. In the NIST evaluation, translation systems typically have to deal with sentences with average length ranging from 27 to 36 words varying on different test sets as shown in Table 1.2. There are cases when the test sentence has up to 268 words. Similar to other NLP tasks, such as parsing and semantic role labeling, the source sentence length has a lot of impact on SMT performance. Translating long sentences is often harder than short sentences because of several reasons. First, the hypothesis search space for long sentences is much larger than short sentences, and as a result, good translations are harder to reach. Second, it takes more time to translate long sentences. Third, long sentences often contain complex syntax and long range dependency structures, therefore, it is not easy for translation models to capture these phenomena.

Test sets	Average Length	Maximum length
mt02	29	81
mt03	28.42	86
mt04	31.76	111
mt05	31.51	101
mt06	27.68	205
mt08-nw	31.92	150
mt08-wb	36.22	268

Table 1.2: Sentence length statistics on NIST MT Arabic test sets

In many translation applications, such as speech-to-speech translation, the fluency may not be very important. For example, in speech-to-speech translation when the user says *”well well well my name you know is is John”* it is almost acceptable if the machine can output to the target language keywords *”my name John”*. On the other hand, complicated sentences impose difficulties not only on translation but also on reading comprehension. For instance, a person in 5th grade can comprehend a comic book easily but will struggle to understand New York Times articles which require at least 12th grade average reading level (Flesch, 1981).

Dependency structures can be used to tackle these problems. Dependency structures represent sentence as a set of dependency relations via dependency grammar, a type of grammar formalism. Normally the dependency relations from a tree connect all the words



(a) Long distance dependency

(b) Semantic relations

Figure 1.2: Example of long distance dependency and semantic relation between words properties for sentence “*water as long as not contaminated is drinkable*”

in a sentence. Dependency structures have been used in various semantic structure theories, for example in theories of semantic relations/cases/theta roles (arguments have defined semantic relations to the head/predicate) or in the predicate calculus (arguments depend on the predicate).

One of the most appealing characteristics of dependency structures is the ability to represent long distance dependency between words with local structures. Figure 1.2(a) shows the distance between words “*water*” and “*is*” in the surface form is 5 words, however, in dependency structure it becomes local. The other main attraction of traditional dependency structures has been its close correspondence to meaning. Figure 1.2(b) shows the relations between words “*water*” and “*is*” is a subject relation while “*is*” and “*drinkable*” is a predicate relation. The adoption of dependency structures would facilitate the machine translation system to reveal deep structures to be learned for modeling translation process.

A dependency-based approach to the problem of word and phrase reordering mitigates the need for long distance relations which become local in dependency tree structures. This property is attractive when machine translation needs to deal with languages with very different word orders, such as between subject-verb-object (SVO) and subject-object-verb (SOV) languages; long distance reordering becomes one of the key points. Dependency structures directly target lexical items which turn out to be simpler in form than phrase-structure trees since there are no constituent labels. Dependencies are usually meaningful - i.e. they usually carry semantic relations and are more abstract than surface order. Moreover, dependency relations between words directly model the semantic structure of a sentence. As such, dependency trees are a desirable prior model for the process of preserving semantic structures from source to target language via translation. Dependency structures have been shown to be a promising direction for several components of SMT such as word alignment (Ma et al., 2008), translation models (Shen et al., 2009; Xu et al., 2009; Carreras and Collins, 2009; Mi and Liu, 2010) and language models (Zhang, 2009; Shen et al., 2009).

1.2 Thesis Statement

This thesis work provides statistical models that incorporate dependency structures into MT systems. Source-side dependency structures are modeled as cohesive soft constraints in a beam-search phrase-based decoder. A source-side dependency tree reordering is proposed to exploits dependency subtree movements and constraints. These movements and constraints enable SMT models to efficiently capture the subtree-to-subtree transitions observed both in the source of word-aligned training data and in decoding time. When integrated into a machine translation system, both cohesive soft constraints and source-side dependency tree reordering models clearly improve the translation quality. In terms of confidence measure, this thesis provides a novel method to predict word-level and sentence-level MT errors with dependency structures features. The proposed confidence scores not only can help MT systems to select better translations but also can be visualized to improve usability. Finally, a novel statistical sentence simplification framework is proposed to simplify the source sentences before translating them. This framework reduces the education level required to understand a text.

1.3 Thesis Summary

We develop various algorithms to statistically incorporate dependency structures into MT components including the decoder, reordering models, confidence measure, and sentence simplification. We achieve improved BLEU and TER scores, increased MT translation quality prediction accuracy, and reduced the hardness of source sentences. We adopt the phrase-based MT system as our baseline. With different resources and different problems to solve, we first expand the baseline system in the following ways:

- Decoder: Given the source dependency tree we want to enforce the cohesive decoding strategy. We proposed four novel cohesive soft constraints namely exhaustive interruption check, interruption count, exhaustive interruption count, and rich interruption count. The cohesive-enhanced decoder performs statistically significant better than the standard phrase-based decoder on English-Spanish. Improvements in between **+0.4** and **+1.8** BLEU points are also obtained on English-Iraqi, Arabic-English, and Chinese-English systems.

- Reordering Models: To go beyond cohesive soft constraints, we investigate efficient algorithms for learning and decoding with source-side dependency tree reordering models. The phrase movements can be viewed as the movement of the subtree $inside$ or $outside$ a source subtree when the decoder is leaving from the previous source state to the current source state. The notions of moving $inside$ and $outside$ a subtree can be interpreted as tracking facts about the subtree-to-subtree transitions observed in the source side of word-aligned training data. With extra guidance on subtree movements, the source-tree reordering models help the decoder make smarter distortion decisions. We observe improvements of **+0.8** BLEU and **-1.4** TER on English-Spanish and **+0.8** BLEU and **-2.3** TER on English-Iraqi.

For confidence measure, we proposed $Goodness$, a method to predict confidence scores for machine translated words and sentences based on a feature-rich classifier using structure features. We develop three novel feature sets to capture different aspects of translation quality which have never been considered during the decoding time, including:

- Source and target dependency structure features that enable the classifier to utilize deep structures to predict translation errors.
- Source POS and phrase features which capture the surface source word context.
- Alignment context features that use both source and target word collocation for judging translation quality.

Experimental results show that by combining the dependency structure, source side information, and alignment context features with word posterior probability and target POS context the MT error prediction accuracy is increased from **69.1** to **72.2** in F-score. Our framework is able to predict error types, namely insertion, substitution and shift. The Pearson correlation with human judgment increases from **0.52** to **0.6**. Furthermore, we show that $Goodness$ can help the MT system to select better translations, and as a result, improvements between **0.4** and **0.9** TER reduction are obtained. We develop a visualization prototype using different font sizes and colors to catch the attention of post-editors whenever translation errors are likely to appear.

Finally, we develop $TriS$, a statistical sentence simplification system with log-linear models, to simplify source sentence before translating them. In contrast to state-of-the-art methods that drive sentence simplification process by hand-written linguistic rules, our method used a margin-based discriminative learning algorithm that operates on a feature set. We decompose the original dependency tree into context dependency structures and incorporate them as feature functions in the proposed model. The other feature functions are defined on statistics of the surface form as well as the syntactic structures of sentences. A stack decoding algorithm is developed to allow us to efficiently generate and search simplification hypotheses. The simplified text produced by the proposed system reduces **1.7** Flesch-Kincaid education level when compared with the original text. We show that a comparison of a state-of-the-art rule-based system to the proposed system demonstrates an improvement of **0.2**, **0.6**, and **4.5** points in ROUGE-2, ROUGE-4, and $AveF_{10}$, respectively. Subjective translation evaluations show that **63%** sentences with **manual** simplification translations are better than the original translation. Meanwhile, when applying **automatic** simplification translations **20%** sentences are better than the original translation.

1.4 Thesis Contribution

This thesis work advances the research on machine translation in the following ways:

- We designed a set of four novel cohesive soft constraints which characterize violations differently and allow penalties to persist as long as violations remain unresolved (Bach et al., 2009b).

- We developed a source-side dependency tree reordering model with $inside$ and $outside$ subtree movements that provide more structure evidence for the decoder to arrange target words in better orders (Bach et al., 2009a).

- The effectiveness robustness of the above models have been justified in multiple language pairs and different scales. We successfully apply the framework in English-Iraqi, English-Spanish, Arabic-English, and Chinese-English. These experiments also cover a wide range of training corpus sizes, ranging from 500 thousand sentence pairs up to 10 million sentence pairs. Furthermore, the effectiveness of our proposed models was shown when we applied them to systems using a 2.7 billion word 5-gram LM, different reordering models and dependency parsers (Bach et al., 2009a).

- We developed $Goodness$, a method for measuring machine translation confidence with source-target dependency structure features. Our method is able to predict error types namely insertion, substitution and shift. Based on this method, the MT error prediction accuracy is increased from **69.1** to **72.2** in F-score. We show that using $Goodness$ for reranking n-best lists improves the translation quality. Furthermore, we propose a method to visualize translation errors using confidence scores in order to improve the translation usability (Bach et al., 2011b).

- We developed $TriS$, a statistical sentence simplifier with log-linear models and margin-based discriminative training. This framework allows MT systems to perform factual-based simplification for source sentences before translating them. $TriS$

successfully reduces the education level required to under a text and improves the ROUGE score over a strong baseline simplification system (Bach et al., 2011a).

1.5 Thesis Structure

The rest of this thesis is structured as following:

In Chapter 2, we review the literature on machine translation, especially using dependency structures in MT.

In Chapter 3, we introduce cohesive soft constraints and demonstrate performance of translation systems with a cohesive-enhanced decoder in language pairs.

In Chapter 4, we present source-side dependency tree reordering models with subtree movements and constraints. This reordering model, combined with cohesive soft constraints in the decoder, demonstrates improvement on machine translation quality.

In Chapter 5, we focus on the confidence estimation problem. We propose $Goodness$, a method for measuring machine translation confidence. We show how machine translation systems can benefit from $Goodness$ through n-best list reranking and visualization prototype.

In Chapter 6, we describe $TriS$, a statistical sentence simplifier with log-linear models and margin-based discriminative training. We evaluate $TriS$ on a simplification task and a subjective machine translation evaluation.

Finally we conclude this thesis work with some conclusions and discussions.

Chapter 2

Literature Review

This chapter gives the overview of SMT, reviews concepts of dependency grammar, parsing and its applications to other fields and finally analyzes current work of using dependency structures in SMT. Section 2.1 gives a representative survey of SMT approaches, including phrase-based and syntax-based methods. Section 2.2 reviews the concepts of dependency grammar, parsing and applications which form the basis of our work. Section 2.3 will analyze recent work in machine translation using dependency structures. These include hierarchical dependency translation and cohesive phrase-based decoding. This chapter also serves as background material for the rest of this thesis on how SMT can be improved with cohesive soft constraints, source-side dependency tree reordering models, confidence scores, and sentence simplification.

2.1 Statistical Machine Translation

Statistical machine translation systems are based on the log-linear model which tries to provide a parameterized form of the probability of translating a sentence f_1^J to e_1^I, subject to

$$\hat{e}_1^{\hat{I}} = \arg\max_{\{e_1^I\}} P(e_1^I | f_1^J) \tag{2.1}$$

$P(e_1^I|f_1^J)$ can be modeled as a log-linear model with components $h_m(.)$ and scaling factors λ_m:

$$\begin{aligned} \hat{e}_1^{\hat{I}} &= \arg\max_{\{e_1^I\}} P(e_1^I|f_1^J) \qquad (2.2) \\ &= \arg\max_{\{e_1^I\}} \exp[\sum_1^M \lambda_m h_m(e_1^I, f_1^J)] \end{aligned}$$

This model can be derived from a word-aligned bitext. There are two ways to learn a word alignment matrix, namely 1) a generative approach based on the well-known so-called IBM word alignment models (Brown et al., 1993) with popular implementations such as GIZA++ (Och and Ney, 2003), MGIZA and PGIZA (Gao and Vogel, 2008); 2) a discriminative approach based on recent work of Liang et al. (2006b); Blunsom and Cohn (2006); Niehues and Vogel (2008). Components $h_m(.)$ are feature functions which can be learnt from phrase pairs or synchronous grammars. Moreover, scaling factors λ_m are trained to directly optimize automatic evaluation metrics like BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and METEOR (Agarwal and Lavie, 2008) using discriminative training algorithms such as minimum error rate training (MERT), margin-infused relax algorithm (MIRA), and pairwise ranking optimization (PRO) (Och, 2003; Watanabe et al., 2007; Hopkins and May, 2011).

2.1.1 Phrase-based Machine Translation

Phrase-based machine translation (PBMT) is driven by a phrasal translation model, which relates phrases (contiguous segments of words) in the source to phrases in the target (Och and Ney, 2004). A generative story of PBMT systems is

1) segment source sentence into phrases;
2) translate each phrase based on phrase tables;
3) permute translated phrases into their final order.

Phrases are extracted from a word alignment matrix (Koehn et al., 2003; Vogel, 2005). DeNero and Klein (2008) prove that finding an optimal phrase alignment over the combinatorial space of bijective phrase alignments is an NP-hard problem. A common feature set in a PBMT system (Koehn et al., 2007) includes language model probability $P(e_1^I)$, reordering model cost, phrase translation probability $P(f_1^J|e_1^I)$, reverse phrase translation probability $P(e_1^I|f_1^J)$, lexical weighting in both directions, phrase penalty, and unknown word penalty.

In PBMT, the $argmax$ of Equation 2.2 is the search problem that we have to maximize over all possible e_1^I and over all possible phrase segmentations. It is infeasible to enumerate all e_1^I. In fact, if one allows unrestricted changes in word order during translation, that alone is sufficient to show it to be NP complete, by analogy to the Traveling Salesman Problem (Knight, 1999). The search in phrase-based machine translation is based on beam search with heuristic scoring functions. It is a kind of $A^\star$ search even though there is no guarantee that scoring functions are admissible.

A beam search phrase-based decoder (Vogel, 2003; Koehn et al., 2007) uses a two-stage process that first builds a translation lattice and then searches for the best path through the lattice. The translation lattice is built by using all available translation pairs from the translation models for the given source sentence and inserting them into a lattice. These translation pairs consist of words or phrases on the source side that cover a part of the source sentence. The decoder inserts an additional edge for each phrase pair and attaches the target side of the translation pair and translations scores to the edge. The translation lattice will now contain a large number of possible paths that cover each source word exactly once (a combination of partial translations of words or phrases). These translation hypotheses will greatly vary in quality and the decoder uses the different knowledge sources and scores to find the best path possible translation hypothesis. This step also allows for limited reordering within the found translation hypotheses. To guide the search, each state in the translation lattice is associated with two costs which are current and future translation costs. The future cost is an estimation for translating the remaining words in the source sentence. The current cost is the total cost of phrases that have been translated so far in the current partial hypothesis, that is the sum of features' costs.

Despite the importance of word ordering, the popular phrase-based translation paradigm (Koehn et al., 2003) devotes surprisingly little modeling capacity to the issue. A very simple reordering model is to base on the cost for word movement only the distance in the source sentence between the previous and the current word or phrase during the translation process. More recently, data-driven models, which condition the probability of phrase-to-phrase transitions on the words involved, have been proposed to address this issue (Tillman, 2004; Koehn et al., 2005; Al-Onaizan and Papineni, 2006; Kuhn et al., 2006; Galley and Manning, 2008). Alternatively, one can employ syntax in the modeling of movement. By viewing language in terms of its hierarchical structure, one can more easily expose regularities in the sorts of movement that occur during translation. Each of these approaches requires a parser-like decoder and represents a departure from phrase-based decoding.

Phrasal decoding can be augmented easily, either by syntactic pre-processing or through search-space constraints. Pre-processing approaches parse the source sentence and use the tree to apply rules which re-order the source into a more target-like structure before the translation begins. These rules can be learned (Xia and McCord, 2004; Rottmann and Vogel, 2007; Tromble and Eisner, 2009) or designed manually (Collins et al., 2005; Wang et al., 2007; Xu et al., 2009). The pre-processing approach benefits from its simplicity and modularity, but it suffers from providing at most a one-best guess at syntactic movement. Search-space constraints limit the phrasal decoder's translation search using syntactic intuitions. Zens et al. (2004) demonstrated how to incorporate formally syntactic binary-bracketing constraints into phrase-based decoding. Recently, it has been shown that syntactic cohesion, the notion that syntactic phrases in the source sentence tend to remain contiguous in the target (Fox, 2002), can be incorporated into phrasal decoding as well, by following the simple intuition that any source subtree that has begun translation, must be completed before translating another part of the tree (Cherry, 2008; Yamamoto et al., 2008; Chang et al., 2009).

2.1.2 Syntax-based Machine Translation

The idea of modeling syntactic information in machine translation is an old idea. A syntactic translation framework has been proposed by Yngve (1958) who viewed translation as a 3-stage process namely

1) analyze source sentence as phrase structure representations;

2) transfer them into equivalent target phrase structures;

3) apply target grammar rules to generate output translation.

The research community observes strong improvements from syntax-based machine translation systems (SBMT) in recent years. The break-through is the combination of syntax with statistics and very large training data, along with synchronous grammar formalisms.

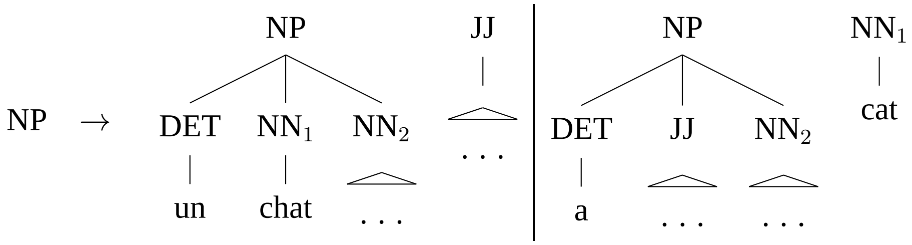
Synchronous grammar formalisms often start from phrase-structure grammars which are based on phrase-structure rules, for example NP → DET JJ NN. The idea of phrase structure comes from the observation that words are grouped with increasing hierarchical orders in trees and labeled with phrase labels such as verb phrase (VP), noun phrase (NP), prepositional phrase (PP) and sentence (S). Leaf nodes are normally labeled by part-of-speech tags. The Chomsky Hierarchy (Chomsky, 1956) can be used to classify phrase-structure grammars according to the form of their productions.

The first class of SBMT tries to explicitly model the translation process via synchronous phrase-structure grammars (SPSG) which can be viewed as a string-to-tree approach. SPSGs create two trees at the same time, one of the source sentence and one of the target sentence of a machine translation application. For example, a French noun phrase *un chat Siamois blanc* with English translation *a white Siamese cat* will have synchronous rules as

$$\begin{aligned}
&\text{NP} \rightarrow DET_1\ NN_2\ NN_3\ JJ_4 \mid DET_1\ JJ_4\ NN_3\ NN_2\\
&\text{NP} \rightarrow \text{un chat Siamois } JJ_1 \mid \text{a } JJ_1 \text{ Siamese cat}\\
&\text{NP} \rightarrow \text{un chat Siamois blanc} \mid \text{a white Siamese cat}
\end{aligned}$$

Each rule will associate with a set of features and typically include features from PBMT. A translation hypothesis is scored as a product of all derivation rules associated with language models. Wu (1997) proposes bilingual bracketing grammar which uses only binary rules and works well in many cases of word alignments and as well as word reordering constraints in decoding algorithms. Chiang (2005, 2007) presents hierarchical phrase models (Hiero) which combine the ideas of phrase-based models and tree structure and proposes an efficient decoding method based on chart parsing. Hiero's grammar does not build on any syntactic annotation and has only one nonterminal node X. Zollmann and Venugopal (2006) add syntactic categories to target-side nonterminals in Hiero which leads to syntax-augmented MT models. DeNeefe et al. (2007) develop rule extraction algorithms which not only learn syntactic translation but also help to improve coverage.

The second class of approaches is tree-to-tree and tree-to-string models which use synchronous tree-substitution grammars (STSG). The SPSG formalism is extended to include not only nonterminal and terminal symbols but also trees on the right hand side of rules. The trees have either nonterminal or terminal symbols at their leafs. All nonterminal symbols on the right hand side are mapped one-to-one between the two languages. For the example of a French noun phrase *un chat Siamois blanc* and an English translation *a white Siamese cat*, a STSG rule could be



STSGs allow the generation of non-isomorphic trees and overcome the child node reordering restriction of flat context-free grammars (Eisner, 2003). STSG rules are applied the same way as SPSG rules, except that additional structure is introduced. If we do not care about this additional structure, SPSG rules can be obtained by flattening STSG rules. Galley et al. (2004, 2006) present the GHKM rule extraction which is similar to phrase-based extraction in that it extracts rules consistent with given word alignments. However, a primary difference is the use of syntax trees on the target side, rather than sequences of words. Since STSGs conventionally only consider 1-best tree, therefore, they are vulnerable to parsing error and rule coverage as a results models lose a larger amount of linguistically unmotivated mappings. Liu et al. (2009) propose a solution by replacing the 1-best tree with a packed forest. Related works in is this area are Liu et al. (2006); Cowan et al. (2006); Zhang et al. (2008); Nesson et al. (2008); DeNeefe and Knight

(2009); Carreras and Collins (2009).

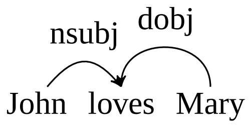
To find the best derivation in SBMT models, cubic time probabilistic bottom-up chart parsing algorithms, such as CKY or Earley, are often applied. The left hand side of both SPSG and STSG rules contains only one nonterminal node which allows to employ efficient dynamic programming decoding algorithms with recombination and pruning strategies (Huang and Chiang, 2007; Koehn, 2010). Probabilistic CKY/Earley decoding style often has to deal with binary-branching grammar to reduce the number of extracted rules, the number of chart entries and the number of stack combinations (Huang et al., 2009). Furthermore, incorporating ngram language models in decoding increases the computational complexity significantly. Venugopal et al. (2007) propose to do a first pass translation without the language model, and then score the pruned search hyper graph in a second pass with the language model. Zollmann et al. (2008) present a systematic comparison between PBMT and SBMT systems in different language pairs and system scales. They show that for language pairs which have sufficiently non-monotonic linguistic properties, SBMT approaches can yield substantial benefits.

2.2 Dependency Grammar, Parsing and Applications

We reviewed the background of the fundamental translation framework in the previous sections. In this section we are going to review dependency grammar, parsing and its applications.

2.2.1 Dependency Grammar

In modern linguistic theories, dependency grammars (DG) have been introduced by the French linguist Lucien Tesnière in the book *Éléments de Syntaxe Structurale* published in 1959. The key idea is all words depend on other words in a sentence. There is a special word called *root* that does not depend on any other. Dependencies are motivated by grammatical function, i.e. both syntactically and semantically. A word depends on another either if it is a complement or a modifier of the latter. In most formulations of DG for example



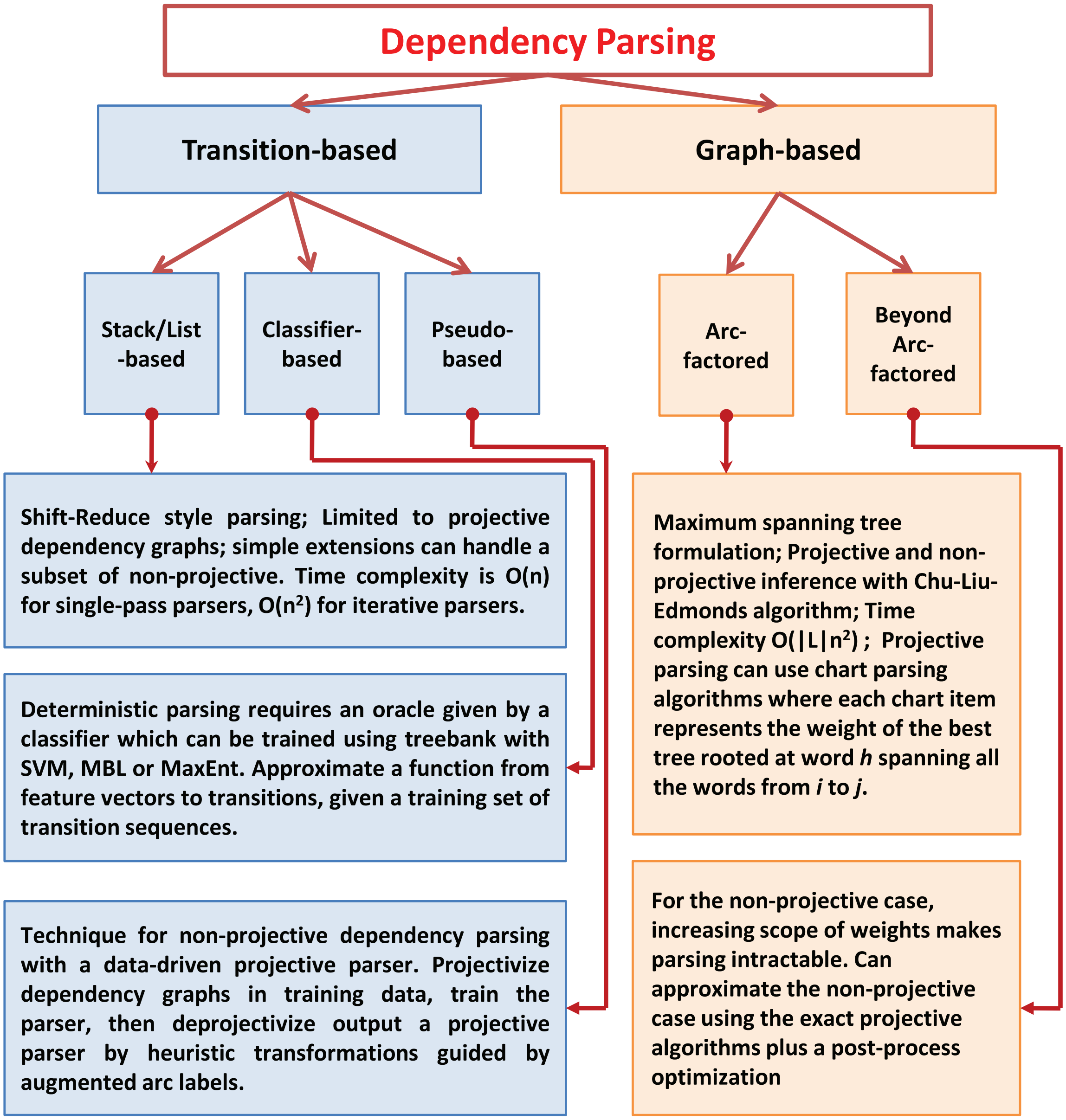
functional heads or governors (e.g. verbs) subcategorize for their complements. The transitive verb like *love* requires two complements (dependents), one noun with the grammatical function subject and one with the function object, hence, a grammatical function can be defined as *love(John, Mary)*.

Dependency structures	Phrase structures
head-dependent relations (directed arcs)	phrases (nonterminal nodes)
functional categories (arc labels)	structural categories (nonterminal labels)
possibly some structural categories (parts-of-speech)	possibly some functional categories (grammatical functions)

Table 2.1: A comparison between representations of dependency structures and phrase-structures

After Lucien Tesnière, Hays (1964) and Gaifman (1965) study mathematical properties of DGs. They show that theoretically it is straightforward to convert a constituency tree to an unlabeled dependency tree. A prerequisite is that every constituent has a unique head child. Robinson (1967) presents two methods to convert a phrase-structure grammar to a DG and reverse. Later on, Robinson (1970) formulates four axioms to govern the well-formedness of dependency structures. Magerman (1995) uses head percolation tables to identify head child in a constituency representation. Head percolation tables were first implemented in Collins' parser (Collins, 1999). The dependency tree is obtained by recursively applying head child and non-head child heuristics (Xia and Palmer, 2001). Table 2.1 shows a comparison between representations of dependency structures and phrase structures.

2.2.2 Dependency Parsing



Shift-Reduce style parsing; Limited to projective dependency graphs; simple extensions can handle a subset of non-projective. Time complexity is O(n) for single-pass parsers, O(n^2) for iterative parsers.

Deterministic parsing requires an oracle given by a classifier which can be trained using treebank with SVM, MBL or MaxEnt. Approximate a function from feature vectors to transitions, given a training set of transition sequences.

Technique for non-projective dependency parsing with a data-driven projective parser. Projectivize dependency graphs in training data, train the parser, then deprojectivize output a projective parser by heuristic transformations guided by augmented arc labels.

Maximum spanning tree formulation; Projective and non-projective inference with Chu-Liu-Edmonds algorithm; Time complexity O(|L|n^2) ; Projective parsing can use chart parsing algorithms where each chart item represents the weight of the best tree rooted at word *h* spanning all the words from *i* to *j*.

For the non-projective case, increasing scope of weights makes parsing intractable. Can approximate the non-projective case using the exact projective algorithms plus a post-process optimization

Figure 2.1: Taxonomy of supervised dependency parsing aprroaches.

From the view of graph theory, Kübler et al. (2009) define a dependency structure for sentence $S = w_0 w_1 ... w_n$ with relation set R as a directed graph $G(V, A)$ where V is a set V of vertices, $V \subseteq [w_0, w_1, ..., w_n]$, A is a set of directed edges, $A \subseteq V \text{x} R \text{x} V$ and if $(w_i, r, w_j) \in A$ then $(w_i, r', w_j) \notin A$ for $\forall r' \neq r$.

The task of dependency parsing is to analyze a sentence in terms of a set of directed links (dependencies) expressing the relationships which form the basis of the predicate argument structure such as head-modifier and head-complement. Projective dependency trees have the subtree, rooted at each word, which covers a contiguous substring of the sentence. In other words projective dependency trees are ones where edges do not cross (when drawn on one side). English is mostly projective and others are arguably less projective, especially Czech, Dutch and German. Projective dependency parsing means searching only for projective trees. Projective dependency grammars generate context-free languages, while non-projective dependency grammars can generate context-sensitive languages.

However, dependency parsing can be seen in a broader sense including any approach to parsing that makes use of word-to-word dependencies, such as lexicalized statistical parsers (Collins, 1999) or parsers based on lexicalized grammar formalisms (LFG, HPSG, CCG, LTAG, ...). Figure 2.1 is a taxonomy of supervised dependency parsing approaches[1]. Besides, unsupervised dependency parsing receives a considerable attention and obtains promising results (Cohen et al., 2008; Headden III et al., 2009).

2.2.3 Applications

Since dependency structures annotate relationship between entities, therefore, it is desirable to extract relations based on dependency structures. Relation extraction methods are useful in discovering protein-protein interactions, and gene-binding conditions (Goertzel et al., 2006). Patterns like “Protein X binds with Protein Y” are often found in biomedical texts, such as MedLine database, where the protein names are entities which are held together by the “bind” relation. Such protein-protein interactions are useful for applications like drug discovery etc. Other relations of interest are, a protein’s location inside

[1] Those who are interested in details of algorithms should read Kübler et al. (2009).

an organism. Such ternary relationships are extracted using linear kernels computed over features (Liu et al., 2007). Cancer researchers can use inferences like “Gene X with mutation Y leads to malignancy Z” in order to isolate cancerous genes. These information patterns can be pieced together by extracting ternary relations between genes, mutations and malignancy conditions in a large corpus of biotext (Fundel et al., 2007; Erkan et al., 2007).

Applications are not only in bio-text mining but also in relation extraction for textual entailment and question answering. If a query to a search engine is “When was Gandhi born ?”, then the expected answer would be “Gandhi was born in 1869”. The template of the answer is <PERSON> born-in <YEAR> which is nothing but the relational triple born-in(PERSON, YEAR) where PERSON and YEAR are the entities. To extract the relational triples, a large database (ex: web) can be queried using a small initial question-answer set (ex: “Gandhi 1869”). The best matching (or most confident) patterns are then used to extract answer templates which in turn can be used to extract new entities from the database(Wu et al., 2009; Mehdad and Magnini, 2009).

2.3 Dependency Structures and Machine Translation

We reviewed the background of the machine translation and dependency structures in the previous sections. In this section we are going to study how dependency structures have been applied to SMT.

The first class of approaches tries to explicitly model dependency structures in MT via tree-to-tree translation. Lin (2004b) propose a translation framework which assembles linear path through a source-side dependency tree. The training algorithm extracts a set of paths on the source dependency trees and determines the corresponding translations of the paths using word alignments. The outcome of training is a set of transfer rules that given a certain path in the source, provide the equivalent translation fragment in the target. Ding and Palmer (2005) develop a similar tree-to-tree system based on synchronous dependency insertion grammars (SDIG). The basic units of SDIGs are elementary trees which are dependency subtrees containing one or more lexical entries. The assumption during decoding is tree transformation is isomorphic at the cross-lingual level and any non-isomorphism is encapsulated within the elementary trees. However, both Lin (2004b) and Ding and Palmer (2005) did not incorporate a language model or discriminative reordering models which led to disappointing performances in terms of BLEU scores.

The dependency treelet translation model proposed by Quirk et al. (2005) is another class of approaches. A treelet is defined as an arbitrary connected subtree of a dependency tree. The treelet system parses the source side of the training data, projects these dependency tree onto the target side using word alignments, then extracts dependency treelet pairs. The decoder applies the bottom-up decoding strategy over the source dependency tree with treelet pairs. Translation hypotheses are scored by a log-linear model incorporating typical features such as language models, word alignment and reordering models. Chang and Toutanova (2007) present a discriminative syntax-based order model that ranks n-best outputs of the treelet system using local features that capture head-relative movements and global features that capture the word movement in a sentence. Menezes and Quirk (2007) introduce dependency order templates which are unlexicalized transduction rules mapping dependency tree containing only POS to unlexicalized target trees. Dependency order templates try to avoid the combinatorial explosion of reordering treelets in Quirk et al. (2005).

The third class of approaches is string-to-dependency models. Shen et al. (2008) develop a string-to-dependency translation framework (HierDec) which constructs the target side from well-formed dependency structures. Their system is similar to the hierarchical phrase translation model of Chiang (2005, 2007) with the following differences 1) the target side of the synchronous rule contains well-formed dependency structures; 2) it operates on dependency structures; 3) a dependency language model on the target side. Shen et al. (2009) strengthen their 2008 HierDec system with linguistic and contextual features such as non-terminal labels, non-terminal length distribution, context language model and source dependency language model.

Other recent works also cover quasi-synchronous dependency grammar (DG) proposed by Smith and Eisner (2006) and later on Gimpel and Smith (2009) develop lattice parsing with quasi-synchronous DG. Owczarzak et al. (2007) and Kahn et al. (2008) develop automatic methods to evaluate machine translation output based on dependency structures.

Chapter 3

Cohesive Soft Constraints in A Beam Search Phrase-based Decoder

In this chapter, we explore the cohesive phrasal decoding approach, focusing on empirical issues left unexplored by previous investigations. Cherry (2008) proposed the notion of a soft cohesion constraint, where detected violations are allowed during decoding, but incur a penalty. The cohesion-enhanced decoder enforces the following constraint: once the decoder begins translating any part of a source subtree, it must cover all the words under that subtree before it can translate anything outside of it. The flexibility of a soft penalty is appealing, given that cohesion does not perfectly characterize translation movement (Fox, 2002). But while cohesive decoding is well-defined for a hard constraint, soft constraints leave room for several design decisions. Should penalties persist as long as violations remain unresolved? Are some violations worse than others? Do cohesive soft constraints also improve systems that already benefit from large language models or lexical re-ordering models? We investigate these questions with a number of variant cohesive soft constraints. Furthermore, experimental results have so far been reported for English, French and Japanese only. We add to this body of work substantially, by experimenting with Spanish, Chinese, Iraqi and Arabic. Finally, we investigate the impact of the choice of parser and parse quality on cohesive decoding.

3.1 Cohesive Soft Constraints

In phrase-based machine translation, decoding the source sentence takes the form of a beam search through the translation space, with intermediate states corresponding to partial translations. The decoding process advances by extending a state with the translation of a source phrase, until each source word has been translated exactly once. Re-ordering occurs when the source phrase to be translated does not immediately follow the previously translated phrase. This is penalized with a discriminatively-trained distortion penalty. In order to calculate the current translation score, each state can be represented by a triple:

- A coverage vector C indicates which source words have already been translated.
- A span $\bar{f}$ indicates the last source phrase translated to create this state.
- A target word sequence stores context needed by the target language model.

As cohesion concerns only movement in the source sentence, we can completely ignore the language model context in our description of the different cohesion constraints, i.e. we will show the decoder state only as a $(\bar{f}, C)$ tuple.

To enforce cohesion during the state expansion process, cohesive phrasal decoding has been proposed in (Cherry, 2008; Yamamoto et al., 2008). The cohesion-enhanced decoder enforces the following constraint: once the decoder begins translating any part of a source subtree, it must cover all the words under that subtree before it can translate anything outside of it. This notion can be applied to any projective tree structure, but we follow Cherry (2008) and use dependency trees, which have been shown to demonstrate greater cross-lingual cohesion than other structures (Fox, 2002). We use a tree data structure to store the dependency tree. Each node in the tree contains surface word form, word position, parent position, dependency type and POS tag. An example of the dependency tree data structure is shown in Figure 3.1. We use T to stand for our dependency tree, and $T(n)$ to stand for the subtree rooted at node n. Each subtree $T(n)$ covers a span of contiguous source words; for subspan $\bar{f}$ covered by $T(n)$, we say $\bar{f} \in T(n)$.

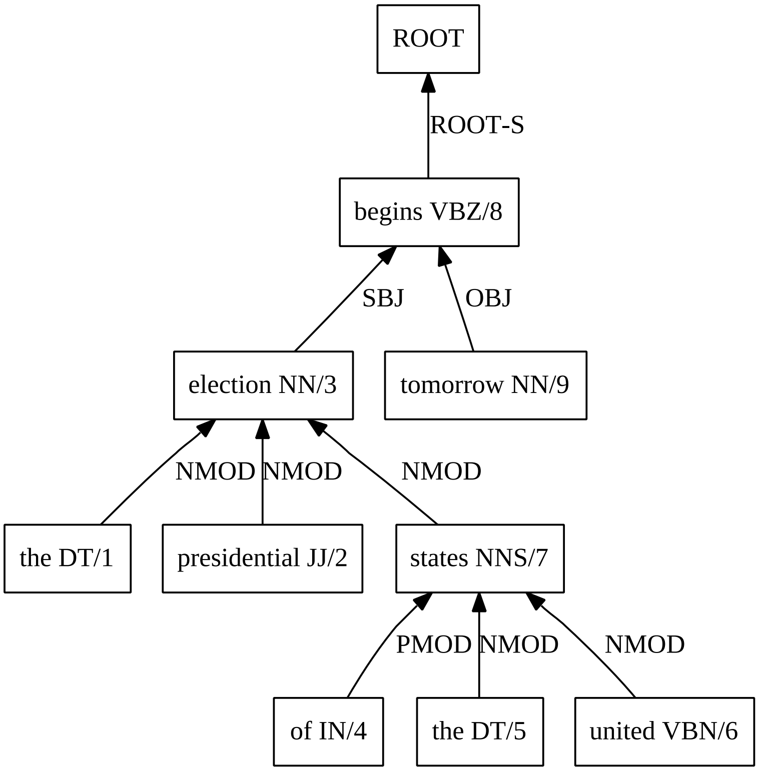


Figure 3.1: Example of an English source-side dependency tree structure for the sentence "the presidential election of the united states begins tomorrow".

Cohesion is checked as we extend a state $(\bar{f}_h, C_h)$ with the translation of $\bar{f}_{h+1}$, creating a new state $(\bar{f}_{h+1}, C_{h+1})$. Algorithm 1 presents the cohesion check described by Cherry (2008). Line 3 selects focal points, based on the last translated phrase. Line 5 climbs from each focal point to find the largest subtree that needs to be completed before the translation process can move elsewhere in the tree. Line 6 checks each such subtree for completion. Since there are a constant number of focal points (always 2) and the tree climb and completion checks are both linear in the size of the source, the entire check can be shown to take linear time.

The selection of only two focal points is motivated by a "**violation free**" assumption. If one assumes that the translation represented by $(\bar{f}_h, C_h)$ contains no cohesion violations,

Algorithm 1 Interruption Check (Coh1) (Cherry, 2008)

```
Input: Source tree $T$, previous phrase $\bar{f}_h$, current phrase $\bar{f}_{h+1}$, coverage vector $C_h$
$Interruption \leftarrow False$
$C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
$F \leftarrow$ the left and right-most tokens of $\bar{f}_h$
for each of $f \in F$ do
    Climb the dependency tree from $f$ until you reach the highest node $n$ such that $\bar{f}_{h+1} \notin T(n)$.
    if $n$ exists and $T(n)$ is not covered in $C_{h+1}$ then
        $Interruption \leftarrow True$
    end if
end for
Return $Interruption$
```

then checking only the end-points of $\bar{f}_h$ is sufficient to maintain cohesion. However, once a soft cohesion constraint has been implemented, this assumption no longer holds.

3.1.1 Exhaustive Interruption Check

Because of the "violation free" assumption, Algorithm 1 implements the design decision to only suffer a violation penalty once, when cohesion is initially broken. However, this is not necessarily the best approach, as the decoder does not receive any further incentive to return to the partially translated subtree and complete it.

For example, Figure 3.2 illustrates a translation candidate of the English sentence "the presidential election of the united states begins tomorrow" into French. We consider $\bar{f}_4$ = "begins", $\bar{f}_5$ = "tomorrow". The decoder already translated "the presidential election" making the coverage vector C_5 = "1 1 1 0 0 0 0 1 1". Algorithm 1 tells the decoder that no violation has been made by translating "tomorrow" while the decoder should be informed that there exists an outstanding violation. Algorithm 1 found the violation when the decoder previously jumped from "presidential" to "begins", and will not find another

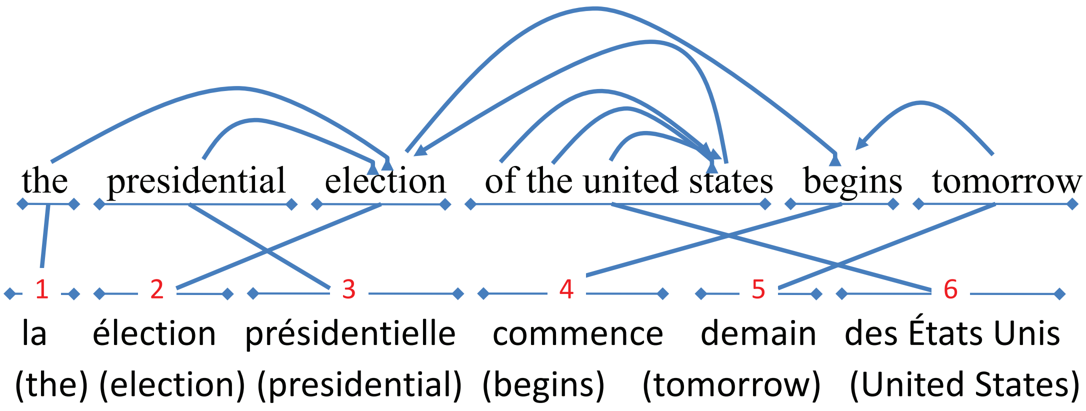


Figure 3.2: A candidate translation where Algorithm 1 does not fire

Algorithm 2 Exhaustive Interruption Check (Coh2)

```
Input: Source tree $T$, previous phrase $f_h$, current phrase $f_{h+1}$, coverage vector $C_h$
$Interruption \leftarrow False$
$C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
$F \leftarrow \{f | C_h(f) = 1\}$
for each of $f \in F$ do
    Climb the dependency tree from $f$ until you reach the highest node $n$ such that $\bar{f}_{h+1} \notin T(n)$.
    if $n$ exists and $T(n)$ is not covered in $C_{h+1}$ then
        $Interruption \leftarrow True$
    end if
end for
Return $Interruption$
```

violation when it jumps from “begins” to “tomorrow”.

Algorithm 2 is a modification of Algorithm 1, changing only line 3. The resulting system checks all previously covered tokens, instead of only the left and right-most tokens of $\bar{f}_h$, therefore, makes no violation-free assumption. In the example above, Algorithm 2 will inform the decoder that translating “tomorrow” also incurs a violation. Because $|F|$ is no longer constant, the time complexity of Coh2 is worse than Coh1. However, we can speed up the interruption check algorithm by hashing cohesion checks, so we only need to run Algorithm 2 once per $(\bar{f}_{h+1}, C_{h+1})$.

3.1.2 Interruption Count and Exhaustive Interruption Count

Algorithm 3 Interruption Count (Coh3)

```
  Input: Source tree $T$, previous phrase $\bar{f}_h$, current phrase $\bar{f}_{h+1}$, coverage vector $C_h$
 1: $ICount \leftarrow 0$
 2: $C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
 3: $F \leftarrow$ the left and right-most tokens of $\bar{f}_h$
 4: for each of $f \in F$ do
 5:     Climb the dependency tree from $f$ until you reach the highest node $n$ such that
        $\bar{f}_{h+1} \notin T(n)$.
 6:     if $n$ exists then
 7:         for each of $e \in T(n)$ and $C_{h+1}(e) = 0$ do
 8:             $ICount = ICount + 1$
 9:         end for
10:     end if
11: end for
12: Return $ICount$
```

Algorithm 1 and 2 described above interpret an interruption as a binary event. As it is possible to leave several words untranslated with a single jump, some interruptions may be worse than others. To implement this observation, an interruption count is used

Algorithm 4 Exhaustive Interruption Count (Coh4)

```
**Input:** Source tree $T$, previous phrase $f_h$, current phrase $f_{h+1}$, coverage vector $C_h$
1:  $ICount \leftarrow 0$
2:  $C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
3:  $F \leftarrow \{f | C_h(f) = 1\}$
4:  for each of $f \in F$ do
5:      Climb the dependency tree from $f$ until you reach the highest node $n$ such that $\bar{f}_{h+1} \notin T(n)$.
6:      if $n$ exists then
7:          for each of $e \in T(n)$ and $C_{h+1}(e) = 0$ do
8:              $ICount = ICount + 1$
9:          end for
10:     end if
11: end for
12: Return $ICount$
```

to assign a penalty to cohesion violations, based on the number of words left uncovered in the interrupted subtree. For the example in Section 3.1.1, Algorithm 4 will return 4 for $ICount$ (“of”; “the”; “united”; “states”). The modification of Algorithm 1 and 2 lead to Interruption Count (Coh3) and Exhaustive Interruption Count (Coh4) algorithms, respectively. The changes only happen in lines 1, 6 and 7. We use an additional bit vector to make sure that if a node has been reached once, it is not counted again during the same interruption check.

3.1.3 Rich Interruption Constraints

The cohesion constraints in Sections 3.1.1 and 3.1.2 do not leverage node information in the dependency tree structures. We propose the rich interruption constraints (Coh5) algorithm to combine four constraints which are Interruption, Interruption Count, Verb Count and Noun Count. The first two constraints are identical to what was described above. Verb and Noun count constraints are enforcing the following rule: a cohesion

Algorithm 5 Rich Interruption Constraints (Coh5)

```
Input: Source tree $T$, previous phrase $\bar{f}_h$, current phrase $\bar{f}_{h+1}$, coverage vector $C_h$
$Interruption \leftarrow False$
$ICount \leftarrow 0$
$VerbCount \leftarrow 0$
$NounCount \leftarrow 0$
$C_{h+1} = C_h \cup \{j | f_j \in \bar{f}_{h+1}\}$
$F \leftarrow$ the left and right-most tokens of $\bar{f}_h$
for each of $f \in F$ do
    Climb the dependency tree from $f$ until you reach the highest node $n$ such that $\bar{f}_{h+1} \notin T(n)$.
    if $n$ exists then
        for each of $e \in T(n)$ and $C_{h+1}(e) = 0$ do
            $Interruption \leftarrow True$
            $ICount = ICount + 1$
            if POS of $e$ is "VB" then
                $VerbCount \leftarrow VerbCount + 1$
            else if POS of $e$ is "NN" then
                $NounCount \leftarrow NounCount + 1$
            end if
        end for
    end if
end for
Return $Interruption$, $ICount$, $VerbCount$, $NounCount$
```

violation will be penalized more in terms of the number of content words that have not been covered. For example, we want to translate the English sentence "the presidential election of the united states begins tomorrow" to French with the dependency structure as in Figure 3.1. We consider $\bar{f}_h$ = "the united states", $\bar{f}_{h+1}$ = "begins". The coverage bit vector C_{h+1} is "0 0 0 0 1 1 1 1 0". Algorithm 5 will return true for $Interruption$, 4 for $ICount$ ("the"; "presidential"; "election"; "of"), 0 for $VerbCount$ and 1 for $NounCount$ ("election").

3.2 Experiments

We built baseline systems using GIZA++ IBM Model 4 (Och and Ney, 2003), Moses' phrase extraction with the grow-diag-final-end heuristic (Koehn et al., 2007), a standard phrase-based decoder (Vogel, 2003), the SRI LM toolkit (Stolcke, 2002), a suffix-array language model (Zhang and Vogel, 2005), a distance-based word reordering model with a window of 3, and the maximum number of target phrases restricted to 10. Results are reported using lowercase BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). All model weights were trained on development sets via minimum-error rate training (MERT) (Venugopal and Vogel, 2005) with 200 unique n-best lists and optimizing toward BLEU. To shorten the training time, a multi-threaded GIZA++ version was used to utilize multi-processor servers (Gao and Vogel, 2008). We used the MALT parser (Nivre et al., 2006)[1] to obtain source English dependency trees and the Stanford parser for Arabic and Chinese (Marneffe et al., 2006). In order to decide whether the translation output of one MT engine is significantly better than another one, we used the bootstrap method (Zhang et al., 2004) with 1000 samples ($p < 0.05$). We performed experiments on English-Iraqi, English-Spanish, Arabic-English and Chinese-English. Detailed corpus statistics are shown in Table 4.3. Table 3.2 shows results in lowercase BLEU and TER.

The first step in validating the proposed approach was to check if it works for the other language pairs. Our English-Iraqi data come from the DARPA TransTac program.

[1] We would like to thank Johan Hall and Joakim Nirve for helpful suggestions on training and using the English dependency model

	English-Iraqi		English-Spanish		Arabic-English		Chinese-English	
	English	Iraqi	English	Spanish	Arabic	English	Chinese	English
sentence pairs	654,556		1,310,127		5,359,543		10,964,230	
unique sent. pairs	510,314		1,287,016		5,111,961		9,041,423	
avg. sentence length	8.4	5.9	27.4	28.6	25.7	29.7	24.9	28.1
# words	5.5 M	3.8 M	35.8 M	37.4 M	138 M	159 M	272.5 M	308.2 M
vocabulary	34 K	109 K	117 K	173 K	690 K	364K	1.4 M	845 K

Table 3.1: Corpus statistics of English-Iraqi, English-Spanish, Arabic-English and Chinese-English systems

	English-Iraqi		English-Spanish		Arabic-English				Chinese-English			
	june08		nct07		mt08-nw		mt08-wb		dev07-nw		dev07-wb	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
Baseline	23.58	61.03	32.04	49.97	48.53	45.03	33.77	56.30	25.14	62.32	23.65	61.66
+Coh1	24.45	58.89	*32.72*	49.18	48.78	44.92	34.15	**56.01**	26.46	61.04	23.95	**61.05**
+Coh2	**24.73**	58.75	*32.81*	49.02	48.47	45.23	**34.20**	56.42	***26.92***	61.24	23.92	61.45
+Coh3	24.19	59.25	*32.87*	48.88	48.70	44.84	33.91	56.29	26.3	61.46	**24.19**	61.51
+Coh4	24.66	**58.68**	*33.20*	48.42	**48.85**	**44.73**	33.86	56.38	26.73	**60.94**	24.03	61.42
+Coh5	24.42	59.05	***33.27***	**48.09**	48.57	45.07	34.10	56.37	26.05	61.69	23.76	61.52

Table 3.2: Scores on held-out evaluation sets of baseline and cohesion-enhanced systems for English-Iraqi, English-Spanish, Arabic-English and Chinese-English language pairs. Bold type is used to indicate highest scores. An italic text indicates the score is statistical significant better than the baseline

The target domain is force protection which includes checkpoints and house searches, and extends to civil affairs, medical, and training dialogs.

We used TransTac T2T July 2007 (july07) as the development set and TransTac T2T June 2008 (june08) as the held-out evaluation set. Each test set has 4 reference translations. We applied the suffix-array LM up to 6-gram with Good-Turing smoothing. In Table 3.2, cohesive soft constraints produced improvements ranging between **0.5** and **1.2** BLEU point on the held-out evaluation set.

We have shown that the proposed cohesion-enhanced decoder outperformed the baseline English-Iraqi systems. This system used a small training size and came from force protection domain. The English-Iraqi pair also differs according to the language family. English is an Indo-European language while Iraqi is a Semitic language of the Afro-Asiatic language family. The next step in validating the proposed approach was to test on a language pair which comes from the same Indo-European language family with a medium training size, different domain and written style.

We used the Europarl and News-Commentary parallel corpora for English-Spanish as provided in the ACL-WMT 2008[2] shared task evaluation. Detailed corpus statistics are given in Table 4.3. We built the baseline system using the parallel corpus with maximum sentence length of 100 words for word alignment and a 4-gram SRI LM with modified Kneyser-Ney smoothing. We used nc-devtest2007(ncd07) as the development set and nc-test2007 (nct07) as the held-out evaluation set. Each test set has 1 translation reference. Table 3.2 shows that we obtained improvements ranging between **0.7** and **1.2** BLEU points. All cohesive soft constraints performed **statistical significant** better than the baseline on the held-out evaluation set.

The previous results indicate that cohesive soft constraints contribute to the improvements of translation systems from English to other languages. However, many of today's high-profile translation tasks are concerned with translation into English. We experimented with the GALE data to test this other direction, and to examine cohesion's effect on state of the art systems, which include other powerful word reordering features, such as large language models.

[2] http://www.statmt.org/wmt08

To validate these questions we present experimental results for the large-scale Arabic-English and Chinese-English systems. Our Arabic-English and Chinese-English data comes from the DARPA GALE program[3] and belongs to the newswire and broadcast news domain. Detailed corpus statistics are shown in Table 4.3. A 5-gram SRI LM was trained from the English Gigaword Corpus V3, which contains several newspapers for the years between 1994 and 2006. We also included the English side of the bilingual training data, resulting in a total of 2.7 billion running words after tokenization. For the Arabic-English system we used NIST MT-06 as the development set and NIST MT-08 NW (mt08-nw) and WB (mt08-wb) as held-out evaluation sets. For the Chinese-English system we used NIST MT-05 as the development set and Dev07Blind NW (dev07-nw) and WB (dev07-wb)[4] as held-out evaluation sets. Each test set has 4 reference translations. Table 3.2 shows results in BLEU and TER. The best improvements in BLEU we obtained are **0.3** on MT-08 NW and **0.4** on MT-08 WB for Arabic-English. We obtained **1.8** BLEU on Dev07Blind NW and **0.5** on Dev07Blind WB for Chinese-English over the baseline. Coh2 performed **statistically significant** better than the baseline system on Dev07Blind NW.

3.3 Discussion and Analysis

Experimental results of cohesive soft constraints on different language pairs have been described in Section 4.2, in this section we vary the ordering capability of the baseline system, and perform other forms of error analysis.

3.3.1 Interactions with reordering models

We first investigate the interactions of cohesive contraints with lexicalized reordering models on the performance of the translation system. The question we are trying to answer is whether the improvements of cohesive soft constraints are subsumed by a strong reordering model. Koehn et al. (2005) proposed the lexicalized reordering model which conditions reordering probabilities on the word of each phrase pair. The lexicalized reordering

[3] This training data was used in GALE P3 Evaluation [4] This test set is distributed by the GALE Rosetta team

model has shown substantial improvements over the distance-based reordering model.

	dev07-nw		dev07-wb	
	BLEU	TER	BLEU	TER
Baseline	25.14	62.32	23.65	61.66
+Lex	26.07	61.56	23.68	61.71
+Lex+Coh1	26.52	62.00	24.47	61.69
+Lex+Coh2	**26.62**	**60.71**	24.95	**60.33**
+Lex+Coh3	26.53	61.62	**25.04**	61.06
+Lex+Coh4	26.53	60.86	24.79	60.69
+Lex+Coh5	26.35	60.74	24.88	60.44

Table 3.3: Performances of the GALE Chinese-English system with lexicalized reordering models compared to cohesion-enhanced systems

Table 3.3 shows the performance of the Chinese-English system on the held-out evaluation set when we include lexicalized reordering models and cohesive soft constraints in the baseline system with a distance-based reordering model[5]. The system with the lexicalized reordering model *+lex* gained over the baseline system by 0.9 BLEU points on dev07-nw set and performed similar on dev07-wb set. However, the performance of *+lex* is still weaker than most cohesive soft constraints in Table 3.2. Furthermore, when cohesive soft constraints are added on top of the lexicalized reordering model we observed a gain by **0.5** BLEU point on dev07-nw and a substantial gain by **1.4** BLEU on dev07-wb set. Coh2 model obtained the best scores in most cases.

After having empirical evidence for the improvements of cohesive soft constraints over systems with lexicalized reordering models, we investigate the impact of the reordering window. Table 3.4 demonstrates the translation performances of systems with different reordering limits and reordering models. The baseline system used a distance-based reordering model with reordering window of 3. Meanwhile, *+lex* and *+lex+w5* used lexicalized

[5] Note that we ran MERT separately for each system

	dev07-nw		dev07-wb	
	BLEU	TER	BLEU	TER
Baseline	25.14	62.32	23.65	61.66
+Lex	26.07	61.56	23.68	61.71
+Lex+w5	26.21	61.06	24.87	60.84
+Lex+w5+Coh1	26.92	60.30	25.27	60.81
+Lex+w5+Coh2	**27.13**	**60.21**	25.12	60.95
+Lex+w5+Coh3	27.09	60.76	25.10	60.79
+Lex+w5+Coh4	26.79	60.50	**25.37**	**60.48**
+Lex+w5+Coh5	26.87	61.04	25.06	61.03

Table 3.4: Performances of the GALE Chinese-English system with lexicalized reordering models and reordering window 5 compared to cohesion-enhanced systems

reordering models with reordering windows of 3 and 5, respectively. *+lex+w5* gained over the *+lex* system by 0.1 BLEU point on dev07-nw and 1.1 BLEU on dev07-wb. However, *+lex+w5* is still weaker than *+lex+Coh2* system in Table 3.3. We add cohesive soft constraints on top of *+lex+w5*. Cohesion-enhanced systems performed better than *+lex+w5* by **0.9** BLEU on dev07-nw and **0.5** BLEU point on dev07-wb.

3.3.2 The decoder behaviors

The cohesive soft constraints essentially act as filters on the generated hypotheses. As longer phrases can induce more cohesion violations, it is interesting to see how big an effect the different cohesive soft constraints have on the selection of phrases used in the final first best translation. The average length of phrases used in the translations is shown in Table 3.5. We see that indeed the cohesion constraints bias toward using shorter phrases.

We also analyzed how often a cohesion violation actually occurs under the different versions. Triple $(\bar{f}_h, \bar{f}_{h+1}, C_{h+1})$ can either trigger a cohesion violation or signal no vio-

	june-08	nc-test2007	mt08-NW	mt08-WB
Baseline	2.3	2.01	1.88	1.54
+Coh1	2.26	1.89	1.81	1.50
+Coh2	2.24	1.92	1.89	1.56
+Coh3	2.26	1.97	1.88	1.54
+Coh4	2.13	2.01	1.87	1.53
+Coh5	2.16	1.89	1.82	1.52

Table 3.5: The average length of phrases used in the translations

	june-08	nc-test2007	mt08-NW	mt08-WB
+Coh1	0.3896	0.4001	0.4786	0.4412
+Coh2	0.4305	0.4547	0.5198	0.4789
+Coh3	0.3887	0.3974	0.4777	0.4404
+Coh4	0.4304	0.4546	0.5198	0.4790
+Coh5	0.3916	0.4003	0.4852	0.4469

Table 3.6: Ratios between the number of times the interruption check fires and the total number of interruptions check in the different variants

lation independent of the actual translation generated. Therefore, we count the number of different triples and how many of them led to a cohesion violation. Results are summarized in Table 3.6. As expected, since Coh 2 and 4 perform exhaustive interruption checks they have higher ratio than the others. The ratios of Coh 1, 3 and 5 are close but not exactly the same because of hypothesis recombination and pruning during the decoding process. This is also true for the Coh 2 and 4.

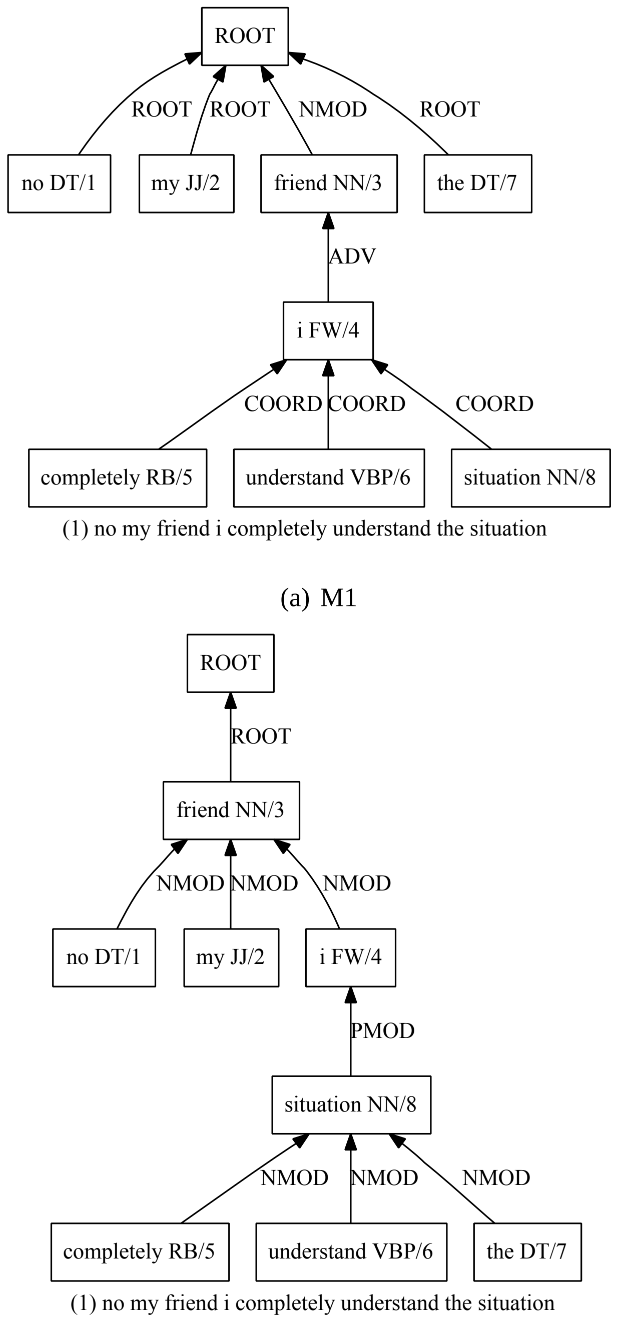
3.3.3 The role of dependency parser

We analyze the influence of the dependency parser on the performance of the translation system. We experimented with the MALT parser and the Stanford parser with default parameters on the English-Iraqi system described in Section 4.2. Performances on the unseen test set are shown in Table 3.7. Experimental results show that 1) either using MALT or Stanford parser the proposed approaches still outperform the baseline; 2) the MALT parser has a tendency to give better BLEU scores than the Stanford parser whereas the Stanford parser is faster than the MALT parser in our experimental setup.

	MALT Parser		Stanford Parser	
	BLEU	TER	BLEU	TER
Baseline	23.58	61.03	23.58	61.03
+ Coh1	24.45	58.89	24.17	58.79
+ Coh2	**24.73**	58.75	24.12	58.83
+ Coh3	24.19	59.25	24.37	58.81
+ Coh4	24.66	**58.68**	24.44	58.71
+ Coh5	24.42	59.05	23.99	59.55

Table 3.7: Comparison between using MALT parser and Stanford parser on English-Iraqi system

A general question of what quality of parser is required for cohesive soft constraints to



(a) M1

(b) M2

Figure 3.3: Dependency trees produced by M1 and M2.

work is important (Quirk and Corston-Oliver, 2006). To answer this question, we trained two MALT parser models, M1 and M2, on different sizes of Penn Treebank V3 data. The performances in term of unlabeled attachment score on the CoNLL-07 dependency test set are 19.41% and 86.21% for M1 and M2, respectively. Figure 3.3 illustrates difference dependency tree structures produced by M1 and M2 models. Table 3.8 shows the comparison of using M1 and M2 for English-Iraqi and English-Spanish systems. The results show that when applying these models to English-Iraqi, M1 performs better than M2 in most cases except Coh4. However, when the models are applied to English-Spanish then M2 is better than M1 in most cases except Coh2. The reason is that M1 and M2 models were only trained on Penn Treebank which belongs to newswire domain. M2's high performance on the newswire data has a positive effect on the Spanish test set, which is also drawn from the newswire domain. Meanwhile, the Iraqi defense text, which is quite different from newswire, seems to have no stable correlation with (newswire) parse quality, with M1 helping in some versions of the cohesion constraint, and M2 helping in others.

	English-Iraqi		English-Spanish	
	M1	M2	M1	M2
Baseline	23.58	23.58	32.04	32.04
+ Coh1	24.16	23.86	31.92	32.29
+ Coh2	24.32	24.30	32.40	32.30
+ Coh3	24.23	24.06	31.89	32.60
+ Coh4	23.86	**24.54**	32.43	32.81
+ Coh5	24.26	24.22	32.53	**33.00**

Table 3.8: The impact of parser quality on the performance of English-Iraqi and English-Spanish systems in BLEU score. The performances in term of unlabeled attachment score on the CoNLL-07 dependency test set are 19.41% and 86.21% for M1 and M2, respectively.

3.4 Summary

In this chapter, we explored cohesive phrasal decoding, focusing on variants of cohesive soft constraints. We proposed four novel cohesive soft constraints namely exhaustive interruption check (Coh2), interruption count (Coh3), exhaustive interruption count (Coh4) and rich interruption constraints (Coh5). Our experimental results show that with cohesive soft constraints the system generates better translations in comparison with strong baselines. To ensure the robustness and effectiveness of the proposed approaches, we conducted experiments on 4 different language pairs, namely English-{Iraqi, Spanish} and {Arabic, Chinese}-English. These experiments also covered a wide range of training corpus sizes, ranging from 500K sentence pairs up to 10 million sentence pairs. Furthermore, the effectiveness of our proposed methods was shown when we applied them to systems using a 2.7 billion words 5-gram LM, different reordering models and dependency parsers. All five approaches give positive results. While the improvements are not statistically significant at the 95% level in most cases, there is nonetheless a consistent pattern indicating that the observed improvements are stable. The most reliable approach seems to be Coh2, a solution which does not make the violation free assumption.

Chapter 4

Source-side Dependency Tree Reordering Models with Subtree Movements and Constraints

In this chapter, to go beyond cohesive soft constraints, we introduce a novel reordering model for phrase-based systems which exploits dependency subtree movements and constraints. In order to do, we must first consider several questions. Should subtree movements be conditioned on source dependency structures? How can we estimate reliable probability distributions from training data? How do we incorporate the reordering model with dependency structures and cohesive soft constraints into a phrase-based decoder? We investigate these questions by presenting the model, training and decoding procedure in Section 4.1. Furthermore, we present experimental results on English-Iraqi and English-Spanish systems in Section 4.2. Finally, we investigate the impact of the proposed models in Section 4.3 .

4.1 Source-tree Reordering Models

Nowadays most statistical machine translation systems are based on log-linear model which tries to provide a parameterized form of the probability of translating a sentence f_1^J to e_1^I. A common feature set includes reordering models which provide the decoder the capability to determine the orientation sequence of phrases. The beam search strategy is used during decoding, in which the intermediate states correspond to partial translations. The decoding process advances by extending a state with the translation of a source phrase and the final state is reached when each source word has been translated exactly once.

Reordering occurs when the source phrase to be translated does not immediately follow the previously translated phrase. The reordering is integrated into the target function by using discriminatively-trained distortion penalties, such as the widely used lexicalized reordering model (Tillman, 2004; Koehn et al., 2005). It can be parameterized as follows:

$$p(O|e,f) = \prod_{i=1}^{n} p(o_i|\bar{e_i}, \bar{f_{a_i}}, a_{i-1}, a_i) \tag{4.1}$$

where **f** is the input sentence; **e** $= (\bar{e_1}, \ldots, \bar{e_n})$ is the target language phrases; **a** $= (a_1, \ldots, a_n)$ is phrase alignments; $\bar{f_{a_i}}$ is a source phrase which has a translated phrase $\bar{e_i}$ defined by an alignment a_i. **O** is the orientation sequence of phrase where each o_i has a value over three possible orientations, (*M*) monotone, (*S*) swap with previous phrase, or (*D*) discontinuous. **O**=$\{M, S, D\}$ and is defined as follows:

$$o_i = \begin{cases} M & \text{if } a_i - a_{i-1} = 1 \\ S & \text{if } a_i - a_{i-1} = -1 \\ D & \text{if } |a_i - a_{i-1}| \neq 1 \end{cases} \tag{4.2}$$

4.1.1 Models

A lexicalized reordering model is defined in terms of transitions between phrases - two phrases in sequence, $previous$ and $next$, have a specific relationship to each other, such as

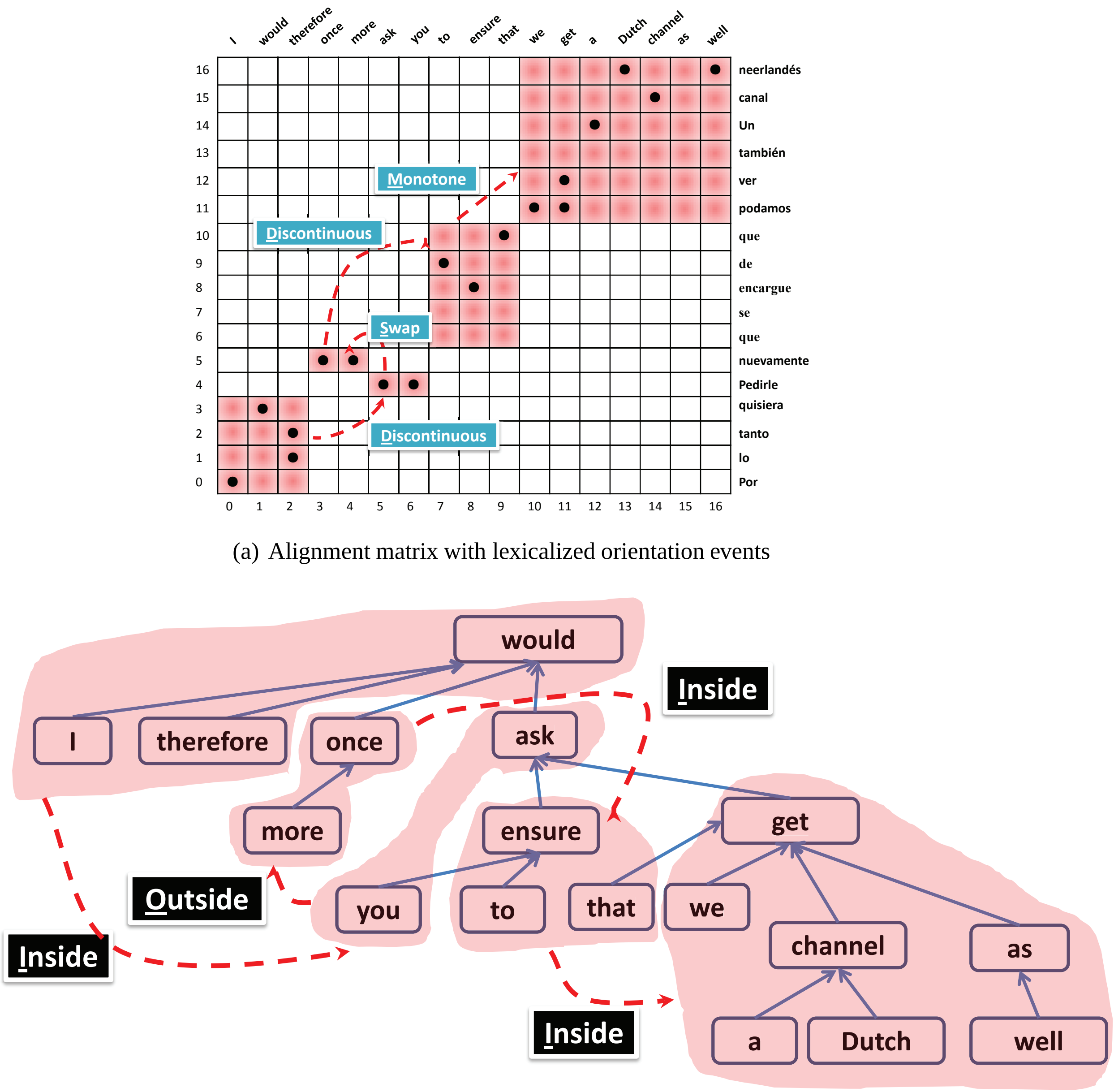
$monotone$, $swap$ or $discontinuous$. Statistics on those relationships make up the model.

Lexicalized reordering models are well-defined for flat word surface structures. However, the models do not leverage source-side syntactic structures which are always available during the decoding time. Previous studies, such as Cherry (2008), show improvements when using source-side dependency structures as cohesive soft constraints. Cohesion constraints tell the decoder which cohesive movements are available, but the decoder has no opinion on the likelihood of these moves.

In a source-tree reordering model, we would condition monolingually and syntactically phrase movements on the source dependency tree. A source-tree reordering model considers in terms of previous source dependency structures. One can think about the phrase movements as the movement of the subtree $inside$ or $outside$ a source subtree when the decoder is leaving from the *previous* source state to the current source state. The notions of moving $inside$ (**I**) and $outside$ (**O**) a subtree can be interpreted as tracking facts about the subtree-to-subtree transitions observed in the source side of word-aligned training data. With extra guidance on subtree movements, our expectation is that source-tree reordering models will help the decoder make smarter distortion decisions.

An example of the source-tree reordering movements is illustrated in Figure 4.1 that contains a word/phrase alignment matrix of a English-Spanish sentence pair, source dependency tree and reordering movements. The lexicalized orientation sequence is $\{$D, S, D, M$\}$ while the subtree movement sequence is $\{$I, O, I, I$\}$. The lexicalized reordering model assigned D for phrase "*ask you*" because the previous extracted phrase "*I would therefore*" was not continuous with "*ask you*". At the same time, the source-tree movement assigned *I* since "*ask you*" is moving $inside$ the subtree rooted at "*would*". In addition, "*once more*" received *O* from the source-tree reordering model since it is $swap$ with "*ask you*" and moving $outside$ the subtree rooted at "*ask*".

Let T denote the source dependency tree and $T(n)$ stands for the subtree rooted at node n. A span $\bar{f}$ indicates the last source phrase translated to create the current state and each $\bar{f}$ has a dependency structure s_h. A subtree $T(n)$ covers a span of contiguous source words is constructed by dependency structures s_h; for subspan $\bar{f}$ covered by $T(n)$, we say $\bar{f} \in T(n)$. We define a subtree that has begun translation but not yet complete, an *open*



(a) Alignment matrix with lexicalized orientation events

(b) Inside/Outside subtree movements on the source dependency tree

Figure 4.1: Source-tree reordering extraction examples for the English-Spanish sentence pair “*I would therefore once more ask you to ensure that we get a Dutch channel as well*”-“*Por lo tanto quisiera pedirle nuevamente que se encargue de que podamos ver tambin un canal neerland*”

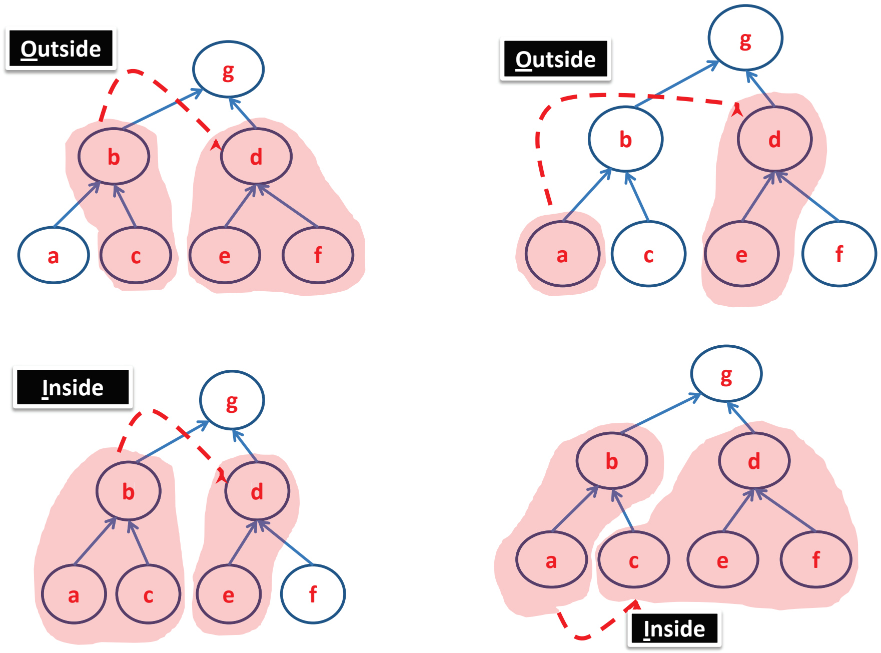


Figure 4.2: Examples of ***inside (I)*** and ***outside (O)*** movements

subtree. On the other hand, when all words under a node have been translated then we call a *completed* subtree. A phrase $\bar{f}$ is moving ***inside (I)*** a $T(n)$ if $\bar{f}$ helps $T(n)$ to be completed, in other words, $T(n)$ covers more contiguous words. A phrase $\bar{f}$ is moving ***outside (O)*** a $T(n)$ if $\bar{f}$ leaves $T(n)$ to be open, in other words, $T(n)$ contains some words which have not been covered yet. $inside$ and $outside$ are the two subtree movements we are going to model and Figure 4.2 shows example movements in different cases.

Mathematically speaking, a source-tree reordering model is defined as follows:

$$p(D|e, f) = \prod_{i=1}^{n} p(d_i|\bar{e}_i, \bar{f}_{a_i}, a_i, s_{i-1}, s_i) \tag{4.3}$$

where s_i and s_{i-1} are dependency structures of source phrases $\bar{f}_{a_i}$ and $\bar{f}_{a_{i-1}}$ respectively; **D** is a random variable which represents the sequence of syntactical phrase movements over the source dependency tree; each d_i takes a value either *inside* (**I**) or *outside* (**O**). $p(D|e, f)$ is the probability of the subtree movement likelihood over the source phrase sequence and their target movements. Since the model essentially constraints phrase movements on the source dependency tree however it does not explicitly provide orientations for a phrase-based decoder. Therefore, we combine our model with the lexicalized reordering model, as a result, a set of events contains $D = o_k\_d_j$ = $\{$*M_I, S_I, D_I, M_O, S_O, D_O*$\}$. The source dependency tree is used here to refine the reordering events provided by a lexicalized reordering model. Finally, the source-tree reordering model is derived as follows:

$$p(D|e,f) = \prod_{i=1}^{n} p((o\_d)_i|\bar{e}_i, \bar{f}_{a_i}, a_{i-1}, a_i, s_{i-1}, s_i) \quad (4.4)$$

4.1.2 Training

To train the model, the system needs to extract $o_k\_d_j$ events for phrase pairs. First, the source side dependency trees of the bilingual training data are provided by using a dependency parser. Given a sentence pair and source dependency tree, when performing the phrase-extract algorithm (Och and Ney, 2004) we also extract the source dependency structure of each phrase pair. The values of o_k are obtained by lexicalized reordering models. To determine whether the current source phrase is moving $inside$ or $outside$ a subtree $T(n)$ with respect to previously extracted phrases we apply the exhaustive interruption check algorithm (Bach et al., 2009b). This algorithm essentially walks through the dependency subtrees of previously extracted phrases and checks whether the subtree is open or completed. The value of d_j is I when the exhaustive interruption check algorithm returns false and O otherwise. Table 4.1 is a snapshot of the output of the reordering extraction procedure. The third column shows source-tree reordering features.

After having all extracted phrase pairs with dependency features, we need to estimate parameters of source-tree reordering models for a particular pair $p((o_j\_d_k)_i|\bar{e}_i, \bar{f}_{a_i})$. An event, such as *M_I*, can be interpreted by three possibilities. First, *M_I* is a joint probability of $monotone$ and $inside$ given a phrase pair. Second, *M_I* can be a conditional probability of $monotone$ given a phrase pair and it is $inside$. Finally, *M_I* can be a conditional probability of $inside$ given a phrase pair and it is $monotone$. The parameter $p((o_j\_d_k)_i|\bar{e}_i, \bar{f}_{a_i})$ is estimated by the maximum likelihood estimation criteria with a smoothing factor γ as

Phrase pairs	Lexicalized	Source-tree
...		
ask you # pedirle	dis swap	D_I $*$
ask you # pedirle	mono mono	M_I
ask you # pedirle	mono mono	M_O
once more # nuevamente	swap dis	S_O $*$
once more # nuevamente	dis swap	D_O
once more # nuevamente que	swap dis	S_O
...		

Table 4.1: Extracted reordering events; $*$ indicates events extracted from the example in Figure 4.1

$$p((o_j\_d_k)_i|\bar{e}_i, \bar{f}_{a_i}, o_j, d_k) = \frac{count(o_k\_d_j) + \gamma}{\sum_k \sum_j (count(o_k\_d_j) + \gamma)} \tag{4.5}$$

if it is a joint probability of subtree movements and lexicalized orientations (*DO*) or

$$p((o_j\_d_k)_i|\bar{e}_i, \bar{f}_{a_i}, d_k) = \frac{count(o_k\_d_j) + \gamma}{\sum_k (count(o_k\_d_j) + \gamma)} \tag{4.6}$$

if it is conditioned on subtree movements (*DOD*) or

$$p((o_j\_d_k)_i|\bar{e}_i, \bar{f}_{a_i}, o_j) = \frac{count(o_k\_d_j) + \gamma}{\sum_j (count(o_k\_d_j) + \gamma)} \tag{4.7}$$

if it is conditioned on lexicalized orientations (*DOO*).

Table 4.2 displays source-tree reordering estimated probabilities for a phrase pair “*ask you*”-“*pedirle*”. Each probability was put under one of the three parameter estimation methods.

	M_I	*S_I*	*D_I*	*M_O*	*S_O*	*D_O*
DO	0.691	0.003	0.142	0.119	0.009	0.038
DOD	0.827	0.003	0.170	0.719	0.053	0.228
DOO	0.854	0.250	0.790	0.146	0.750	0.210

Table 4.2: $inside$ and $outside$ probabilities for phrase "*ask you*"- "*pedirle*" according to three parameter estimation methods

4.1.3 Decoding

The beam search strategy is unchanged from the phrase-based system. Our proposed source-tree reordering models concern mono-lingually and syntactically movements in the source sentence. However, computing source-tree reordering model scores can be done in two scenarios 1) not using and 2) using cohesive soft constraints. Cohesive soft constraints can be enforced by the interruption check algorithm (Cherry, 2008; Bach et al., 2009b). One can consider the first scenario as the decoder does not have any information about the source dependency tree during decoding time, therefore, we allow the decoder to consider both events $inside$ and $outside$. The decision of selecting a preferable feature is made by the tuning procedure. On the other hand, when the source dependency tree is available, subtree movements are informed to the decoder via cohesive soft constraints, as a result, we are able to allow the decoder to make a harder choice to consider either $inside$ or $outside$.

More specifically, if the decoder chooses to decode without cohesive soft constraints then after detecting the orientation of the current phrase, for example $swap$, the decoder will trigger two subtree movement features *S_I* and *S_O* and sum up both features in the log-linear combination. In other words, the decoder considers both events that the current phrase is moving $inside$ and $outside$ a subtree $T(n)$ given it is $swap$ orientation on flat word structures.

In the second scenario, the decoder uses cohesive soft constraints after detecting the orientation of the current phrase, for example $swap$. The decoder only considers one source-tree reordering feature. The choice of feature depends on the output of the interruption check algorithm on the current phrase. If the return is $inside$ then *S_I* will be used otherwise *S_O*.

4.2 Experimental Results

We built baseline systems using GIZA++ Och and Ney (2003), Moses' phrase extraction with the grow-diag-final-and heuristic Koehn et al. (2007), a standard phrase-based decoder Vogel (2003), the SRI LM toolkit Stolcke (2002), the suffix-array language model Zhang and Vogel (2005), a lexicalized reordering model with a reordering window of 3, and the maximum number of target phrases restricted to 5. Results are reported using lowercase BLEU Papineni et al. (2002) and TER Snover et al. (2006). All model weights were trained on development sets via minimum-error rate training (MERT) Venugopal and Vogel (2005) with an unique 200-best list and optimizing toward BLEU. To shorten the training time, a multi-threaded GIZA++ version was used to utilize multi-processor servers Gao and Vogel (2008). We used the MALT parser Nivre et al. (2006) to get English dependency trees. We perform experiments on English→Spanish and English→Iraqi tasks. Detailed corpus statistics are shown in Table 4.3.

	English→Spanish		English→Iraqi	
	English	Spanish	English	Iraqi
sent. pairs	1,310,127		654,556	
uniq. pairs	1,287,016		510,314	
avg. sent. length	27.4	28.6	8.4	5.9
# words	35.8 M	37.4 M	5.5 M	3.8 M
vocabulary	117 K	173 K	34 K	109 K

Table 4.3: Corpus statistics of English→Spanish and English→Iraqi systems

We experiment systems in different configurations of the source-tree reordering model such as DO, DOD and DOO means parameters estimation using Equation 4.5, 4.6 and 4.7 respectively. Moreover, Coh means the decoder triggers cohesive constraints for source-tree reordering models Cherry (2008). Bold type is used to indicate highest scores.

Our first step in validating the proposed approach is to check with the English→Spanish system. We used the Europarl and News-Commentary parallel corpora for English→Spanish as provided in the ACL-WMT 2008[1] shared task evaluation. We built the baseline system using the parallel corpus restricting sentence length to 100 words for word alignment and a 4-gram SRI LM with modified Kneyser-Ney smoothing. We used nc-devtest2007(ncd07) as the development set; nc-test2007 (nct07) as in-domain and newstest2008 (net08) as out-domain held-out evaluation sets. Each test set has 1 translation reference. Table 4.4 shows that the best obtained improvements are **+0.8** BLEU point and **-1.4** TER score on the held-out evaluation sets. Moreover, the proposed methods also obtained improvements on the out-domain test set (net08).

	nct07		net08	
	BLEU	TER	BLEU	TER
Baseline	32.89	65.25	20.11	83.09
Coh	33.33	64.72	19.80	82.84
DO	32.99	65.05	20.27	82.65
DO+Coh	33.28	64.77	20.61	82.35
DOD	33.17	64.54	20.33	82.12
DOD+Coh	33.46	64.41	20.58	82.05
DOO	33.10	64.51	20.51	82.12
DOO+Coh	**33.67**	**64.03**	**20.71**	**81.70**

Table 4.4: Scores of baseline and improved baseline systems with source-tree reordering models on English→Spanish

We also validated the proposed approach on English→Iraqi. However, we have a smaller training corpus which comes from force protection domains and is spoken lan-

[1] http://www.statmt.org/wmt08

	june08		nov08	
	BLEU	TER	BLEU	TER
Baseline	25.18	56.70	18.40	62.91
Coh	25.34	57.30	18.01	61.52
DO	25.31	57.30	18.43	60.98
DO+Coh	25.53	57.20	19.13	61.45
DOD	25.34	57.53	18.90	61.81
DOD+Coh	25.50	**56.29**	**19.15**	60.93
DOO	25.25	56.76	18.40	**60.64**
DOO+Coh	**25.58**	56.37	18.59	61.45

Table 4.5: Scores of baseline and improved baseline systems with source-tree reordering models on English→Iraqi

guage style. This data is used in the DARPA TransTac program. The English→Iraqi pair also differs according to the language family. English is an Indo-European language while Iraqi is a Semitic language of the Afro-Asiatic language family.

We used 429 sentences of TransTac T2T July 2007 (july07) as the development set; 656 sentences of TransTac T2T June 2008 (june08) and 618 sentences of November 2008 (nov08) as the held-out evaluation sets. Each test set has 4 reference translations. We used a suffix-array LM up to 6-gram with Good-Turing smoothing. In Table 4.5, source-tree reordering models produced the best improvements of **+0.8** BLEU point and **-2.3** TER score on the held-out evaluation sets.

4.3 Discussion and Analysis

In this section we perform detail error analysis from where different scenarios emerge and questions arise for our assumptions.

		En-Ir				En-Es			
		jun08		nov08		nc07		nt08	
System		BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
Baseline	*tail*	29.45	76.50	24.41	87.69	23.36	92.93	24.41	134.04
	mid	38.61	53.60	35.89	61.07	31.08	66.75	22.61	86.32
	head	61.38	25.80	60.90	28.16	44.58	47.45	35.34	59.54
Coh	*tail*	**+0.56**	**+1.35**	**+1.29**	**+5.27**	**+0.67**	**+1.80**	**+0.07**	**+1.27**
	mid	**+0.14**	-0.91	**+0.48**	**+1.08**	**+0.22**	**+0.07**	-0.02	-0.19
	head	**+0.37**	-1.69	-3.11	-4.68	-0.17	-0.73	-0.48	**+1.27**
DO	*tail*	**+0.28**	**+0.66**	**+1.91**	**+7.03**	**+0.49**	**+1.94**	**+0.87**	**+2.32**
	mid	**+0.07**	-1.15	**+0.58**	**+1.44**	**+0.24**	**+0.45**	**+0.12**	**+0.28**
	head	-0.28	-2.48	-1.31	-3.07	-0.28	-0.71	-0.11	-0.77
DO+Coh	*tail*	**+1.07**	**+1.95**	**+1.72**	**+5.19**	**+0.66**	**+1.78**	**+0.52**	**+1.60**
	mid	**+0.80**	-0.85	**+0.92**	**+1.32**	**+0.19**	**+0.21**	**+0.13**	**+0.25**
	head	-0.37	-2.41	-1.59	-3.62	-0.25	-0.75	-0.01	-1.11
DOD	*tail*	**+0.46**	**+0.06**	**+1.96**	**+4.84**	**+0.35**	**+1.91**	**+0.75**	**+2.84**
	mid	**+0.53**	-1.35	**+0.43**	**+0.29**	**+0.01**	-0.15	**+0.05**	**+0.41**
	head	**+0.27**	-1.03	-0.61	-2.33	-0.79	-1.33	-0.37	-1.37
DOD+Coh	*tail*	**+1.19**	**+2.70**	**+2.10**	**+5.89**	**+0.49**	**+0.43**	**+0.27**	**+1.30**
	mid	**+0.44**	-0.37	**+0.42**	**+1.16**	**+0.01**	-0.85	**+0.12**	**+0.99**
	head	**+0.32**	-1.25	-0.66	-2.02	-0.37	-1.35	-0.26	-2.05
DOO	*tail*	**+1.18**	**+2.41**	**+2.37**	**+7.36**	**+0.35**	**+1.92**	**+0.59**	**+0.39**
	mid	**+0.13**	-0.62	**+0.28**	**+1.83**	**+0.01**	-0.15	**+0.06**	-0.38
	head	-0.50	-2.13	-0.58	-2.63	-0.79	-1.34	-0.47	-1.52
DOO+Coh	*tail*	**+1.28**	**+2.70**	**+2.03**	**+5.88**	**+0.65**	**+1.61**	**+0.69**	**+1.10**
	mid	**+0.74**	-0.52	**+0.19**	**+0.82**	**+0.18**	-0.02	**+0.12**	-0.05
	head	**+0.22**	-1.02	-1.61	-4.16	-0.40	-1.07	-0.22	-1.00

Table 4.6: Distribution of improvements over different portions of the test sets, where for TER the sign is reversed so that positive numbers means improve in TER, i.e., lower TER score. The improvements are marked by bold text.

4.3.1 Breakdown improvement analysis

As we can see from the results, there are improvements on all the different test sets. However, one could expect that the methods may work for a portion of the data but not others. We divide the test sets into three portions based on sentence-level TER of the baseline system. Let μ and σ be the mean and standard deviation of the sentence-level TER of the whole test set. We define three subsets $head$, $tail$ and mid as the sentence whose TER score is lower than $\mu - \frac{1}{2}\sigma$, higher than $\mu + \frac{1}{2}\sigma$ and the rest, respectively. We then fix the division of the three subsets, and calculate the BLEU and TER scores on them for every system. From Table 4.6, the proposed methods tend to output better TER and BLEU for the $tail$ subsets, the improvements on the mid subsets are smaller, and loss can be observed on the $head$ subsets. The splitting of different sets also reflects on the length of sentences, as shown in Table 4.7, the tail parts are generally long sentences. The breakdown analysis suggests a more subtle model taking into account the sentence lengths could bring in more improvements, especially, on the $tail$ set in which the baseline model loses.

	jun08	nov08	nc07	nt08
$head$	7.92	6.27	20.39	13.07
mid	12.31	11.09	28.07	22.78
$tail$	13.91	14.08	35.29	25.33

Table 4.7: Average reference lengths

4.3.2 Interactions of reordering models

To further investigate the impact of the proposed models, we perform several analyses to examine whether there are significant differences in 1) the average phrase length that the decoder outputs; 2) the total number of reorderings occurred in the hypothesis and 3) the average reordering distance for all the reordering events. Table 4.8 shows the statistics on the four aspects for all the test sets. For the average phrase length, we can observe a smaller value when applying the proposed models on English-Spanish tasks. However, on

	Number of Reorderings				Frequency of Reordering				Average Phrase Length				Average Reordering Distance			
	En-Es		En-Ir		En-Es		En-Ir		En-Es		En-Ir		En-Es		En-Ir	
	nc07	nt08	jun08	nov08	nc07	nt08	jun08	nov08	nc07	nt08	jun08	nov08	nc07	nt08	jun08	nov08
Baseline	*1507*	**1684**	*39*	*24*	16.3	16.4	119	164	2.02	1.80	2.20	2.34	2.61	2.44	2.79	2.17
Coh	2045	**2903**	46	21	10.0	12.8	99	178	1.90	1.71	2.25	2.48	2.67	2.58	2.81	2.50
DO	**2189**	2113	97	58	11.6	13.4	47	64	1.95	1.76	2.25	2.47	2.57	2.46	2.88	3.05
DO+Coh	1929	1900	155	88	13.6	15.3	30	44	1.89	1.71	2.17	2.37	2.47	2.33	2.74	2.88
DOD	1735	2592	123	60	14.9	10.7	38	65	1.92	1.88	2.17	2.36	2.73	2.57	2.79	2.93
DOD+Coh	2070	2021	148	**90**	12.8	14.5	32	43	1.88	1.70	2.18	2.37	2.50	2.39	2.64	2.81
DOO	1735	1785	164	49	14.9	16.1	30	79	1.92	1.73	2.10	2.37	2.73	2.60	2.72	2.98
DOO+Coh	1818	1959	**247**	66	14.1	14.6	19	59	1.93	1.74	2.15	2.37	2.53	2.42	2.64	2.88

Table 4.8: Statistics on four aspects of the final hypothesis over different systems; 1. the number of reorderings, 2. the number of words in the hypotheses divided by the number of reordering, i.e. a larger number means more sparse observation of reorderings, 3. the average phrase length and, 4. the average reordering distance

English-Iraqi the picture is contradicting when on one set the phrase length is generally longer and on the other set both longer and shorter statistics can be observed in different systems. Generally, there is no evidence to support a claim that the proposed models have consistent impact on the length of phrases chosen by the decoder. The observation is not surprising since the proposed reordering models are more likely to affect the decoder's behavior on reorderings.

When analyzing the average reordering distance, a more consistent picture can be discovered. The average reordering distance is larger than the corresponding systems with only inside/outside subtree movements. Whereas we cannot observe similar phenomenon comparing the system with only cohesive constraints and the baseline, which indicates that the cohesive constraints actually have the effect of restricting long distance reorder generated by the inside/outside subtree movements. The most interesting observation is the *number of reorderings* in the hypothesis. To make it easier to think about how sparse the reordering events are, we present the occurrence rate of reorderings, i.e. the number of words divided by the number of reorderings, as listed in the parentheses inside Table 4.8. An interesting phenomenon is that in English-Iraqi tasks, the output is generally monotone in the baseline, and the number of reorderings increases dramatically by applying the inside/outside subtree movements. However, solely applying cohesive constraints does not increase the number of reorderings. In English-Spanish tasks, although all the features generate more reordering events than the baseline, applying only the cohesion constraints also increases the number of reorderings dramatically.

When combining the statistics of Table 4.8 the most significant effect the source-tree reordering models contribute is the number of reorderings. Instead of constraining the reordering, the models enable more reorderings to be generated. As shown in Table 4.10, in the training data there are generally more reorderings than we observed in the decoding results. It indicates the baseline reordering model is not subtle enough to encode accurately information in a more generalized way, so that more reorderings can be generated without losing performance. The source-tree reordering models provide a more discriminative mechanism to estimate reordering events. For example, in Table 4.10 the probability mass of monotone and discontinuous events are different given that the phrase is encoded with inside or outside subtree movements. Moreover, the reordering issue is more language-specific than general translation models, and the conditions for a reordering event to happen vary among languages. Providing more features that are conditioned on different information, such as include inside/outside subtree movements and cohesive constraints, could benefit the system performance by enabling MERT to choose the most prominent ones from a larger basis.

4.3.3 The effect of inside/outside events

All the analysis above inspired us to carry out a more direct analysis of the decoder behaviors. As the main motivation of the proposed approach is to model the behavior of $inside/outside$ subtree events, natural assumptions could be that

- different target languages should have different probabilities of generating a sequence that has outside subtree events on the same source language and
- whether the model could change the behavior of generating outside subtree events.
- Furthermore, comparing to baseline system, do the changes, i.e. generating more or less outside subtree events than baseline, bring improvements to those sentences?

From Table 4.9, the number of sentences having outside subtree events has not changed much when decoding with subtree movement features in English-Spanish tasks, while this number generally increases in English-Iraqi tasks. Moreover, when decoding with both subtree movements and cohesive constraints, we observe that the number of sentences having outside subtree events sharply decreases, whereas it increases in English-Iraqi. This result shows an interesting correlation with the performance improvements in Table 4.4 and 4.5, where the systems with cohesive constraints generally outperform those without. If we consider the cohesive constraints as hard constraints, then the outside subtree events are considered as violations, however in English-Iraqi tasks, the performance becomes better with more “violations”. The observation further consolidates our suggestion that subtle models should be preferred for future developments, because the features may encode the

	En-Es		En-Ir	
	nc07	nt08	jun08	nov08
Baseline	29.35	38.52	9.30	9.39
Coh	20.23	29.40	8.23	8.90
DO	30.34	32.57	12.35	11.65
DO+Coh	12.26	13.07	15.40	13.11
DOD	32.39	37.64	12.65	11.00
DOD+Coh	15.94	23.99	11.89	11.97
DOO	28.75	32.08	12.35	11.65
DOO+Coh	18.44	25.50	16.77	10.68

Table 4.9: The percentage of sentences having $outside$ subtree events

information that the violation of constraints is actually preferred, no matter whether it is because of the nature of the particular language or the style of the source (spoken, written, etc.).

	M_I	*S_I*	*D_I*	*M_O*	*S_O*	*D_O*
En-Es	0.38	0.01	0.14	0.3	0.01	0.15
En-Ir	0.62	0.01	0.13	0.16	0.01	0.07

Table 4.10: Distributions of the six source-tree reordering events estimated from English-Spanish and English-Iraqi training data

Table 4.10 displays the overall event distributions of source-tree reordering models. It appears clearly that occurrences of S_I and S_O are too sparsely seen in the training data which assigns nearly 98% of its probability mass to other events. The table strongly suggests that from training data the source-tree reordering models observed $monotone$ and $inside$ movements more often than other categories. Finally, it shows that the proposed reordering model provides a more fine-grained reordering events for phrase-based MT in comparison with the lexicalized reordering model.

4.4 Summary

In this chapter, our major contribution is a novel source-tree reordering model that exploits dependency subtree movements and constraints. These movements and constraints enable us to efficiently capture the subtree-to-subtree transitions observed both in the source of word-aligned training data and in decoding time. Representing subtree movements as features allows MERT to train the corresponding weights for these features relative to others in the model. We show that this model provides improvements for four held-out evaluation sets and for two language pairs.

Chapter 5

Measuring Machine Translation Confidence with Source-Target Dependency Structures

Past research mainly focused on incorporating dependency structures into decoder and reordering models. We have made significant progress towards producing user-acceptable translation output in some language pairs. However, there is still no efficient way for MT systems to inform users which words are likely translated correctly and how confident it is about the whole sentence. In this chapter, we propose a novel framework to predict word-level and sentence-level MT errors with a large number of novel features. Experimental results show that the MT error prediction accuracy is increased from **69.1** to **72.2** in F-score. The Pearson correlation between the proposed confidence measure and the human-targeted translation edit rate (HTER) is **0.6**. Improvements between **0.4** and **0.9** TER reduction are obtained with the n-best list reranking task using the proposed confidence measure. Also, we present a visualization prototype of MT errors at the word and sentence levels with the objective to improve post-editor productivity.

5.1 Motivation

State-of-the-art Machine Translation (MT) systems are making progress to generate more usable translation outputs. In particular, statistical machine translation systems (Koehn et al., 2007; Bach et al., 2007; Shen et al., 2008) have advanced to a state that the translation quality for certain language pairs (e.g. Spanish-English, French-English, Iraqi-English) in certain domains (e.g. broadcasting news, force-protection, travel) is acceptable to users.

However, a remaining open question is how to predict confidence scores for machine translated words and sentences. An MT system typically returns the best translation candidate from its search space, but still has no reliable way to inform users which word is likely to be correctly translated and how confident it is about the whole sentence. Such information is vital to realize the utility of machine translation in many areas. For example, a post-editor would like to quickly identify which sentences might be incorrectly translated and in need of correction. Other areas, such as cross-lingual question-answering, information extraction and retrieval, can also benefit from the confidence scores of MT output. Finally, even MT systems can leverage such information to do n-best list reranking, discriminative phrase table and rule filtering, and constraint decoding (Hildebrand and Vogel, 2008).

Numerous attempts have been made to tackle the confidence estimation problem. The work of Blatz et al. (2004) is perhaps the best known study of sentence and word level features and their impact on translation error prediction. Along this line of research, improvements can be obtained by incorporating more features as shown in (Quirk, 2004; Sanchis et al., 2007; Raybaud et al., 2009; Specia et al., 2009). Soricut and Echihabi (2010) developed regression models which are used to predict the expected BLEU score of a given translation hypothesis. Improvement also can be obtained by using target part-of-speech and null dependency link in a MaxEnt classifier (Xiong et al., 2010). Ueffing and Ney (2007) introduced word posterior probabilities (WPP) features and applied them in the n-best list reranking. From the usability point of view, back-translation is a tool to help users to assess the accuracy level of MT output (Bach et al., 2007). Literally, it translates backward the MT output into the source language to see whether the output of backward translation matches the original source sentence.

However, previous studies had a few shortcomings. First, source-side features were not extensively investigated. Blatz et al. (2004) only investigated source n-gram frequency statistics and source language model features, while other work mainly focused on target side features. Second, previous work attempted to incorporate more features but faced scalability issues, i.e., to train many features we need many training examples and to train discriminatively we need to search through all possible translations of each training example. Another issue of previous work was that they are all trained with BLEU/TER score computing against the translation references which is different from predicting the human-targeted translation edit rate (HTER) which is crucial in post-editing applications (Snover et al., 2006; Papineni et al., 2002). Finally, the back-translation approach faces a serious issue when forward and backward translation models are symmetric. In this case, back-translation will not be very informative to indicate forward translation quality.

In this chapter, we predict error types of each word in the MT output with a confidence score, extend it to the sentence level, then apply it to n-best list reranking task to improve MT quality, and finally design a visualization prototype. We try to answer the following questions:

- Can we use structure and context feature such as dependency structures, source-side information, and alignment context to improve error prediction performance?
- Can we predict more translation error types i.e substitution, insertion, deletion and shift?
- How good do our prediction methods correlate with human correction?
- Do confidence measures help the MT system to select a better translation?
- How confidence score can be presented to improve end-user perception?

In Section 5.2, we describe the models and training method for the classifier. We describe novel features including dependency structures, source-side, and alignment context in Section 5.3. Experimental results and analysis are reported in Section 5.4. Section 5.5 and 5.6 present applications of confidence scores.

5.2 Confidence Measure Model

5.2.1 Problem setting

Confidence estimation can be viewed as a sequential labeling task in which the word sequence is MT output and word labels can be Bad / $Good$ or $Insertion$ / $Substitution$ / $Shift$ / $Good$. We first estimate each individual word confidence and extend it to the whole sentence. Arabic text is fed into an Arabic-English SMT system and the English translation outputs are corrected by humans in two phases. In phase one, a bilingual speaker corrects the MT system translation output. In phase two, another bilingual speaker does quality checking for the correction done in phase one. If bad corrections were spotted, they correct them again. In this chapter we use the final correction data from phase two as the reference thus HTER can be used as an evaluation metric. We have 75 thousand sentences with 2.4 million words in total from the human correction process described above.

We obtain training labels for each word by performing TER alignment between MT output and the phase-two human correction. From TER alignments we observed that out of total errors are 48% substitution, 28% deletion, 13% shift, and 11% insertion errors. Based on the alignment, each word produced by the MT system has a label: good, insertion, substitution and shift. Since a deletion error occurs when it only appears in the reference translation, not in the MT output, our model will not predict deletion errors in the MT output.

5.2.2 Word-level model

In our problem, a training instance is a word from MT output, and its label when the MT sentence is aligned with the human correction. Given a training instance x, y is the true label of x; f stands for its feature vector $f(x, y)$; and w is feature weight vector. We define a feature-rich classifier $score(x, y)$ as follow

$$score(x, y) = w.f(x, y) \tag{5.1}$$

To obtain the label, we choose the class with the highest score as the predicted label for that data instance. To learn optimized weights, we use the Margin Infused Relaxed Algorithm or MIRA (Crammer and Singer, 2003; McDonald et al., 2005) which is an online learner closely related to both the support vector machine and perceptron learning framework. MIRA has been shown to provide state-of-the-art performance for sequential labeling task (Rozenfeld et al., 2006), and is also able to provide an efficient mechanism to train and optimize MT systems with lots of features (Watanabe et al., 2007; Chiang et al., 2009). In general, weights are updated at each step time t according to the following rule:

$$\begin{aligned} w_{t+1} &= \arg\min_{w_{t+1}} ||w_{t+1} - w_t|| \\ \text{s.t.} \quad score(x, y) &\geq score(x, y') + L(y, y') \end{aligned} \tag{5.2}$$

where $L(y, y')$ is a measure of the loss of using y' instead of the true label y. In this problem $L(y, y')$ is 0-1 loss function. More specifically, for each instance x_i in the training data at a time t we find the label with the highest score:

$$y' = \arg\max_{y} score(x_i, y) \tag{5.3}$$

the weight vector is updated as follow

$$w_{t+1} = w_t + \tau(f(x_i, y) - f(x_i, y')) \tag{5.4}$$

τ can be interpreted as a step size; when τ is a large number we want to update our weights aggressively, otherwise weights are updated conservatively.

$$\begin{aligned} \tau &= \max(0, \alpha) \\ \alpha &= \min\left\{C, \frac{L(y, y') - (score(x_i, y) - score(x_i, y'))}{||f(x_i, y) - f(x_i, y')||_2^2}\right\} \end{aligned} \tag{5.5}$$

where C is a positive constant used to cap the maximum possible value of τ. In practice, a cut-off threshold n is the parameter which decides the number of features kept (whose occurrence is at least n) during training. Note that MIRA is sensitive to constant C, the cut-off feature threshold n, and the number of iterations. The final weight is typically normalized by the number of training iterations and the number of training instances. These parameters are tuned on a development set.

5.2.3 Sentence-level model

Given the feature sets and optimized weights, we use the Viterbi algorithm to find the best label sequence. To estimate the confidence of a sentence S we rely on the information from the forward-backward inference. One approach is to directly use the conditional probabilities of the whole sequence. However, this quantity is the confidence measure for the label sequence predicted by the classifier and it does not represent the goodness of the whole MT output. Another more appropriated method is to use the marginal probability of $Good$ label which can be defined as follow:

$$p(y_i = Good|S) = \frac{\alpha(y_i|S)\beta(y_i|S)}{\sum_j \alpha(y_j|S)\beta(y_j|S)} \tag{5.6}$$

$p(y_i = Good|S)$ is the marginal probability of label $Good$ at position i given the MT output sentence S. $\alpha(y_i|S)$ and $\beta(y_i|S)$ are forward and backward values. Our confidence estimation for a sentence S of k words is defined as follow:

$$Goodness(S) = \frac{\sum_{i=1}^{k} p(y_i = Good|S)}{k} \tag{5.7}$$

$Goodness(S)$ is ranging between 0 and 1, where 0 is equivalent to an absolutely wrong translation and 1 is a perfect translation. Essentially, $Goodness(S)$ is the arithmetic mean which represents the goodness of translation per word in the whole sentence.

5.3 Confidence Measure Features

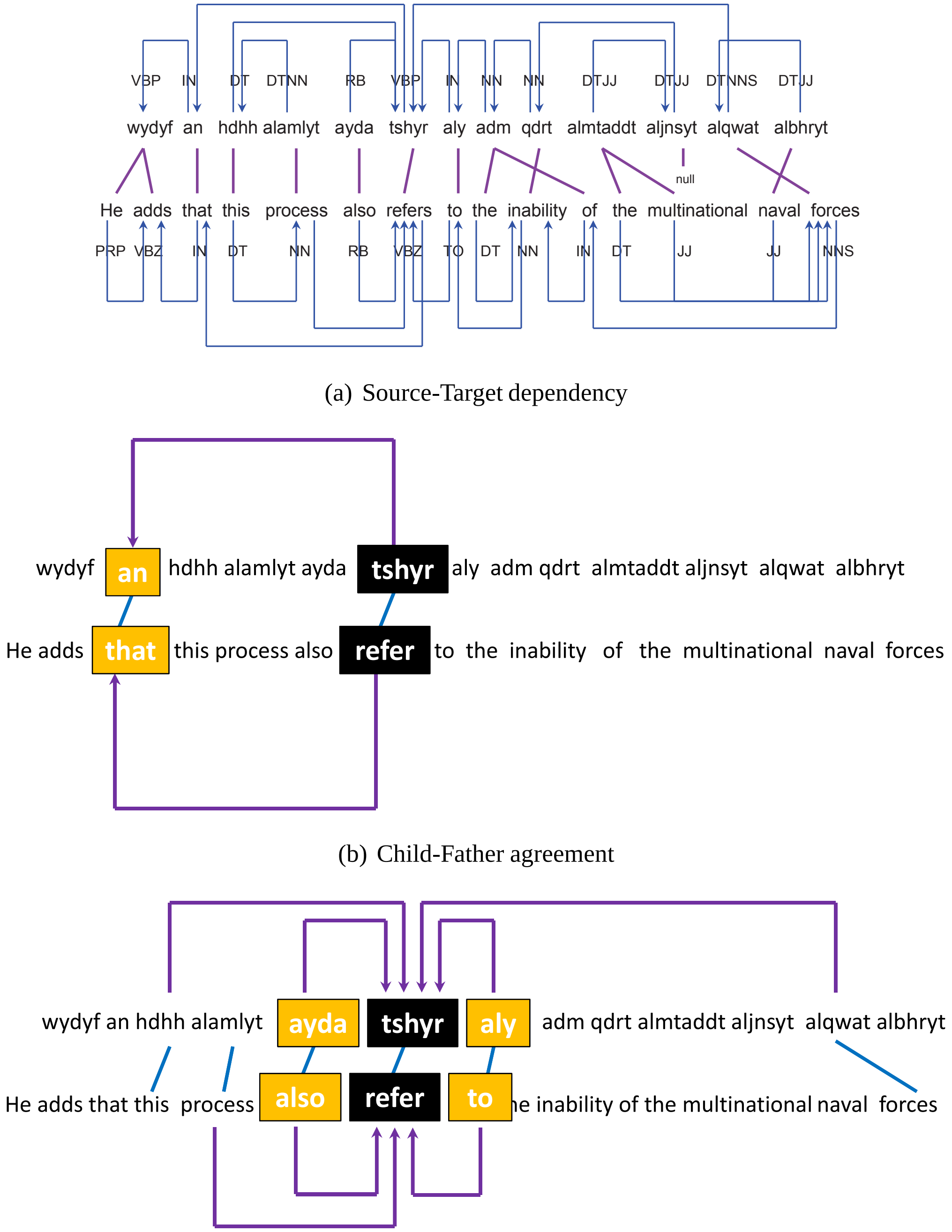
Features are generated from feature types: abstract templates from which specific features are instantiated. Features sets are often parameterized in various ways. In this section, we describe three new feature sets introduced on top of our baseline classifier which has WPP and target POS features (Ueffing and Ney, 2007; Xiong et al., 2010).

5.3.1 Source and target dependency structure features

Dependency structures have been extensively used in various translation systems (Shen et al., 2008; Ma et al., 2008; Bach et al., 2009a). The adoption of dependency structures might enable the classifier to utilize deep structures to predict translation errors. Source and target structures are unlikely to be isomorphic as shown in Figure 5.1(a). However, we expect some high-level linguistic structures are likely to transfer across certain language pairs. For example, prepositional phrases (PP) in Arabic and English are similar in a sense that PPs generally appear at the end of the sentence (after all the verbal arguments) and to a lesser extent at its beginning (Habash and Hu, 2009). We use the Stanford parser to obtain dependency trees and POS tags (Marneffe et al., 2006).

Child-Father agreement: The motivation is to take advantage of the long distance dependency relations between source and target words. Given an alignment between a source word s_i and a target word t_j. A child-father agreement exists when s_k is aligned to t_l, where s_k and t_l are father of s_i and t_j in source and target dependency trees, respectively. Figure 5.1(b) illustrates that "*tshyr*" and "*refers*" have a child-father agreement. To verify our intuition, we analysed 243K words of manual aligned Arabic-English bitext. We observed 29.2% words having child-father agreements. In term of structure types, we found 27.2% of copula verb and 30.2% prepositional structures, including object of a preposition, prepositional modifier, and prepositional complement, are having child-father agreements.

Children agreement: In the child-father agreement feature we look up in the dependency


VBP IN DT DTNN RB VBP IN NN NN DTJJ DTJJ DTNNS DTJJ
wydyf an hdhh alamlyt ayda tshyr aly adm qdrt almtaddt aljnsyt alqwat albhryt
null
He adds that this process also refers to the inability of the multinational naval forces
PRP VBZ IN DT NN RB VBZ TO DT NN IN DT JJ JJ NNS
wydyf an hdhh alamlyt ayda tshyr aly adm qdrt almtaddt aljnsyt alqwat albhryt
He adds that this process also refer to the inability of the multinational naval forces
wydyf an hdhh alamlyt ayda tshyr aly adm qdrt almtaddt aljnsyt alqwat albhryt
He adds that this process also refer to he inability of the multinational naval forces

(a) Source-Target dependency

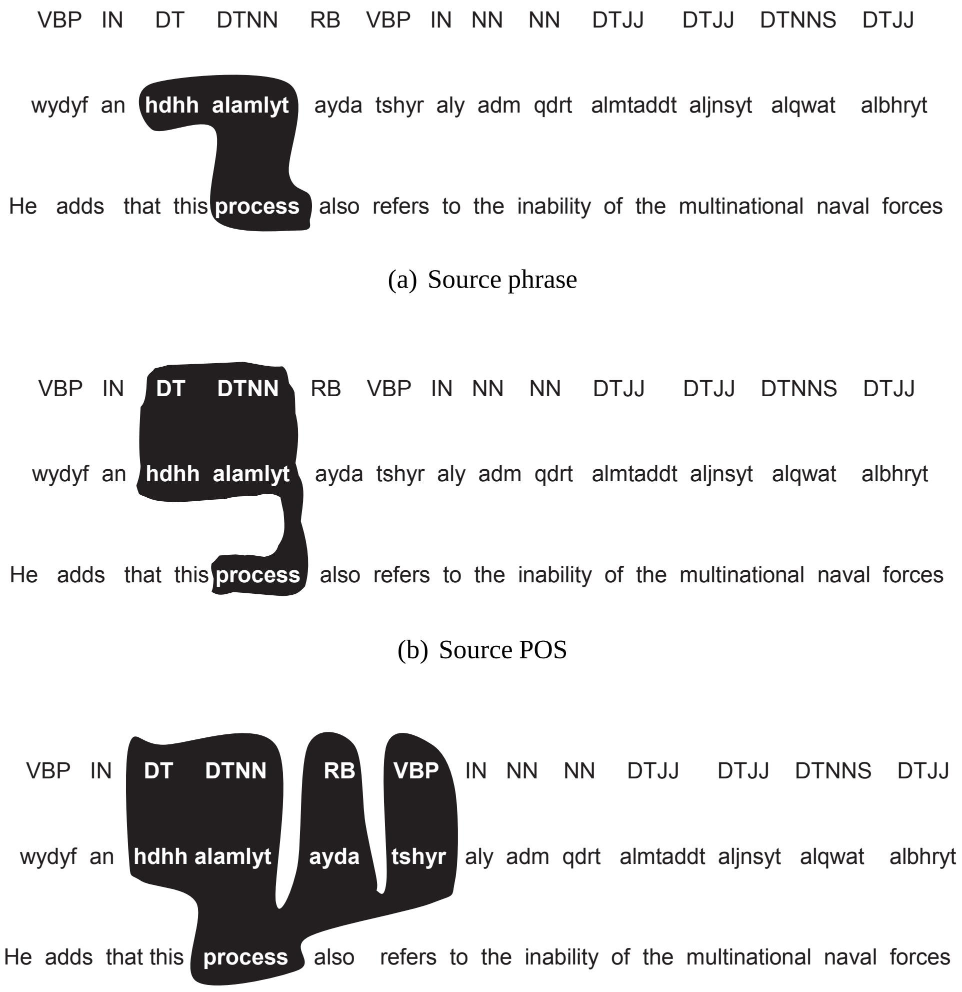
(b) Child-Father agreement

(c) Children agreement

Figure 5.1: Dependency structures features.

tree, however, we also can look down to the dependency tree with a similar motivation. Essentially, given an alignment between a source word s_i and a target word t_j, how many children of s_i and t_j are aligned together? For example, "*tshyr*" and "*refers*" have 2 aligned children which are "*ayda-also*" and "*aly-to*" as shown in Figure 5.1(c).

5.3.2 Source-side features


VBP IN DT DTNN RB VBP IN NN NN DTJJ DTJJ DTNNS DTJJ
wydyf an hdhh alamlyt ayda tshyr aly adm qdrt almtaddt aljnsyt alqwat albhryt
He adds that this process also refers to the inability of the multinational naval forces

(a) Source phrase

VBP IN DT DTNN RB VBP IN NN NN DTJJ DTJJ DTNNS DTJJ
wydyf an hdhh alamlyt ayda tshyr aly adm qdrt almtaddt aljnsyt alqwat albhryt
He adds that this process also refers to the inability of the multinational naval forces

(b) Source POS

VBP IN DT DTNN RB VBP IN NN NN DTJJ DTJJ DTNNS DTJJ
wydyf an hdhh alamlyt ayda tshyr aly adm qdrt almtaddt aljnsyt alqwat albhryt
He adds that this process also refers to the inability of the multinational naval forces

(c) Source POS and phrase in right context

Figure 5.2: Source-side features.

From MT decoder log, we can track which source phrases generate target phrases. Furthermore, one can infer the alignment between source and target words within the phrase pair using simple aligners such as IBM Model-1 alignment.

Source phrase features: These features are designed to capture the likelihood that source phrase and target word co-occur with a given error label. The intuition behind them is that if a large percentage of the source phrase and target have often been seen together with the same label, then the produced target word should have this label in the future. Figure 5.2(a) illustrates this feature template where the first line is source POS tags, the second line is the Buckwalter romanized source Arabic sequence, and the third line is MT output. The source phrase feature is defined as follow

$$f_{102}(process) = \begin{cases} 1 & \text{if source-phrase=“}\textit{hdhh alamlyt}\text{”} \\ 0 & \text{otherwise} \end{cases}$$

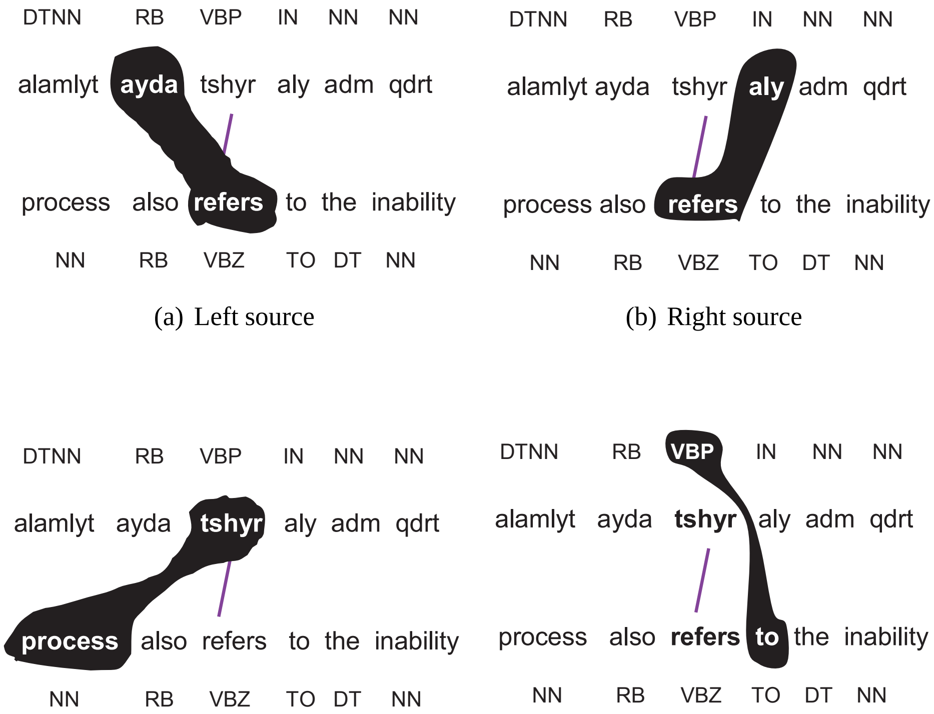
Source POS: Source phrase features might be susceptible to sparseness issues. We can generalize source phrases based on their POS tags to reduce the number of parameters. For example, the example in Figure 5.2(a) is generalized as in Figure 5.2(b) and we have the following feature:

$$f_{103}(process) = \begin{cases} 1 & \text{if source-POS=“ }\textit{DT DTNN}\text{ ”} \\ 0 & \text{otherwise} \end{cases}$$

Source POS and phrase context features: This feature set allows us to look at the surrounding context of the source phrase. For example, in Figure 5.2(c) we have “*hdhh alamlyt*” generates “*process*”. We also have other information such as on the right hand side the next two phrases are “*ayda*” and “*tshyr*” or the sequence of source target POS on the right hand side is “*RB VBP*”. An example of this type of feature is

$$f_{104}(process) = \begin{cases} 1 & \text{if source-POS-context=“ }\textit{RB VBP}\text{ ”} \\ 0 & \text{otherwise} \end{cases}$$

5.3.3 Alignment context features


DTNN RB VBP IN NN NN
alamlyt ayda tshyr aly adm qdrt
process also refers to the inability
NN RB VBZ TO DT NN

(a) Left source

(b) Right source

(c) Left target

(d) Source POS & right target

Figure 5.3: Alignment context features.

The IBM Model-1 feature performed relatively well in comparison with the WPP feature as shown by Blatz et al. (2004). In our work, we incorporate not only the IBM Model-1 feature but also the surrounding alignment context. The key intuition is that collocation is a reliable indicator for judging if a target word is generated by a particular source word (Huang, 2009). Moreover, the IBM Model-1 feature was already used in several steps of a translation system such as word alignment, phrase extraction and scoring. Also the impact of this feature alone might fade away when the MT system is scaled up.

We obtain word-to-word alignments by applying IBM Model-1 to bilingual phrase pairs that generated the MT output. The IBM Model-1 assumes one target word can only be aligned to one source word. Therefore, given a target word we can always identify which source word it is aligned to.

Source alignment context feature: We anchor the target word and derive context features surrounding its source word. For example, in Figure 5.3(a) and 5.3(b) we have an alignment between "*tshyr*" and "*refers*" The source contexts "*tshyr*" with a window of one word are "*ayda*" to the left and "*aly*" to the right.

Target alignment context feature: Similar to source alignment context features, we anchor the source word and derive context features surrounding the aligned target word. Figure 5.3(c) shows a left target context feature of word "*refers*". Our features are derived from a window of four words.

Combining alignment context with POS tags: Instead of using lexical context we have features to look at source and target POS alignment context. For instance, the feature in Figure 5.3(d) is

$$f_{141}(refers) = \begin{cases} 1 & \text{if source-POS = "\textit{VBP}"} \\ & \text{and target-context = "\textit{to}"} \\ 0 & \text{otherwise} \end{cases}$$

5.4 Experiments

5.4.1 Arabic-English translation system

The SMT engine is a phrase-based system similar to the description in (Tillmann, 2006), where various features are combined within a log-linear framework. These features include source-to-target phrase translation score, source-to-target and target-to-source word-to-word translation scores, language model score, distortion model scores and word count. The training data for these features are 7M Arabic-English sentence pairs, mostly newswire and UN corpora released by LDC. The parallel sentences have word alignment automatically generated with HMM and MaxEnt word aligner (Ge, 2004; Ittycheriah and Roukos, 2005). Bilingual phrase translations are extracted from these word-aligned parallel corpora. The language model is a 5-gram model trained on roughly 3.5 billion English words.

Our training data contains 72k sentences Arabic-English machine translation with human corrections which include of 2.2M words in newswire and weblog domains. We have a development set of 2,707 sentences, 80K words (dev); an unseen test set of 2,707 sentences, 79K words (test). Feature selection and parameter tuning has been done on the development set in which we experimented values of C, n and iterations in range of [0.5:10], [1:5], and [50:200] respectively. The final MIRA classifier was trained by using pocket crf toolkit[1] with 100 iterations, hyper-parameter C was 5 and cut-off feature threshold n was 1.

We use precision (P), recall (R) and F-score (F) to evaluate the classifier performance and they are computed as follow:

$$\begin{aligned} P &= \frac{\text{the number of correctly tagged labels}}{\text{the number of tagged labels}} \\ R &= \frac{\text{the number of correctly tagged labels}}{\text{the number of reference labels}} \\ F &= \frac{\text{2*P*R}}{\text{P+R}} \end{aligned} \tag{5.8}$$

5.4.2 Contribution of feature sets

We designed our experiments to show the impact of each feature separately as well as their cumulative impact. We trained two types of classifiers to predict the error type of each word in MT output, namely $Good$ / Bad with a binary classifier and $Good$ / $Insertion$ / $Substitution$ / $Shift$ with a 4-class classifier. Each classifier is trained with different feature sets as follow:

- WPP: we reimplemented WPP calculation based on n-best lists as described in Ueffing and Ney (2007).

[1] http://pocket-crf-1.sourceforge.net/

- WPP + target POS: only WPP and target POS features are used. This is a similar feature set used by Xiong et al. (2010).
- Our features: the classifier has source side, alignment context, and dependency structure features; WPP and target POS features are excluded.
- WPP + our features: adding our features on top of WPP.
- WPP + target POS + our features: using all features.

	binary		4-class	
	dev	test	dev	test
WPP	69.3	68.7	64.4	63.7
+ dependency structures	69.9	69.5	64.9	64.3
+ source side	72.1	**71.6**	66.2	**65.7**
+ alignment context	71.4	70.9	65.7	65.3
WPP+ target POS	69.6	69.1	64.4	63.9
+ dependency structures	70.4	70	65.1	64.4
+ source side	72.3	**71.8**	66.3	**65.8**
+ alignment context	71.9	71.2	66	65.6

Table 5.1: Contribution of different feature sets measure in F-score.

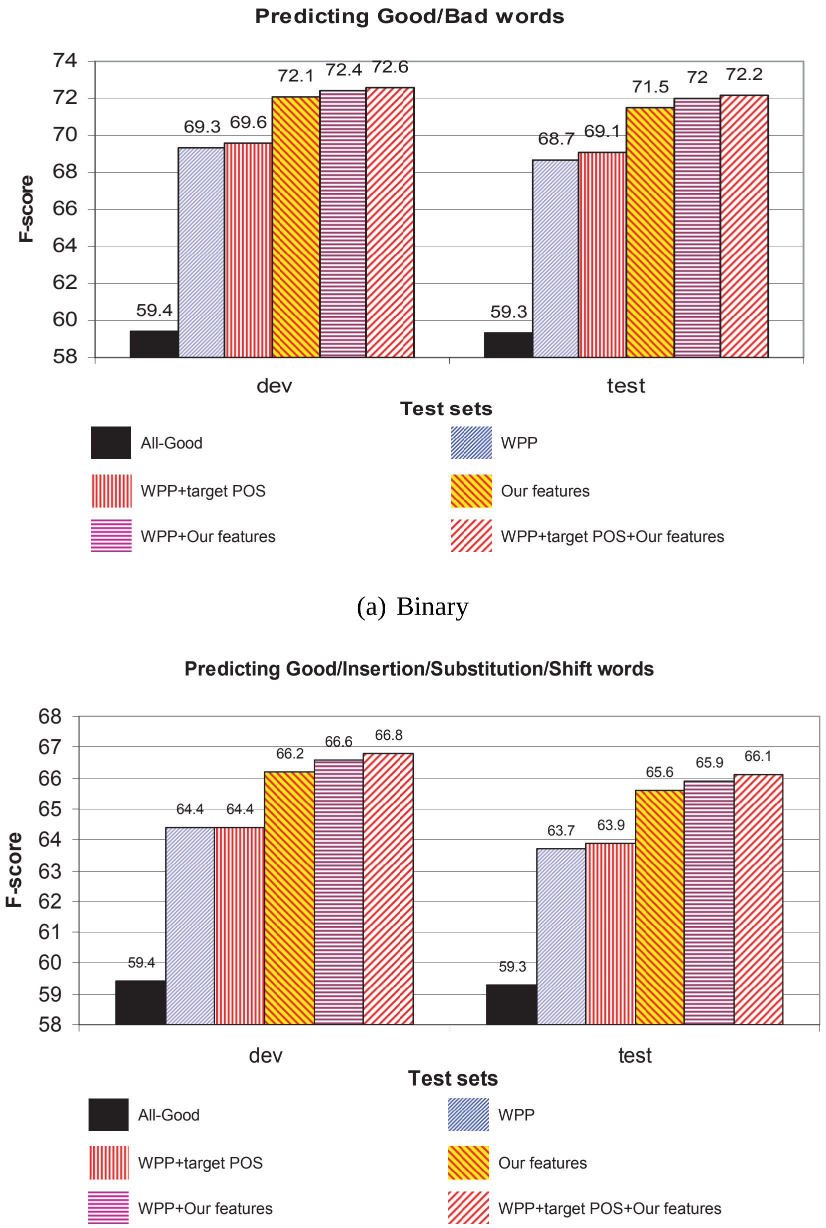
To evaluate the effectiveness of each feature set, we apply them on two different baseline systems: using WPP and WPP+target POS, respectively. We augment each baseline with our feature sets separately. Table 5.1 shows the contribution in F-score of our proposed feature sets. Improvements are consistently obtained when combining the proposed features with baseline features. Experimental results also indicate that source-side information, alignment context and dependency structures have unique and effective levers to improve the classifier performance. Among the three proposed feature sets, we observe the source side information contributes the most gain, which is followed by the alignment context and dependency structure features.

5.4.3 Performance of classifiers

We trained several classifiers with our proposed feature sets as well as baseline features. We compare their performances, including a naive baseline All-Good classifier, in which all words in the MT output are labeled as good translations. Figure 6.6 shows the performance of different classifiers trained with different feature sets on development and unseen test sets. On the unseen test set our proposed features outperform WPP and target POS features by 2.8 and 2.4 absolute F-score respectively. Improvements of our features are consistent in development and unseen sets as well as in binary and 4-class classifiers. We reach the best performance by combining our proposed features with WPP and target POS features. Experiments indicate that the gaps in F-score between our best system with the naive All-Good system is 12.9 and 6.8 in binary and 4-class cases, respectively. Table 5.2 presents precision, recall, and F-score of individual class of the best binary and 4-class classifiers. It shows that $Good$ label is better predicted than other labels, meanwhile, $Substitution$ is generally easier to predict than $Insertion$ and $Shift$.

	Label	P	R	F
Binary	Good	74.7	80.6	77.5
	Bad	68	60.1	63.8
4-class	Good	70.8	87	78.1
	Insertion	37.5	16.9	23.3
	Substitution	57.8	44.9	50.5
	Shift	35.2	14.1	20.1

Table 5.2: Detailed performance in precision, recall and F-score of binary and 4-class classifiers with WPP+target POS+Our features on the unseen test set.



(a) Binary

(b) 4-class

Figure 5.4: Performance of binary and 4-class classifiers trained with different feature sets on the development and unseen test sets.

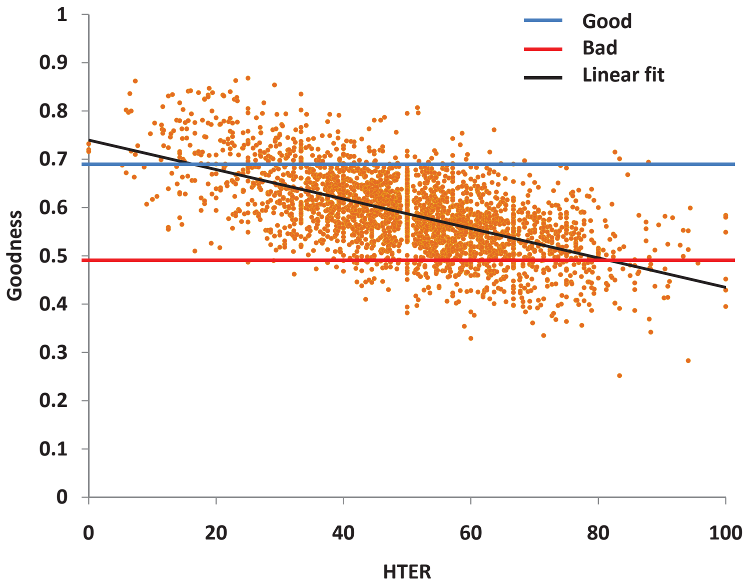


Figure 5.5: Correlation between Goodness and HTER.

5.4.4 Correlation between Goodness and HTER

We estimate sentence level confidence score based on Equation 5.7. Figure 5.5 illustrates the correlation between our proposed $Goodness$ sentence level confidence score and the human-targeted translation edit rate (HTER). The Pearson correlation between $Goodness$ and HTER is 0.6, while the correlation of WPP and HTER is 0.52. This experiment shows that $Goodness$ has a large correlation with HTER. The black bar is the linear regression line. Blue and red bars are thresholds used to visualize good and bad sentences respectively. We also experimented $Goodness$ computation in Equation 5.7 using geometric mean and harmonic mean; their Pearson correlation values are 0.5 and 0.35 respectively.

5.5 Improving MT quality with N-best list reranking

Experiments reporting in Section 5.4 indicate that the proposed confidence measure has a high correlation with HTER. However, it is not very clear if the core MT system can benefit

	Dev		Test	
	TER	BLEU	TER	BLEU
Baseline	49.9	31.0	50.2	30.6
2-best	49.5	31.4	49.9	30.8
5-best	**49.2**	**31.4**	**49.6**	**30.8**
10-best	49.2	31.2	49.5	30.8
20-best	49.1	31.0	49.3	30.7
30-best	49.0	31.0	49.3	30.6
40-best	49.0	31.0	49.4	30.5
50-best	49.1	30.9	49.4	30.5
100-best	49.0	30.9	49.3	30.5

Table 5.3: Reranking performance with $Goodness$ score.

from confidence measure by providing better translations. To investigate this question we present experimental results for the n-best list reranking task. The MT system generates top n hypotheses and for each hypothesis we compute sentence-level confidence scores. The best candidate is the hypothesis with highest confidence score. Table 5.3 shows the performance of reranking systems using $Goodness$ scores from our best classifier in various n-best sizes. We obtained 0.7 TER reduction and 0.4 BLEU point improvement on the development set with a 5-best list. On the unseen test, we obtained 0.6 TER reduction and 0.2 BLEU point improvement. Although, the improvement of BLEU score is not obvious, TER reductions are consistent in both development and unseen sets.

Figure 5.6 shows the improvement of reranking with $Goodness$ score. Besides, the figure illustrates the upper and lower bound performances with TER metric in which the lower bound is our baseline system and the upper bound is the best hypothesis in a given n-best list. Oracle scores of each n-best list are computed by choosing the translation candidate with lowest TER score.

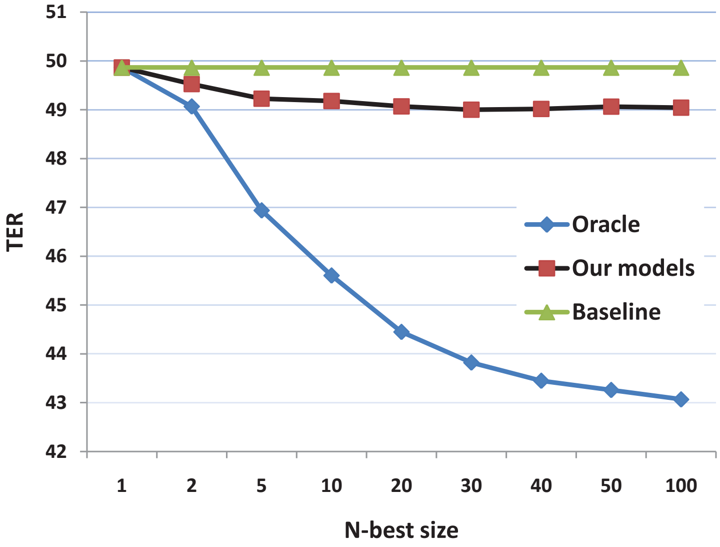


Figure 5.6: A comparison between reranking and oracle scores with different n-best size in TER metric on the development set.

5.6 Visualizing translation errors

Besides the application of confidence score in the n-best list reranking task, we propose a method to visualize translation error using confidence scores. Our purpose is to visualize word and sentence-level confidence scores with the following objectives 1) easy for spotting translations errors; 2) simple and intuitive; and 3) helpful for post-editing productivity. We define three categories of translation quality (good/bad/decent) on both word and sentence level. On word level, the marginal probability of good label is used to visualize translation errors as follow:

$$L_i = \begin{cases} good & \text{if } p(y_i = Good|S) \geq 0.8 \\ bad & \text{if } p(y_i = Good|S) \leq 0.45 \\ decent & \text{otherwise} \end{cases}$$

On sentence level, the $Goodness$ score is used as follow:

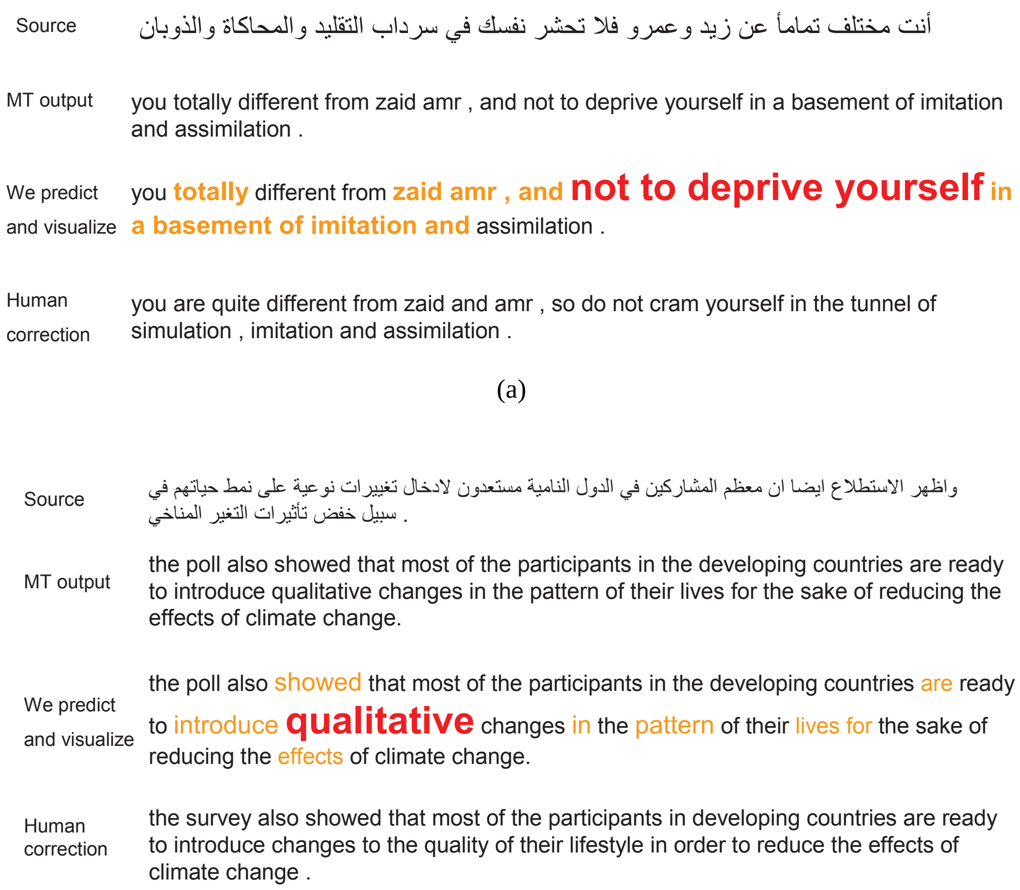
$$L_S = \begin{cases} good & \text{if } Goodness(S) \geq 0.7 \\ bad & \text{if } Goodness(S) \leq 0.5 \\ decent & \text{otherwise} \end{cases}$$

	Choices	Intention
Font size	big	bad
	small	good
	medium	decent
Colors	red	bad
	black	good
	orange	decent

Table 5.4: Choices of layout

Different font sizes and colors are used to catch the attention of post-editors whenever translation errors are likely to appear as shown in Table 5.4. Colors are applied on word level, while font size is applied on both word and sentence level. The idea of using font size and colour to visualize translation confidence is similar to the idea of using tag/word cloud to describe the content of websites[2]. The reason we are using big font size and red color is to attract post-editors' attention and help them find translation errors quickly. Figure 5.7 shows an example of visualizing confidence scores by font size and colors. It shows that "*not to deprive yourself*", displayed in big font and red color, is likely to be bad translations. Meanwhile, other words, such as "*you*", "*different*", "*from*", and "*assimilation*", displayed in small font and black color, are likely to be good translation. Medium font and orange color words are decent translations.

[2] http://en.wikipedia.org/wiki/Tag_cloud



Source أنت مختلف تماماً عن زيد وعمرو فلا تحشر نفسك في سرداب التقليد والمحاكاة والذوبان

MT output you totally different from zaid amr , and not to deprive yourself in a basement of imitation and assimilation .

We predict and visualize you **totally** different from **zaid amr , and not to deprive yourself in a basement of imitation and** assimilation .

Human correction you are quite different from zaid and amr , so do not cram yourself in the tunnel of simulation , imitation and assimilation .

(a)

Source واظهر الاستطلاع ايضا ان معظم المشاركين في الدول النامية مستعدون لادخال تغييرات نوعية على نمط حياتهم في سبيل خفض تأثيرات التغير المناخي .

MT output the poll also showed that most of the participants in the developing countries are ready to introduce qualitative changes in the pattern of their lives for the sake of reducing the effects of climate change.

We predict and visualize the poll also showed that most of the participants in the developing countries are ready to introduce **qualitative** changes in the pattern of their lives for the sake of reducing the effects of climate change.

Human correction the survey also showed that most of the participants in developing countries are ready to introduce changes to the quality of their lifestyle in order to reduce the effects of climate change .

(b)

Figure 5.7: MT errors visualization based on confidence scores.

5.7 Summary

In this chapter we proposed a method to predict confidence scores for machine translated words and sentences based on a feature-rich classifier using linguistic and context features. Our major contributions are three novel feature sets including dependency structures, source side information, and alignment context. Experimental results show that by combining the source side information, alignment context, and dependency structure features with word posterior probability and target POS context (Ueffing & Ney 2007; Xiong et al., 2010), the MT error prediction accuracy is increased from **69.1** to **72.2** in F-score. Our framework is able to predict error types namely insertion, substitution and shift. The Pearson correlation with human judgment increases from **0.52** to **0.6**. Furthermore, we show that the proposed confidence scores can help the MT system to select better translations and as a result improvements between **0.4** and **0.9** TER reduction are obtained. Finally, we demonstrate a prototype to visualize translation errors.

Chapter 6

A Statistical Sentence Simplification Model and Its Application in Machine Translation

In the NIST MT evaluations, translation systems typically have to deal with sentences with average length ranging from 27 to 36 words varying on different test sets as shown in Table 6.1. There are cases when the test sentence has up to 268 words. Similar to other NLP tasks, such as parsing and semantic role labeling, the source sentence length has a lot of impact on SMT performance. Translating long sentences is often harder than short sentences because of several reasons. First, hypotheses search space for long sentences is much larger than short sentences, as a result, good translations are harder to reach. Second, it takes more time to translate long sentences. Third, long sentences often contain complex syntax and long distance dependency structures, therefore, it is not easy for translation models to capture these phenomena. In many translation applications, such as speech-to-speech translation, the fluency might not be very important. For example, in speech-to-speech translation when the user says *”well well well my name you know is is John”* it is almost acceptable if the machine can output to the target language keywords *”my name John”*.

Test sets	Average Length	Maximum length
mt02	29	81
mt03	28.42	86
mt04	31.76	111
mt05	31.51	101
mt06	27.68	205
mt08-nw	31.92	150
mt08-wb	36.22	268

Table 6.1: Sentence length statistics on NIST MT Arabic test sets

Moreover, complicated sentences impose difficulties on reading comprehension. For instance, a person in 5th grade can comprehend a comic book easily but will struggle to understand New York Times articles which require at least 12th grade average reading level (Flesch, 1981). Complicated sentences also challenge natural language processing applications including, but not limited to, text summarization, question answering, information extraction, and machine translation (Chandrasekar et al., 1996). An example of this is syntactic parsing in which long and complicated sentences will generate a large number of hypotheses and usually fail in disambiguating the attachments.

Therefore, it is desirable to pre-process complicated sentences and generate simpler counter parts. There are direct applications of sentence simplification. Daelemans et al. (2004) applied sentence simplification so that the automatically generated closed caption can fit into limited display area. The Facilita system generates accessible content from Brazilian Portuguese web pages for low literacy readers using both summarization and simplification technologies (Watanabe et al., 2009).

This chapter tackles sentence-level factual simplification (SLFS). The objective of SLFS is twofold. First, SLFS will process the syntactically complicated sentences. Second, while preserving the content meaning, SLFS outputs a sequence of simple sentences. SLFS is an instance of the broader spectrum of text-to-text generation problems, which includes summarization, sentence compression, paraphrasing, and sentence fusion. Comparing to sentence compression, sentence simplification requires the conversion to be lossless in sense of semantics. It is also different from paraphrasing in that it generates multiple sentences instead of one sentence with different constructions.

There are certain specific characteristics that complicate a sentence, which include length, syntactic structure, syntactic and lexical ambiguity, and an abundance of complex words. As suggested by its objective, sentence simplification outputs “simple sentences”. Intuitively, a simple sentence is easy to read and understand, and arguably easily processed by computers. A more fine-tuned definition on a simple sentence is suggested in Klebanov et al. (2004), and is termed Easy Access Sentences (EAS). EAS in English is defined as

- EAS is a grammatical sentence;
- EAS has one finite verb;
- EAS does not make any claims that were not present, explicitly or implicitly;
- An EAS should contain as many named entities as possible.

While the last two requirements are difficult to quantify, the first two provide a practical guideline for sentence simplification. We treat the sentence simplification process as a process of statistical machine translation. Given the input of a syntactically complicated sentence, we translate it into a set of EAS that preserves as much information as possible from the original sentence. We develop the algorithm that can generate a set of EAS from the original sentence and a model to incorporate features that indicate the merit of the simplified candidates. The model is discriminatively trained on a data set of manually simplified sentences.

We briefly review related work in the area of text-to-text generation in Section 6.1. The proposed model for statistical sentence simplification is presented in Section 6.2. In Section 6.3 we introduce the decoding algorithm. Section 6.4 and 6.5 describe the discriminative training method we use and the feature functions. Experiments and analysis are present in Section 6.6, followed by Section 6.7 with the application of sentence simplification in a English-German MT system. Finally, we conclude this work in Section 6.8.

6.1 Related Work

Given the problematic nature of text-to-text generation that takes a sentence or a document as the input and optimizes the output toward a certain objective, we briefly review state-of-art approaches of text-to-text generation methods.

Early approaches in summarization focus on extraction methods which try to isolate and then summarize the most significant sentences or paragraphs of the text. However, this has been found to be insufficient because it usually generates incoherent summaries. Barzilay and McKeown (2005) proposed sentence fusion for multi-document summarization, which produces a sentence that conveys common information of multiple sentences based upon dependency tree structures and lexical similarity.

Sentence compression generates a summary of a single sentence with minimal information loss, which can also be treated as sentence-level summarization. This approach applies word deletion, in which non informative words will be removed from the original sentence. A variety of models were developed based on this perspective, ranging from generative models (Knight and Marcu, 2002; Turner and Charniak, 2005) to discriminative models (McDonald, 2006) and Integer Linear Programming (Clarke, 2008). Another line of research treats sentence compression as machine translation, in which tree-based translation models have been developed (Galley and McKeown, 2007; Cohn and Lapata, 2008; Zhu et al., 2010). Woodsend and Lapata (2011) proposed a framework to combine tree-based simplification with ILP.

In contrast to sentence compression, sentence simplification generates multiple sentences from one input sentence and tries to preserve the meaning of the original sentence. The major objective is to transform sentences in complicated structures to a set of easy-to-read sentences, which will be easier for human to comprehend, and hopefully easier for computers to deal with.

Numerous attempts have been made to tackle the sentence simplification problem. One line of research has explored simplification with linguistic rules. Jonnalagadda (2006) developed a rule-based system that take into account the discourse information. This method is applied on simplification of biomedical text (Jonnalagadda et al., 2009) and protein-protein information extraction (Jonnalagadda and Gonzalez, 2010). Chandrasekar and Srinivas (1997) automatically induced simplification rules based on dependency trees. Additionally, Klebanov et al. (2004) develop a set of rules that generate a set of EAS from syntactically complicated sentences. Heilman and Smith (2010) proposed an algorithm for extracting simplified declarative sentences from syntactically complex sentences.

The rule-based systems performs well on English. However, in order to develop a more **generic framework** for other languages, a statistical framework is preferable. In this work, we follow this direction to treat the whole process as a statistical machine translation task with an online large-margin learning framework. The method is generalizable to other languages given labeled data. To ensure the information is preserved, we build a table of EAS for each object, and use stack decoding to search for the optimal combination of EAS. A feature vector is assigned to each combination and we use an end-to-end discriminative training framework to tune the parameters given a set of training data. Our method is different from Klebanov et al. (2004) in the way that we applied statistical model to rank the generated sentences. The difference between our method and Heilman and Smith (2010) is that we integrate linguistic rules into the decoding process as soft constraints in order to explore a much larger search space.

6.2 Statistical Sentence Simplification Models

Assume that we are given an English sentence e, which is to be simplified into a set $\mathcal{S}$ of k simple sentences $\{s_1, ..., s_i, ..., s_k\}$. Among all possible simplified sets, we will select the set with the highest probability $\hat{\mathcal{S}}(e) = \arg\max_{\forall \mathcal{S}} Pr(\mathcal{S}|e)$. As the true probability distribution of $Pr(\mathcal{S}|e)$ is unknown, we have to approximate $Pr(\mathcal{S}|e)$ by developing a log-linear model $p(\mathcal{S}|e)$. In contrast to noisy-channel models (Knight and Marcu, 2002; Turner and Charniak, 2005) we directly compute simplification probability by a conditional exponential model as follow:

$$p(\mathcal{S}|e) = \frac{exp[\sum_{m=1}^{M} w_m f_m(\mathcal{S}, e)]}{\sum_{\mathcal{S}'} exp[\sum_{m=1}^{M} w_m f_m(\mathcal{S}', e)]} \tag{6.1}$$

where $f_m(\mathcal{S}, e), m = 1, ..., M$ are feature functions on each sentence; there exists a model parameter w_m are feature weights to be learned.

In this framework, we need to solve decoding, learning, and modeling problems. The *decoding problem*, also known as the search problem, is denoted by the $\arg\max$ operation which finds the optimal $\mathcal{S}$ that maximize model probabilities. The *learning problem* amounts to obtaining suitable parameter values w_1^M subject to a loss function on training samples. Finally, the *modeling problem* amounts to developing suitable feature functions that capture the relevant properties of the sentence simplification task. Our sentence simplification model can be viewed as English-to-English log-linear translation models. The defining characteristic that makes the problem difficult is that we need to translate from one syntactically complicated sentence to k simple sentences, and k is not predetermined.

6.3 Decoding

This section presents a solution to the *decoding problem*. The solution is based on a stack decoding algorithm that finds the best $\mathcal{S}$ given an English sentence e. Our decoding algorithm is inspired by the decoding algorithms in speech recognition and machine translation (Jelinek, 1998; Koehn et al., 2007). For example, with a sentence e "*John comes from England, works for IMF, and is an active hiker*", the stack decoding algorithm tries to find $\mathcal{S}$, which is a set of three sentences: "*John comes from England*", "*John works for IMF*" and "*John is an active hiker*". Note that $\mathcal{S}$ is a set of k simple sentences $\mathcal{S} = \{s_1, ..., s_i, ..., s_k\}$. We can assume the items s_i are drawn from a finite set $\mathbb{S}$ of grammatical sentences that can be derived from e. Therefore, the first step is to construct the set $\mathbb{S}$.

6.3.1 Constructing simple sentences

We define a simple English sentence as a sentence with SVO structure, which has one subject, one verb and one object. Our definition is similar to the definition of EAS, mentioned in section 1. However, we only focus on the SVO structure and other constraints are relaxed. We assume both subjects (S) and objects (O) are noun phrases (NP) in the parse

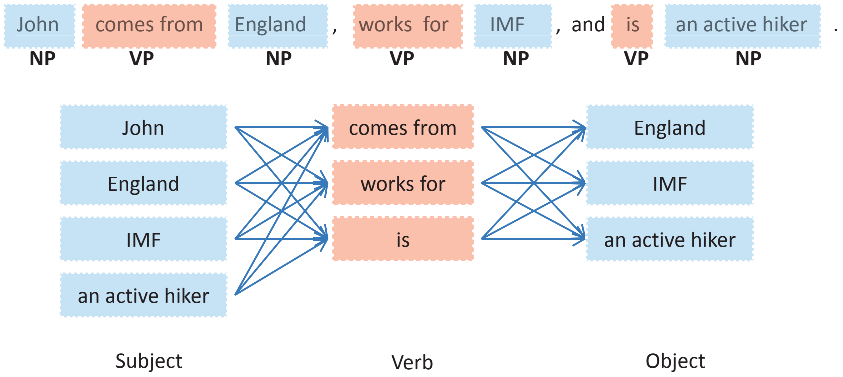


Figure 6.1: Constructing simple sentences

tree. For a given English sentence e, we extract a list S_{NP} of NPs and a list S_V of verbs. S_{NP} has an additional empty NP in order to handle intransitive verbs. A straightforward way to construct simple sentences is to enumerate all possible sentences based on S_{NP} and S_V. That results in $|S_{NP}|^2|S_V|$ simple sentences.

Figure 6.1 illustrates the constructions for "*John comes from England, works for IMF, and is an active hiker*". The system extracts a noun phrase list S_{NP} {*John, England, IMF, an active hiker*} and a verb list S_V {*comes from, works for, is*}. Our model constructs simple sentences such as "*John comes from England*" , "*John comes from IMF*" and "*John comes from an active hiker*". The total number of simple sentences, $|\mathbb{S}|$, is 48.

6.3.2 Decoding algorithm

Given a list of simple sentences $\mathbb{S}$, a number of possible combinations could be applied. The decoder's objective is to construct and find the best simplification candidate $\mathcal{S} \subset \mathbb{S}$ which conveys the closest meaning with the original sentence. We call $\mathcal{S}$ a *hypothesis* in the context of the decoder. Simple sentences are constructed beforehand and associated with a feature vector. We employs a stack decoding algorithm. The rationale is to construct a hypothesis that **covers all noun phrases and verb phrases of the original sentence**.

The decoding task is to find the optimal solution over all possible combinations of simple sentences, given the feature values and learned feature weights. Depending on **the number of simple sentences per hypothesis**, the search space grows exponentially. Since each simple sentence contains an object, we can group the candidate sentences by its object. An object is a noun phrase of the original sentence which is extracted by using a noun phrase chunker. Each object has an order depending on the position of the last word. For instance, *"IMF"* is object number two and *"an active hiker"* is an object number three in Figure 6.1. Any noun phrase can serve as an object except the NP at the beginning of a sentence. Therefore *"John"* will not be a potential object. The decoder will use object order as a feature in order to control the order of simplified sentences in a hypothesis. For example, given 2 hypothesis *"John come from England; John works for IMF; John is an active hiker"* and *"John is an active hiker; John come from England; John works for IMF"*. The decoder will prefer the first hypothesis since its objects are in the same sequence with the original sentence.

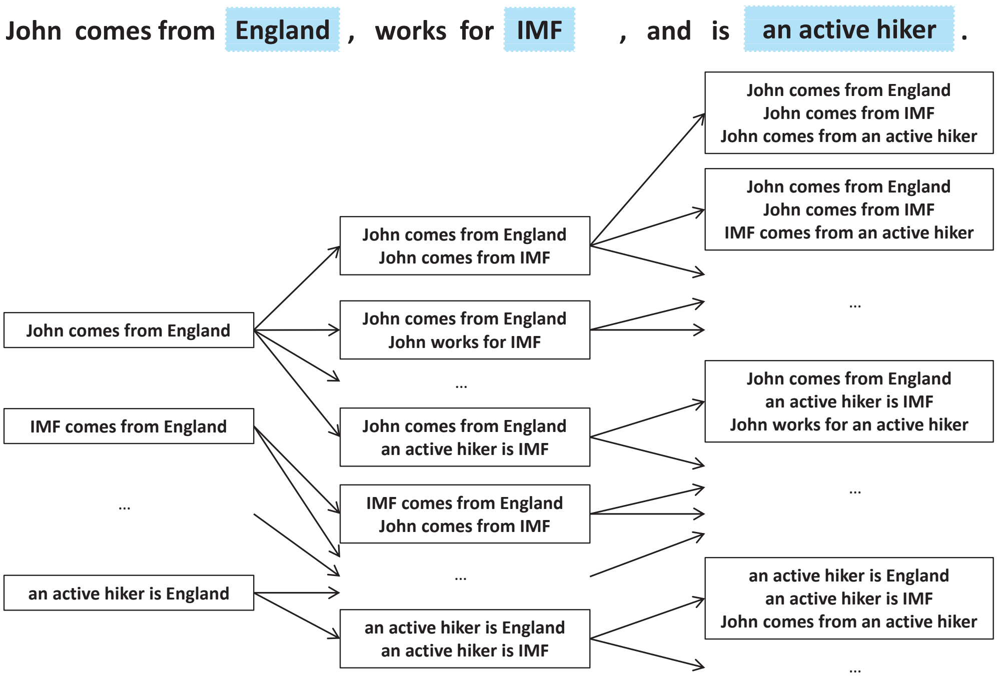


Figure 6.2: Decoding by objects

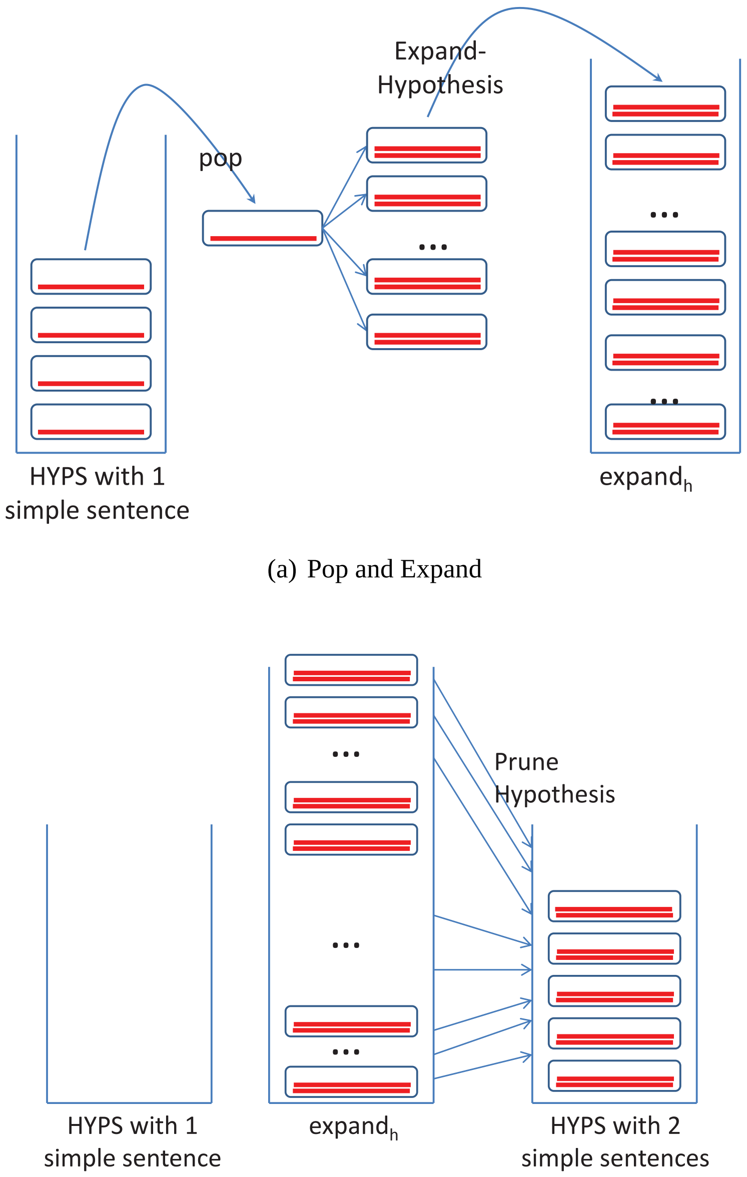
Figure 6.2 demonstrates the idea of decoding via objects. We have three potential objects “*England*”, “*IMF*” and “*an active hiker*”. The algorithm first finds potential simple sentences which have “*England*” as object. After finishing “England”, the algorithm expands to “*IMF*” and “*an active hiker*”. Based on model scores, the decoder will choose a k-best hypothesis.

Algorithm 6 : K-Best Stack Decoding

```
1:  Initialize an empty hypothesis list HypList
2:  Initialize HYPS is a stack of 1-simple-sentence hypotheses
3:  for $i = 0$ to $|S_V|$ do
4:      Initialize stack $expand_h$
5:      while HYPS is not empty do
6:          pop $h$ from HYPS
7:          $expand_h \leftarrow$ Expand-Hypothesis($h$)
8:      end while
9:      $expand_h \leftarrow$ Prune-Hypothesis($expand_h$, stack-size)
10:     HYPS $\leftarrow expand_h$
11:     Store hypotheses of $expand_d$ into HypList
12: end for
13: SortedHypList $\leftarrow$ Sort-Hypothesis(HypList)
14: Return K-best hypotheses in SortedHypList
```

Algorithm 6 is a version of stack decoding for sentence simplification. The decoding process advances by extending a state that is equivalent to a stack of hypotheses. Line 1 and 2 initialize *HYPS* stack and *HypList*. A *HYPS* stack maintains a current search state, meanwhile *HypList* stores potential hypotheses after each state. *HYPS* is initialized with hypotheses containing one simple sentence. Line 3 starts a loop over states. The number of maximum states is equal to the size of S_V plus one. Lines 4-8 represent the hypothesis expansion.

Figure 6.3(a) illustrates the pop-expand process of *HYPS* stack with 1-simple-sentence hypotheses. The expansion in this situation expands to a 2-simple-sentence hypotheses-stack $expand_h$. The size of $expand_h$ will exponentially increase according to the size of



(a) Pop and Expand

(b) Hypothesis pruning

Figure 6.3: A visualization for stack decoding

S_V and S_{NP}. Therefore, we prefer to maintain $expand_h$ within a limit number (***stack-size***) of hypotheses. Line 9 helps the decoder to control the size of $expand_h$ by applying different pruning strategies: word coverage, model score or both. Figure 6.3(b) illustrates the pruning process on $expand_h$ with 2-simple-sentence hypotheses. Line 10 replaces the current state with a new state of the expanded hypotheses. Before moving to a new state, *HypList* is used to preserve potential hypotheses of the current state. Line 13 sorts hypotheses in *HypList* according to their model scores and a K-best list is returned in line 14.

6.4 Learning

Since defining a log-linear sentence simplification model and decoding algorithm has been completed, this section describes a discriminative learning algorithm for the *learning problem*. We learn optimized weight vector w by using the Margin Infused Relaxed Algorithm or MIRA (Crammer and Singer, 2003), which is an online learner closely related to both the support vector machine and perceptron learning framework. In general, weights are updated at each step time i according to:

$$\begin{aligned} w_{i+1} = \arg\min\nolimits_{w_{i+1}} ||w_{i+1} - w_i|| \\ \text{s.t.} \quad score(\mathcal{S}, e) \geq score(\mathcal{S}', e) + L(\mathcal{S}, \mathcal{S}') \end{aligned} \tag{6.2}$$

where $L(\mathcal{S}, \mathcal{S}')$ is a measure of the loss of using $\mathcal{S}'$ instead of the simplification reference $\mathcal{S}$; $score()$ is a cost function of e and $\mathcal{S}$ and in this case is the decoder score.

Algorithm 7 is a version of MIRA for training the weights of our sentence simplification model. On each iteration, MIRA considers a single instance from the training set $(\mathcal{S}_t, e_t)$ and updates the weights so that the score of the correct simplification ε_t is greater than the score of all other simplifications by a margin proportional to their loss. However, given a sentence there are an exponential amount of possible simplification candidates. Therefore, the optimizer has to deal with an exponentially large number of constraints. To tackle this, we only consider K-best hypotheses and choose m-oracle hypotheses to

Algorithm 7 : MIRA training for Sentence Simplifier

```
  training set $\tau = \{f_t, e_t\}_{t=1}^{T}$ has T original English sentences with the feature vector $f_t$ of $e_t$.
  $\varepsilon$ is the simplification reference set.
  m-oracle set $O = \{\}$.
  The current weight vector $w^i$.
i=0
for $j = 1$ to Q do
    for $t = 1$ to T do
        $H \leftarrow$ get_K_Best($\mathcal{S}_t$ ; $w^i$)
        $O \leftarrow$ get_m_Oracle($H$ ; $\varepsilon_t$)
        $\gamma = \sum_{o=1}^{m} \sum_{h=1}^{K} \alpha(e_o, e_h; \varepsilon_t)(f_{e_o} - f_{e_h})$
        $w^{i+1} = w^i + \gamma$
        $i = i + 1$
    end for
end for
Return $\frac{\sum_{i=1}^{Q*T} w^i}{Q*T}$
```

support the weight update decision. This idea is similar to the way MIRA has been used in dependency parsing and machine translation (McDonald et al., 2005; Liang et al., 2006a; Watanabe et al., 2007).

On each update, MIRA attempts to keep the new weight vector as close as possible to the old weight vector. Subject to margin constraints keep the score of the correct output above the score of the guessed output by updating an amount given by the loss of the incorrect output. In line 6, α can be interpreted as an update step size; when α is a large number we want to update our weights aggressively, otherwise weights are updated conservatively. α is computed as follow:

$$\begin{aligned} &\alpha = \max(0, \delta) \\ &\delta = \min\left\{ C, \frac{L(e_o, e_h; \varepsilon_t) - [score(e_o) - score(e_h)]}{||\mathcal{S}_{e_o} - f_{e_h}||_2^2} \right\} \end{aligned} \tag{6.3}$$

where C is a positive constant used to cap the maximum possible value of α; $score()$ is the decoder score; and $L(e_o, e_h; \varepsilon_t)$ is the loss function.

$L(e_o, e_h; \varepsilon_t)$ measures the difference between oracle e_o and hypothesis e_h according to the gold reference ε_t. L is crucial to guide the optimizer to learn optimized weights. We defined $L(e_o, e_h; \varepsilon_t)$ as follow

$$L(e_o, e_h; \varepsilon_t) = AveF_N(e_o, \varepsilon_t) - AveF_N(e_h, \varepsilon_t) \tag{6.4}$$

where $AveF_N(e_o, \varepsilon_t)$ and $AveF_N(e_h, \varepsilon_t)$ is the average n-gram (n=[2:N]) cooccurrence F-score of (e_o, ε_t) and (e_h, ε_t), respectively.

In this case, we optimize the weights directly against the $AveF_N$ metric over the training data. $AveF_N$ can be substituted by other evaluation metrics such as the ROUGE family metric (Lin, 2004a). Similar to the perceptron method, the actual weight vector during decoding is averaged across the number of iterations and training instances; and it is computed in line 11.

6.5 Modeling

We now turn to the *modeling problem*. Our fundamental question is: given the model in Equation 6.1 with M feature functions, what linguistic features can be leveraged to capture semantic information of the original sentence? We address the question in this section by describing features that cover different levels of linguistic structures. Our model incorporates 177 features based on information from the original English sentence e which contains chunks, syntactic and dependency parse trees (Ramshaw and Marcus, 1995; Marneffe et al., 2006).

6.5.1 Simple sentence level features

A simplification hypothesis s contains k simple sentences. Therefore, it is crucial that our model chooses reasonable simple sentences to form a hypothesis. For each simple sentence s_i we incorporated the following feature functions:

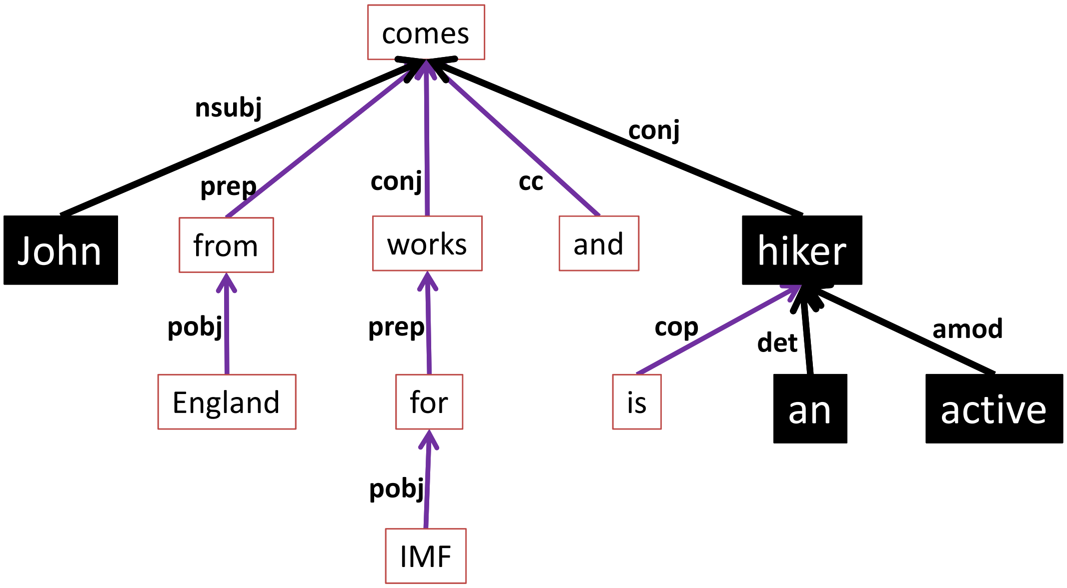


Figure 6.4: Dependency structure distance

Dependency Structures It is possible that the decoder constructs semantically incorrect simple sentences, in which S, V, and O do not have any semantic connection. One way to possibly reduce this kind of mistake is analyze the dependency chain between S, V, and O on the original dependency tree of e. Our dependency structure features include the minimum and maximum distances of (S:O), (S:V), and (V:O). In Fig 6.4, the minimum and maximum distances between "*John*" and "*an active hiker*" are 2 and 3, respectively.

Word Count These features count the number word in subject (S), verb (V) and object (O), also counting the number of proper nouns in S and the number of proper nouns in O.

Distance between NPs and Verbs These features focus on the number of NPs and VPs in between S, V, and O. This feature group includes the number of NPs between S and V, the number of NPs between V and O, the number of VPs between S and V, the number of VPs between V and O.

Syntactic Structures Another source of information is the syntactic parse tree of e, which can be used to extract syntactic features. The sentence-like boundary feature considers the path from S to O along the syntactic parse tree to see whether it crosses the sentence-like boundary (e.g. relative clauses). For example in the original sentence "*John comes from England and works for IMF which stands for International Monetary Funds*", the simple sentence "*IMF stands for International Monetary Funds*" has sentence-like boundary feature is triggered since the path from "*IMF*" to "*International Monetary Funds*" on the syntactic tree of the original sentence contains an SBAR node.

Another feature is the PP attachment feature. This checks if the O contains a prepositional phrase attachment or not. Moreover, the single pronoun feature will check if S and O are single pronoun or not. The last feature is the VO common ancestor, which looks at the syntactic tree to see whether or not V and O share the same VP tag as a common ancestor.

6.5.2 Interactive simple sentence features

A collection of grammatically sound simplified sentences does not necessarily make a good hypothesis. Dropping words, unnecessary repetition, or even wrong order can make the hypothesis unreadable. Therefore, our model needs to be equipped with features that are capable to measure the interactiveness across simple sentences and are also able to represent s in the best possible manner. We incorporated the following features into our model:

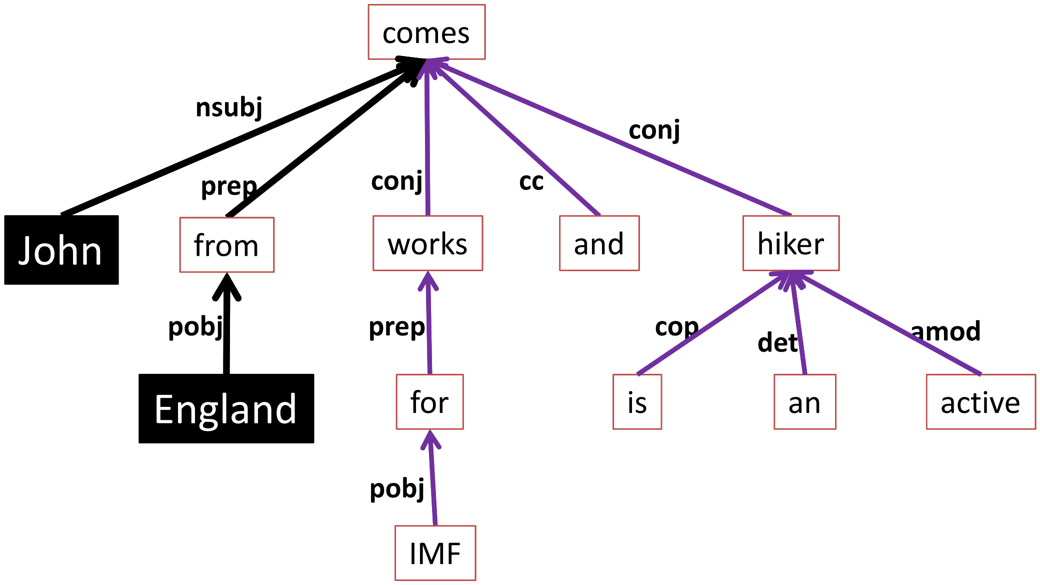


Figure 6.5: Typed dependency structure binary feature

Typed Dependency At simple sentence level we examine dependency chains of S, V and O, while at the hypothesis level we analyze the typed dependency between words. In Fig 6.5, considering"*John*" and "*England*" the typed features, such as has_Object, has_Subject, and has_Prep, will be fired with true values since the dependency link between "*John*" and "*England*" contains these types. Meanwhile, other typed dependency structure feature, such as has_Cop and has_Det, will has false values. Our model has 46 typed dependencies which are represented by the 92 count features for the 1st and 2nd simple sentence.

Sentence Count This group of features consider the number of sentences in the hypothesis. It consists of an integral feature of sentence count $sci = |S|$, and a group of binary features $scb_k = \delta(|S|) = k$ where $k \in [1, 6]$ is the number of sentence.

NP and Verb Coverage The decoder's objective is to improve the chance of generating hypotheses that cover all NP and verbs of the original sentence e. These features count the number of NPs and verbs that have been covered by the hypothesis, by the 1st and 2nd simple sentences. Similarly, these features also count the number of missing NPs and verbs.

S and O cross sentences These features count how many times S of the 1st simple sentence is repeated as S of the 2nd simple sentence in a hypothesis. They also count the number of times O of the 1st sentence is the S of 2nd sentence.

Readability This group of features computes statistics related to readability. It includes Flesch, Gunning-Fog, SMOG, Flesch-Kincaid, automatic readability index, and average all scores (Flesch, 1948; Gunning, 1968; McLaughlin, 1969; Kincaid et al., 1975). Also, we compute the edit-distance of hypothesis against the original sentence, and the average word per simple sentence.

6.6 Experiments and Analysis

6.6.1 Data

To enable the study of sentence simplification with our statistical models, we search for *parallel* corpora, in which the sources are original English sentences and the target is its simplification reference. For example, the source is "*Lu is married to Lian Hsiang , who is also a vajra master , and is referred as Grand Madam Lu* ". The simplification reference contains 3 simple sentences which are "*Lu is married to Lian Hsiang*"; "*Lian Hsiang is also a vajra master*"; "*Lu is referred as Grand Madam Lu*". To the best of our knowledge, there is no such publicly available corpora under these conditions[1].

[1] We are aware of data sets from (Cohn and Lapata, 2008; Zhu et al., 2010), however, they are more suitable in sentence compression task than in our task.

Our first attempt is to collect data automatically from original English and Simple English Wikipedia, based on the suggestions of Napoles and Dredze (2010). However, we found that the collected corpus is unsuitable for our model. For example, consider the original sentence "*Hawking was the Lucasian Professor of Mathematics at the University of Cambridge for thirty years, taking up the post in 1979 and retiring on 1 October 2009*". The Simple Wikipedia reads "*Hawking was a professor of mathematics at the University of Cambridge (a position that Isaac Newton once had)*" and "*He retired on October 1st 2009*". The problems with this are that "*(a position that Isaac Newton once had)*" did not appear in the original text, and the pronoun "*He*" requires our model to perform anaphora resolution which is out of scope of this work.

We finally decided to collect a set of sentences for which we obtained one manual simplification per sentence. The corpus contains 854 sentences, among which 25% sentences are from the New York Times and 75% sentences are from Wikipedia. The average sentence length is 30.5 words. We reserved 100 sentences for the unseen test set and the rest is for the development set and training data. The annotators were given instructions that explained the task and defined sentence simplification with the aid of examples. They were encouraged not to introduce new words and try to simplify by restructuring the original sentence. They were asked to simplify while preserving all important information and ensuring the simplification sentences remained grammatically correct[2]. Some examples from our corpus are given below:

Original: "*His name literally means Peach Taro ; as Taro is a common Japanese boy 's name , it is often translated as Peach Boy .*"
Simplification: "*His name literally means Peach Taro*" ; "*Taro is a common Japanese boy 's name*" ; "*Taro is often translated as Peach Boy*"

Original: "*These rankings are likely to change thanks to one player , Nokia , which has seen its market share shrink in the United States .*"
Simplification: "*These rankings are likely to change thanks to one player , Nokia*" ; "*Nokia*

[2] Our corpus will be made publicly available for other researchers.

has seen its market share shrink in the United States"

6.6.2 Evaluation methods

Evaluating sentence simplification is a difficult problem. One possible way to overcome this is to use readability tests. There have been readability tests such as Flesch, Gunning-Fog, SMOG, Flesch-Kincaid, etc. (Flesch, 1948; Gunning, 1968; McLaughlin, 1969; Kincaid et al., 1975). In this work, we will use Flesch-Kincaid grade level which can be interpret as the number of years of education generally required to understand a text.

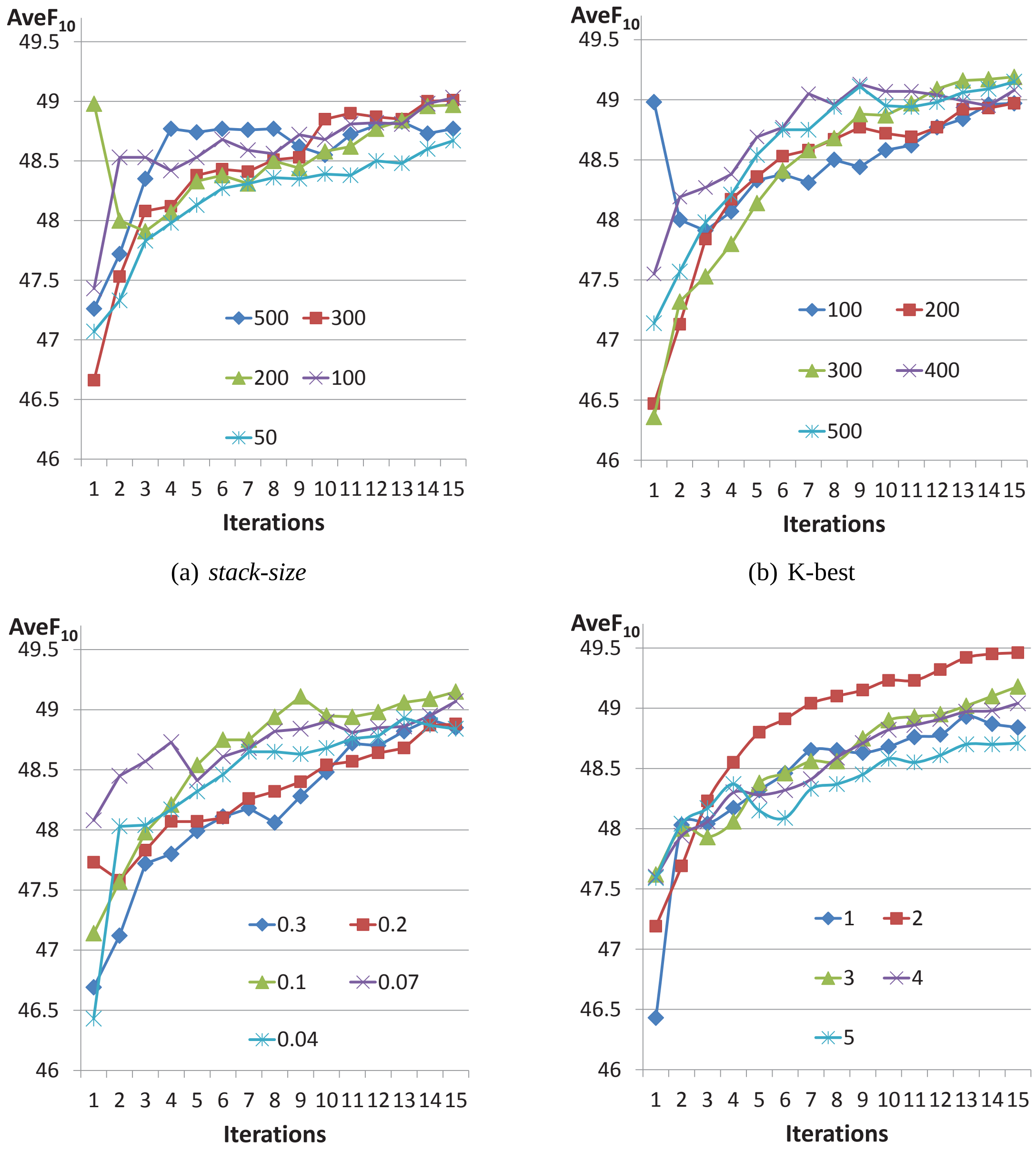
Furthermore, automatic evaluation of summaries has also been explored recently. The work of Lin (2004a) on the ROUGE family metric is perhaps the best known study of automatic summarization evaluation. Other methods have been proposed such as Pyramid (Nenkova et al., 2007). Recently, Aluisio et al. (2010) proposed readability assessment for sentence simplification.

Our models are optimized toward $AveF_{10}$, which is the average F-score of n-gram concurrence between hypothesis and reference in which n is from 2 to 10. Besides $AveF_{10}$, we will report automatic evaluation scores on the unseen test set in Flesch-Kincaid grade level, ROUGE-2 and ROUGE-4. When we evaluate on a test set, a score will be reported as the average score per sentence.

6.6.3 Model behaviors

How well does our system learn from the labeled corpus? To answer this question we investigate the interactions of model and decoder hyper parameters over the training data. We performed controlled experiments on *stack-size*, K-best, C, and m-oracle parameters. For each parameter, all other model and decoder values are fixed, and the only change is with the parameter's value of interest. Figure 6.6 illustrates these experiments with parameters over the training data during 15 MIRA training iterations with $AveF_{10}$ metric. The weight vector w is initialized randomly.

In Figure 6.6(a), we experimented with 5 different values from 100 to 500 hypotheses



(a) *stack-size*

(b) K-best

(c) Constant C

(d) m-oracle

Figure 6.6: Performance of the sentence simplifier on training data over 15 iterations when optimized toward $AveF_{10}$ metric and under various conditions.

per stack. The expected outcome is when we use a larger *stack-size* the decoder may has more chance to find better hypotheses. However, a larger *stack-size* will obviously cost more memory and run time is slower. Therefore, we want to find a *stack-size* that compromises conditions. These experiments show that with a *stack-size* of 200, our model performed reasonably well in comparison with 300 and 500. A *stack-size* of 100 is no better than 200, while a *stack-size* of 50 is much worse than 200.

In Figure 6.6(b), we experimented with 5 different values of K-best list with K from 100 to 500. We observed a K-best list of 300 hypotheses seems to perform well compare to other values. In terms of stability, the curve of 300-best list appears less fluctuation than other curves over 15 iterations.

C is the hyper-parameter which is used in Equation 6.3 for weight updating in MIRA. Figure 6.6(c) shows experiments with different constant C. If C is a large number, it means our model prefers an aggressive weight updating scheme, otherwise, our model updates weights conservatively. When C is 0.3 or 0.2 the performance is worse than 0.1 or 0.07 and 0.04.

The last controlled experiments are shown in Figure 6.6(d), in which we test different values of m ranging from 1 to 5. These experiments show that using 2 oracle hypotheses consistently leads to better performances in comparison with other values.

6.6.4 Performance on the unseen test set

After exploring different model configurations we trained the final model with *stack-size* = 200; K-best = 300; C = 0.04; and m-oracle = 2. $AveF_{10}$ score of the final system on the training set is 50.69 which is about one $AveF_{10}$ point better than any system in Figure 6.6. We use the final system to evaluate on the unseen test set. Also, we compare our system with the rule-based system (henceforth H&S) proposed by Heilman and Smith (2010).[3]

[3] We thank Michael Heilman for providing us his code. We could not reach the authors of (Zhu et al., 2010) in order to obtain outputs. Kristian Woodsend kindly provided us **partial outputs** of (Woodsend and Lapata, 2011), therefore we did not include their outputs in this section.

Original	Reference	H&S	Our system
9.6	8.2	8.3	7.9

Table 6.2: Flesch-Kincaid grade level of original, reference, H&S, and our proposed simplification on the unseen test set.

We first compare our system with H&S in the Flesch-Kincaid grade level, which indicates comprehension difficulty when reading an English text. The higher the number the more difficult the text. Table 6.2 shows the original text requires a reader of grade level 9 or 10. Both H&S and us provided simplification candidates, which are easier to read compared to the original text. Our model generated simpler hypotheses than the reference, while H&S outputs were slightly more difficult to read than the reference.

System	$AveF_{10}$	ROUGE-2	ROUGE-4
H&S	51.0	82.2	72.3
Our system	55.5	82.4	72.9

Table 6.3: Results on the unseen test set with $AveF_{10}$, ROUGE-2 and ROUGE-4 scores. Our system outperforms the rule-based system proposed by Heilman and Smith (2010).

Next, we compare our system with H&S in ngram-based metrics such as $AveF_{10}$, ROUGE-2 and ROUGE-4 as shown in Table 6.3. Our results are better than H&S by 0.2 and 0.6 point in ROUGE-2 and ROUGE-4, respectively. More interestingly, our system outperformed H&S by 4.5 points in $AveF_{10}$, which is a metric considering both precision and recall up to 10-gram. Over 100 sentences of the unseen test set, H&S outperforms us in 43 sentences, but is worse than our system in 51 sentences.

Table 6.4 shows examples of our system on the unseen test set. We present examples in cases where the proposed model works well and does not work well. In the case of negative examples, a reason that our model can not generate the correct hypothesis is the number of simple sentences per hypothesis. The majority of training examples has 2 or 3 simple sentences per hypothesis, and very few examples contain more than 3. As a result, when the model is discriminatively trained it has a strong tendency to generate hypothesis with 2 or 3 simple sentences.

	Positive examples
O	In 2011 , IBM gained worldwide attention for its artificial intelligence program Watson , which was exhibited on Jeopardy against game show champions Ken Jennings and Brad Rutter .
S	Watson was exhibited on Jeopardy against game show champions Ken Jennings and Brad Rutter . In 2011 , IBM gained worldwide attention for its artificial intelligence program Watson .
R	In 2011 , IBM gained worldwide attention for its artificial intelligence program Watson . Watson was exhibited on Jeopardy against game show champions Ken Jennings and Brad Rutter .
O	He told Radiozurnal that he was halting the campaign for Christmas and would restart it in the new year .
S	He told Radiozurnal . He was halting the campaign for Christmas . He would restart it in the new year .
R	He told Radiozurnal . He was halting the campaign for Christmas . He would restart it in the new year .
	Negative examples
O	He drives a 10-year-old Opel Corsa , but lives in a pleasant town house in the sleepy capital, Maseru, with wireless Internet and a housekeeper who comes twice a week .

S	He drives a 10-year-old Opel Corsa . He lives in a pleasant town house in the sleepy capital, Maseru, with wireless Internet and a housekeeper who .
R	He drives a 10-year-old Opel Corsa . He lives in a pleasant town house in the sleepy capital, Maseru, with wireless Internet and a housekeeper . a housekeeper comes twice a week .
O	An elderly Georgian woman was scavenging for copper to sell as scrap when she accidentally sliced through an underground cable and cut off Internet services to all of neighbouring Armenia , it emerged on Wednesday .
S	An elderly Georgian woman was scavenging for copper to sell . scrap cut off Internet services to all of neighbouring Armenia .
R	An elderly Georgian woman was scavenging for copper to sell as scrap . she accidentally sliced through an underground cable . she cut off Internet services to all of neighbouring Armenia . it emerged on Wednesday .

Table 6.4: We show the original sentence (O), our simplification (S), and simplification reference (R). Positive examples are cases when our simplifications closely match with the reference. Meanwhile, negative examples show cases when our model can not produce good simplifications.

6.7 Application to Machine Translation

Experiments reporting in previous sections demonstrate the effectiveness of the proposed sentence simplification model on different evaluation metrics. However, it is not very clear if a machine translation system can benefit from sentence simplification. In this section, we try to answer the following questions:

- Does manual simplification help the MT system to generate better translations?
- To what extend automatic simplification will be helpful for machine translation?

6.7.1 Experiment setup

To investigate the above questions we present experimental results on a English-German translation system. The SMT engine is a Moses phrase-based system (Koehn et al., 2007) which was built follow the guidelines of the 2011 machine translation workshop[4]. The translation model was trained on 1.5M sentence pairs. The 5gram language model was trained on 49M words of Europarl and News Commentary corpora.

We present a human evaluation designed to determine whether native speakers prefer manual simplification translation output. First, we draw 70 sentences with the average sentence length 31 from the news-2008, news-2009, and news-2009 test sets. We manually simplify 70 sentences. Some examples from our manual simplification (MS) are given in Table 6.6.

We use the baseline English-German to generate two sets of translation output. The first set comes with the input as original source English sentences . The other set is generated by translating manual simplification sentences. For each sentence, we provide two human annotators a set of translation reference (Ref), original translation (Orig), and manual simplification translation (Man). The annotators are asked to indicate which of the two system translations Orig or Man they prefer. Some comparison criteria we suggest for the annotators to consider are

- Information: compare to the reference the Man is better than the Orig because it contains more information.
- Grammaticality: the Orig is better than the Man because it is more grammatically correct.

[4] http://www.statmt.org/wmt11/baseline.html

- Readability: the Man is better because it is easier to read and comprehend.

6.7.2 Experimental results

		Annotator #2		
		Man	Orig	sum(#1)
Annotator #1	Man	**32**	9	41
	Orig	12	**17**	29
	sum(#2)	44	26	

Table 6.5: Confusion matrix from human evaluation for manual simplification translation

The aggregate results of our human evaluation are shown in the bottom row and right-most column of Table 6.5. The inter-annotator reliability is 0.37 which indicates a fair agreement between annotators. The annotators prefer manual simplification translation in over **63%** of the test sentences, while prefer the original in less than **37%** of the test sentences. There are a few more off-diagonal points than one might expect, but it is clear that the two annotators are in agreement with respect to manual simplification translation improvements.

	Positive examples
Src	but it is about a long term advantage , with a certain degree of indetermination , because the team can be eliminated first of change , and in addition with this action the players fulfill a sanction game and go to the second cycle of cards , in which the suspension by card accumulation takes place with one less than in the first cycle .
MS	but it is about a long term advantage , with a certain degree of indetermination . because the team can be eliminated first of change . and in addition with this action the players fulfill a sanction game . the players go to the second cycle of cards .

	in which the suspension by card accumulation takes place with one less than in the first cycle .
Orig	aber , weil die mannschaft kann zuerst behoben werden , und aušerdem mit dieser aktion , die akteure erfüllen eine sanktion spiel und werden in den zweiten zyklus der karten , in dem die aussetzung von karte anhäufung mit einer weniger als in der ersten runde .
Man	aber es geht um eine langfristige vorteile , mit einem gewissen grad an indetermination . denn die mannschaft abgebaut werden kann erstens der klimawandel . und aušerdem mit dieser aktion , die akteure erfüllen eine sanktion spiel . die akteure auf den zweiten zyklus der karten . in dem die aussetzung von karte anhäufung mit einer weniger als in der ersten runde .
Ref	aber es handelt sich um einen langfristigen , bis zu einem bestimmten grad ungewissen vorteil , da das team irgendwann auch ausscheiden könnte , aušerdem erhalten die spieler mit dieser vorgehensweise eine sanktion und gehen in den zweiten kartenzyklus über , in dem sie wegen kartenanhäufung mit einer karte weniger als im ersten zyklus ausscheiden .
Src	though berdych is charged in this case as well , he is still waiting for the verdict in the first instance .
MS	though berdych is charged in this case as well . he is still waiting for the verdict in the first instance .
Orig	wenn aufgeladen ist auch in diesem fall ist er immer noch darauf warten , dass das urteil in der ersten instanz .
Man	wenn aufgeladen ist auch in diesem fall . er ist noch immer darauf warten , dass das urteil in der ersten instanz .
Ref	berdych selbst ist in dieser sache zwar auch angeklagt , doch steht ihm zunächst das erstinstanzliche urteil bevor .

Negative examples

Src	when the oil prices went down , around 1980 , the idea of exploiting marine energy was put aside , but now the appeals by the environmentalists and the new increases in oil prices have given impetus to the sector .
MS	when the oil prices went down , around 1980 . the idea of exploiting marine energy was put aside . but now the appeals by the environmentalists and the new increases in oil prices have given impetus to the sector .
Orig	wenn die ölpreise ging , um 1980 , die vorstellung von der nutzung der energie beiseite lassen , aber jetzt die appelle der umweltschützer und der neuen anstieg der ölpreise haben impulse für den sektor .
Man	wenn die ölpreise ging , um 1980 . die idee der nutzung der energie beiseite . aber jetzt die appelle der umweltschützer und der neuen anstieg der ölpreise vorschub geleistet haben , in der sich der sektor .
Ref	als der preis des öls um 1980 einbrach vergas man die idee der meeresenergie , jetzt aber drängen die umweltorganisationen und der gestiegene ölpreis auf neue impulse in diesem sektor .
Src	when a patient is admitted to the hospital , one of the first things done is to check for the presence of mrsa , but hospitals are still vulnerable to infection .
MS	when a patient is admitted to the hospital . one of the first things done is to check for the presence of mrsa . but hospitals are still vulnerable to infection .
Orig	wenn ein patient ist , die an das krankenhaus , eines der ersten themen zu prüfen , für die präsenz von mrsa , aber die krankenhäuser sind immer noch anfällig für infektionen .
Man	wenn ein patient ist , die an das krankenhaus . eine der ersten dinge , die zu prüfen , für die präsenz von mrsa .

	aber die krankenhäuser sind immer noch anfällig für infektionen .
Ref	bei ihrer einlieferung ins krankenhaus werden die patienten zwar auf mrsa untersucht , eine infektion kann dennoch nicht immer vermieden werden .

Table 6.6: Manual simplification and original translation examples for English-German.

Table 6.6 shows examples of manual simplification translation. We present examples in cases where the proposed model works and does not work well. We show the English original source sentence (Src), manual simplification (MS), translation of the original English (Orig), manual simplification translation (Man), and translation reference (Ref). Positive examples are cases when our annotators agree that Man is better than Orig. Meanwhile, negative examples show cases manual simplification does not provide better translations than original text.

		Annotator #2		
		Man	Orig	sum(#1)
Annotator #1	Man	**9**	10	19
	Orig	7	**44**	51
	sum(#2)	16	54	

Table 6.7: Confusion matrix from human evaluation for automatic simplification translation

We further investigate the question to see if automatic simplification will be helpful for machine translation. In stead of translating manual simplification, we translation the automatic simplification generated by our proposed sentence simplification. We repeat the same human evaluation experiment as performed with manual simplification. The inter-annotator reliability is 0.35 which indicates a fair agreement between annotators. The annotators prefer original simplification translation in over **80%** of the test sentences, and prefer the automatic simplification translation in less than **20%** of the test sentences.

6.8 Summary

In this chapter we proposed a novel method for sentence simplification based on log-linear models. Our major contributions are the stack decoding algorithm, the discriminative training algorithm, and the 177 feature functions within the model. We have presented insight the analyses of our model in controlled settings to show the impact of different model hyper parameters. We demonstrated that the proposed model outperforms a state-of-the-art rule-based system on ROUGE-2, ROUGE-4, and $AveF_{10}$ by **0.2**, **0.6**, and **4.5** points, respectively. Subjective translation evaluations show that **63%** sentences with **manual** simplification translations are better than the original translation. Meanwhile, when applying **automatic** simplification translations **20%** sentences are better than the original translation.

Chapter 7

Conclusions

In this chapter we conclude the dissertation by summarizing the thesis work and proposing several directions for future research.

7.1 Summary

We develop various algorithms to statistically incorporate dependency structures into MT components including the decoder, reordering models, confidence measure, and sentence simplification. We achieve improved BLEU and TER scores, increased MT translation quality prediction accuracy, and reduced the hardness of source sentences. We adopt the phrase-based MT system as our baseline. With different resources and different problems to solve, we first expand the baseline system in the following ways:

- Decoder: Given the source dependency tree we want to enforce the cohesive decoding strategy. We proposed four novel cohesive soft constraints namely exhaustive interruption check, interruption count, exhaustive interruption count, and rich interruption count. The cohesive-enhanced decoder performs statistically significant better than the standard phrase-based decoder on English-Spanish. Improvements in between **+0.4** and **+1.8** BLEU points are also obtained on English-Iraqi, Arabic-

English and Chinese-English systems.

- Reordering Models: To go beyond cohesive soft constraints, we investigate efficient algorithms for learning and decoding with source-side dependency tree reordering models. The phrase movements can be viewed as the movement of the subtree $inside$ or $outside$ a source subtree when the decoder is leaving from the previous source state to the current source state. The notions of moving $inside$ and $outside$ a subtree can be interpreted as tracking facts about the subtree-to-subtree transitions observed in the source side of word-aligned training data. With extra guidance on subtree movements, the source-tree reordering models help the decoder make smarter distortion decisions. We observe improvements of **+0.8** BLEU and **-1.4** TER on English-Spanish and **+0.8** BLEU and **-2.3** TER on English-Iraqi.

For confidence measure, we proposed $Goodness$, a method to predict confidence scores for machine translated words and sentences based on a feature-rich classifier using structure features. We develop three novel feature sets to capture different aspects of translation quality which have never been considered during the decoding time, including:

- Source and target dependency structure features that enable the classifier to utilize deep structures to predict translation errors.
- Source POS and phrase features which capture the surface source word context.
- Alignment context features that use both source and target word collocation for judging translation quality.

Experimental results show that by combining the dependency structure, source side information, and alignment context features with word posterior probability and target POS context the MT error prediction accuracy is increased from **69.1** to **72.2** in F-score. Our framework is able to predict error types namely insertion, substitution and shift. The Pearson correlation with human judgment increases from **0.52** to **0.6**. Furthermore, we show that $Goodness$ can help the MT system to select better translations and as a result improvements between **0.4** and **0.9** TER reduction are obtained. We develop a visualization prototype using different font sizes and colors to catch the attention of post-editors whenever translation errors are likely to appear.

Finally, we develop $TriS$, a statistical sentence simplification system with log-linear models, to simplify source sentence before translating them. In contrast to state-of-the-art methods that drive sentence simplification process by hand-written linguistic rules, our method used a margin-based discriminative learning algorithm operates on a feature set. We decompose the original dependency tree into context dependency structures and incorporate them as feature functions in the proposed model. The other feature functions are defined on statistics of surface form as well as syntactic of the sentences. A stack decoding algorithm is developed to allow us to efficiently generate and search simplification hypotheses. The simplified text produced by the proposed system reduces **1.7** Flesch-Kincaid education level when compared with the original text. We show that a comparison of a state-of-the-art rule-based system to the proposed system demonstrates an improvement of **0.2**, **0.6**, and **4.5** points in ROUGE-2, ROUGE-4, and $AveF_{10}$, respectively. Subjective translation evaluations show that **63%** sentences with **manual** simplification translations are better than the original translation. Meanwhile, when applying **automatic** simplification translations **20%** sentences are better than the original translation.

7.2 Conclusion

Dependency structures are important linguistic resources that bring long distance dependency between words to local and represent the semantic relation between words. In this thesis work, we mainly focus on modeling and incorporating dependency structures into statistical machine translation systems. We draw the following conclusions from our thesis work:

1. Cohesive soft constraints can benefit machine translations. This claim is supported by experiments that cover a wide range of training corpus sizes, ranging from 500K sentence pairs up to 10 million sentence pairs. Furthermore, the effectiveness of

our proposed methods was shown when we applied them to systems using a 2.7 billion word 5-gram LM, different reordering models and dependency parsers. All five cohesive constraints give positive results. We observed a consistent pattern indicating that the observed improvements are stable across test sets.

2. Effectively exploiting dependency subtree movements and cohesive constraints, source-tree reordering models substantially improve translation quality. These movements and constraints enable us to efficiently capture the subtree-to-subtree transitions observed both in the source of word-aligned training data and in decoding time. Providing more features that are conditioned on different information, such as include $inside/outside$ subtree movements and cohesive constraints, benefit the system performance Moreover, further improvement can be obtained by enabling MERT to choose the most prominent ones from a larger basis.

3. The proposed confidence estimation method is capable to predict the quality of machine translated words and sentences based on a feature-rich classifier using dependency structures and context features. It is also able to predict translation error types namely insertion, substitution, and shift. The proposed confidence estimation method correlates well with the human judgment. The core MT engine can benefit from the proposed method in the n-best list reranking task.

4. For sentence simplification, a log-linear model equipped with a stack decoding algorithm, a discriminative training algorithm, and 177 dependency structure and syntactic feature functions is capable to produce better simplification candidates than a rule-based system. When applying to machine translation, subjective evaluation results suggest that machine translation quality is benefit from manual and automatic simplifications.

7.3 Discussion and Future Research Directions

Although we have developed a series of approaches to statistically model and incorporate dependency structures into machine translation systems, this problem has not been fully solved. There remain many intriguing research problems which can be further explored. Here we propose some possible directions for future research:

7.3.1 Improve Reordering Models

The cohesive soft constraints and the source-side dependency tree reordering model are implemented around the interruption check in order to encourage finishing a subtree before translating something else. It is very effective for phrase-based decoding which searches over an entire space within the distortion limit in order to advance a hypothesis. However, it is not straightforward to apply the models and constraints to a bottom-up chart-based decoding algorithm since the hierarchical model already conducts principled reordering search with synchronous rules. One may combine our models with the hierarchical phrase reordering model (Galley and Manning, 2008) by extending the parameterization of our models to explicitly represent source-side subtree movements during the decoding time. Moreover, one can take advantage from our analysis and design novel dependency constraints. An example of this line is the work done by Gao et al. (2011). We believe such extensions will generalize more subtle reordering events on source dependency trees.

7.3.2 Improve Confidence Estimation

Confidence estimation is emerging as a vital component for the success of commercialized machine translation when there is no availability of the reference translations. Our work on $Goodness$ can be expanded in several directions. First, one can apply confidence estimation to perform a second-pass constraint decoding. After the first pass decoding, our confidence estimation model can label which word is likely to be correctly translated. The second-pass decoding utilizes the confidence information to constrain the search space and hopefully can find a better hypothesis than in the first pass. This idea is very similar to the multi-pass decoding strategy employed by speech recognition engines or the coarse-to-fine strategy in parsing.

Another idea is to test different visualization strategies to see if it truly benefits the customers. One may perform a user study on our visualization prototype to see if it increases the productivity of post-editors. In addition, our work based on a large manually collected training data which is system-dependent and not always available in other language pairs. One can work on the problem of building the confidence estimation in a cheaper way. There are some work following in these directions recently for example Popović et al. (2011) and Specia et al. (2011).

7.3.3 Improve Automatic Sentence Simplification

Our work shares the same line of research with Klebanov et al. (2004); Heilman and Smith (2010) in which we all focus on sentence-level factual simplification. However, a major focus of our work is on log-linear models which offer a new perspective for sentence simplification on decoding, training, and modeling problems. To contrast, consider rule-based systems Klebanov et al. (2004); Daelemans et al. (2004); Siddharthan (2006); Heilman and Smith (2010), in which sentence simplification processes are driven by hand-written linguistic rules. The linguistic rules represent prior information about how each word and phrase can be restructured. In our model, each linguistic rule is encoded as a feature function and we allow the model to learn the optimized feature weights based on the nature of training data. A potential issue is the proposed model might be susceptible to the sparseness issue. We alleviated this issue by using structure level and count feature functions which are lexically independent.

There are some directions to expand in this area. Our model can generate repeatedly noun phrases repeatedly in multiple simple sentences, one may augment the proposed model to cope with anaphora resolution. Lexical simplification is another direction since we focus on structure simplification. To address the data sparsity issue, one may use crowd-sourcing such as Amazon Mechanical Turk to collect more training data.

Related to application for machine translation, we think the most important issue is to know which sentences should be simplified before translating them. An approach one may try it to build a high-precision binary classifier to classify a source sentence with labels of simplify/not-simplify. We evaluated the impact of sentence simplification on machine translation using subjective human evaluation. However, it is not necessary the best suitable way. One may consider an evaluation strategy which is more information-oriented and efficient-oriented. For example, employ a questionnaire on the target translation output and based on answers, one can measure the time and the number of correct answers. We believe such extensions will bring more value of the sentence simplification to machine translation.

Bibliography

Abhaya Agarwal and Alon Lavie. Meteor, M-BLEU and M-TER: Evaluation metrics for high-correlation with human rankings of machine translation output. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 115–118, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Yaser Al-Onaizan and Kishore Papineni. Distortion models for statistical machine translation. In *Proceedings of ACL-COLING'06*, pages 529–536, Sydney, Australia, 2006.

Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9. Association for Computational Linguistics, 2010.

Nguyen Bach, Matthias Eck, Paisarn Charoenpornsawat, Thilo Kohler, Sebastian Stuker, ThuyLinh Nguyen, Roger Hsiao, Alex Waibel, Stephan Vogel, Tanja Schultz, and Alan Black. The CMU TransTac 2007 Eyes-free and Hands-free Two-way Speech-to-Speech Translation System. In *Proceedings of the IWSLT'07*, Trento, Italy, 2007.

Nguyen Bach, Qin Gao, and Stephan Vogel. Source-side dependency tree reordering models with subtree movements and constraints. In *Proceedings of the Twelfth Machine Translation Summit (MTSummit-XII)*, Ottawa, Canada, August 2009a. International Association for Machine Translation.

Nguyen Bach, Stephan Vogel, and Colin Cherry. Cohesive constraints in a beam search phrase-based decoder. In *Proceedings of NAACL-HLT'09*, Boulder, Colorado, May/June 2009b. Association for Computational Linguistics.

Nguyen Bach, Qin Gao, Stephan Vogel, and Alex Waibel. TriS: A statistical sentence simplifier with log-linear models and margin-based discriminative training. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, ChiangMai, Thailand, November 2011a. Association for Computational Linguistics.

Nguyen Bach, Fei Huang, and Yaser Al-Onaizan. Goodness: A method for measuring machine translation confidence. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 211–219, Portland, Oregon, USA, June 2011b. Association for Computational Linguistics.

Regina Barzilay and Kathleen R. McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31, September 2005.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. Confidence estimation for machine translation. In *The JHU Workshop Final Report*, Baltimore, Maryland, USA, April 2004.

Phil Blunsom and Trevor Cohn. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 65–72, Sydney, Australia, July 2006. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331, 1993.

Xavier Carreras and Michael Collins. Non-projective parsing for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 200–209, Singapore, August 2009. Association for Computational Linguistics.

Raman Chandrasekar and Bangalore Srinivas. Automatic induction of rules for text simplification. *Knowledge Based Systems*, 10(3):183–190, 1997.

Raman Chandrasekar, Doran Christine, and Bangalore Srinivas. Motivations and methods for text simplification. In *Proceedings of 16th International Conference on Computational Linguistics*, pages 1041–1044. Association for Computational Linguistics, 1996.

Pi-Chuan Chang and Kristina Toutanova. A discriminative syntactic word order model for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 9–16, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. Discriminative reordering with chinese grammatical relations features. In *Proceedings of NAACL-HLT'09: Third Workshop on Syntax and Structure in Statistical Translation (SSST-3)*, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Colin Cherry. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 72–80, Columbus, Ohio, June 2008. Association for Computational Linguistics.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL'05*, pages 263–270, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2): 201–229, June 2007.

David Chiang, Kevin Knight, and Wei Wang. 11,001 new features for statistical machine translation. In *Proceedings of HLT-ACL*, pages 218–226, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, September 1956.

James Clarke. *Global Inference for Sentence Compression: An Integer Linear Programming Approach*. PhD thesis, University of Edinburgh, 2008.

Shay B. Cohen, Kevin Gimpel, and Noah A. Smith. Logistic normal priors for unsupervised probabilistic grammar induction. In *Advances in Neural Information Processing Systems 22 NIPS 2008*, pages 321–328. MIT Press, 2008.

Trevor Cohn and Mirella Lapata. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

Michael Collins. Head-driven statistical models for natural language parsing. In *PhD Thesis*, USA, 1999. The University of Pennsylvania.

Michael Collins, Philipp Koehn, and Ivona Kucerova. Clause restructuring for statistical machine translation. In *Proceedings of ACL'05*, pages 531–540, Ann Arbor, USA, June 2005.

Brooke Cowan, Ivona Kučerová, and Michael Collins. A discriminative model for tree-to-tree translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 232–241, Sydney, Australia, July 2006. Association for Computational Linguistics.

Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003. ISSN 1532-4435.

Walter Daelemans, Anja Höthker, and Erik Tjong Kim Sang. Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-04)*, pages 1045–1048, 2004.

Steve DeNeefe and Kevin Knight. Synchronous tree adjoining machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 727–736, Singapore, August 2009. Association for Computational Linguistics.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. What can syntax-based MT learn from phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 755–763, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

John DeNero and Dan Klein. The complexity of phrase alignment problems. In *Proceedings of ACL-08: HLT*, pages 25–28, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Yuan Ding and Martha Palmer. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 541–548, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

Jason Eisner. Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208, Sapporo, Japan, July 2003. Association for Computational Linguistics.

Gunes Erkan, Arzucan Ozgur, and Dragomir R. Radev. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 228–237, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Rudolf Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

Rudolf Flesch. *How to write plain English.* Barnes & Noble, 1981.

Heidi J. Fox. Phrasal cohesion and statistical machine translation. In *Proceedings of EMNLP'02*, pages 304–311, Philadelphia, PA, July 6-7 2002.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelEx - relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007. ISSN 1367-4803.

Haïm Gaifman. Dependency systems and phrase-structure systems. *Information and Control*, 4, 1965.

Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of EMNLP'08*, Hawaii, USA, 2008.

Michel Galley and Kathleen McKeown. Lexicalized Markov grammars for sentence compression. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 180–187, Rochester, New York, April 2007. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proceedings of HLT-NAACL'04*, Boston, USA, May 2004.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July 2006. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. Parallel implementations of word alignment tool. In *Proceedings of the ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*, Columbus, Ohio, USA, 2008.

Yang Gao, Philipp Koehn, and Alexandra Birch. Soft dependency constraints for reordering in hierarchical phrase-based translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 857–868, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Niyu Ge. Max-posterior HMM alignment for machine translation. In *Presentation given at DARPA/TIDES NIST MT Evaluation workshop*, 2004.

Kevin Gimpel and Noah A. Smith. Feature-rich translation by quasi-synchronous lattice parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 219–228, Singapore, August 2009. Association for Computational Linguistics.

Ben Goertzel, Hugo Pinto, Ari Heljakka, Michael Ross, Cassio Pennachin, and Izabela Goertzel. Using dependency parsing and probabilistic inference to extract relationships between genes, proteins and malignancies implicit among multiple biomedical research abstracts. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 104–111, New York, New York, June 2006. Association for Computational Linguistics.

Robert Gunning. *The technique of clear writing*. McGraw-Hill New York, NY, 1968.

Nizar Habash and Jun Hu. Improving arabic-chinese statistical machine translation using english as pivot language. In *Proceedings of the 4th Workshop on Statistical Machine Translation*, pages 173–181, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

David G. Hays. Dependency theory: A formalism and some observations. In *United State Air Force Project RAND*. The RAND Corporation, 1964.

William P. Headden III, Mark Johnson, and David McClosky. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Michael Heilman and Noah Smith. Extracting simplified statements for factual question generation. In *Proceedings of the 3rd Workshop on Question Generation.*, 2010.

Almut Silja Hildebrand and Stephan Vogel. Combination of machine translation systems via hypothesis selection from combined n-best lists. In *Proceedings of the 8th Conference of the AMTA*, pages 254–261, Waikiki, Hawaii, October 2008.

Mark Hopkins and Jonathan May. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Fei Huang. Confidence measure for word alignment. In *Proceedings of the ACL-IJCNLP '09*, pages 932–940, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

Liang Huang and David Chiang. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Liang Huang, Hao Zhang, Daniel Gildea, and Kevin Knight. Binarization of synchronous context-free grammars. *Computational Linguistics*, 35(4), December 2009.

Abraham Ittycheriah and Salim Roukos. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of the HTL-EMNLP'05*, pages 89–96, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

Frederick Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, 1998.

Siddhartha Jonnalagadda. Syntactic simpli?cation and text cohesion. Technical report, University of Cambridge, 2006.

Siddhartha Jonnalagadda and Graciela Gonzalez. Sentence simplification aids protein-protein interaction extraction. *CoRR*, 2010.

Siddhartha Jonnalagadda, Luis Tari, Jörg Hakenberg, Chitta Baral, and Graciela Gonzalez. Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 177–180, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Jeremy G. Kahn, Mari Ostendorf, and Brian Roark. Automatic syntactic mt evaluation with expected dependency pair match. In *Proceedings of AMTA 2008, workshop MetricsMATR: NIST Metrics for Machine Translation Challenge*, Waikiki, Hawaii, US, October 2008. AMTA.

Peter Kincaid, Robert Fishburne, Richard Rogers, and Brad Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel (Research Branch Report 8-75). *Memphis, TN: Naval Air Station*, 1975.

Beata Klebanov, Kevin Knight, and Daniel Marcu. Text simplification for information seeking applications. In *On the Move to Meaningful Internet Systems, Lecture Notes in Computer Science*, volume 3290, pages 735–747, Morristown, NJ, USA, 2004. Springer-Verlag.

Kevin Knight. Squibs and discussions: Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615, December 1999.

Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91 – 107, 2002.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press; 1st edition, 2010.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based machine translation. In *Proceedings of HLT-NAACL'03*, pages 48–54, Edmonton, Canada, 2003.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the IWSLT'05*, 2005.

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 *Proceedings of ACL'07*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. *Dependency Parsing*. Morgan & Claypool Publishers, 2009.

Roland Kuhn, Denis Yuen, Michel Simard, Patrick Paul, George Foster, Eric Joanis, and Howard Johnson. Segment choice models: Feature-rich models for global distortion in statistical machine translation. In *Proceedings of HLT-NAACL'06*, pages 25–32, New York, NY, 2006.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. An end-to-end discriminative approach to machine translation. In *Proceedings of ACL'06*, pages 761–768, Sydney, Australia, 2006a. Association for Computational Linguistics.

Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June 2006b. Association for Computational Linguistics.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004a. Association for Computational Linguistics.

Dekang Lin. A path-based transfer model for machine translation. In *Proceedings of Coling 2004*, pages 625–630, Geneva, Switzerland, Aug 23–Aug 27 2004b. COLING.

Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string alignment template for statistical machine translation. In *Proceedings of ACL'06*, pages 609–616, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

Yang Liu, Yajuan Lü, and Qun Liu. Improving tree-to-tree translation with packed forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 558–566, Suntec, Singapore, August 2009. Association for Computational Linguistics.

Yudong Liu, Zhongmin Shi, and Anoop Sarkar. Exploiting rich syntactic information for relation extraction from biomedical articles. In *Proceedings of NAACL'07*, pages 97–100, Morristown, NJ, USA, 2007. Association for Computational Linguistics.

Yanjun Ma, Sylwia Ozdowska, Yanli Sun, and Andy Way. Improving word alignment using syntactic dependencies. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 69–77, Columbus, Ohio, June 2008. Association for Computational Linguistics.

David M. Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 276–283, Cambridge, Massachusetts, USA, June 1995. Association for Computational Linguistics.

Marie-Catherine Marneffe, Bill MacCartney, and Christopher Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC'06*, Genoa, Italy, 2006.

Ryan McDonald. Discriminative sentence compression with soft syntactic evidence. In *Proceedings 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 297–304, 2006.

Ryan McDonald, Koby Crammer, and Fernando Pereira. Flexible text segmentation with structured multilabel classification. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 987–994, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

Hary McLaughlin. SMOG grading: A new readability formula. *Journal of Reading*, 12 (8):639–646, 1969.

Yashar Mehdad and Bernardo Magnini. Optimizing textual entailment recognition using particle swarm optimization. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 36–43, Suntec, Singapore, August 2009. Association for Computational Linguistics.

Arul Menezes and Chris Quirk. Using dependency order templates to improve generality in translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 1–8, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Haitao Mi and Qun Liu. Constituency to dependency translation with forests. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1433–1442, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Courtney Napoles and Mark Dredze. Learning simple wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, pages 42–50, Los Angeles, CA, USA, June 2010. Association for Computational Linguistics.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, 4, May 2007. ISSN 1550-4875.

Rebecca Nesson, Giorgio Satta, and Stuart M. Shieber. Optimal k-arization of synchronous tree-adjoining grammar. In *Proceedings of ACL-08: HLT*, pages 604–612, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Jan Niehues and Stephan Vogel. Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC'06*, Genoa, Italy, 2006.

Franz J. Och and Hermann Ney. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 1:29, pages 19–51, 2003.

Franz J. Och and Hermann Ney. The alignment template approach to statistical machine translation. In *Computational Linguistics*, volume 30, pages 417–449, 2004.

Franz Josef Och. Minimum error rate training in statistical machine translation. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of ACL'03*, pages 160–167. Association for Computational Linguistics, 2003.

Karolina Owczarzak, Josef van Genabith, and Andy Way. Dependency-based automatic evaluation for machine translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 80–87, Rochester, New York, April 2007. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, PA, July 2002.

Maja Popović, David Vilar, Eleftherios Avramidis, and Aljoscha Burchardt. Evaluation without references: Ibm1 scores as evaluation metrics. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 99–103, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

Chris Quirk. Training a sentence-level machine translation confidence measure. In *Proceedings of the 4th LREC*, 2004.

Chris Quirk and Simon Corston-Oliver. The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of EMNLP'06*, Sydney, Australia, 2006.

Chris Quirk, Aruk Menezes, and Colin Cherry. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of ACL'05*, pages 271–279, Ann Arbor, USA, June 2005.

Lance Ramshaw and Mitchell Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, MIT, June 1995.

Sylvain Raybaud, Caroline Lavecchia, David Langlois, and Kamel Smaili. Error detection for statistical machine translation using linguistic features. In *Proceedings of the 13th EAMT*, Barcelona, Spain, May 2009.

Jane J. Robinson. Methods for obtaining corresponding phrase structure and dependency grammars. In *Proceedings of the 1967 Conference on Computational Linguistics*, pages 1–25, Morristown, NJ, USA, 1967. Association for Computational Linguistics.

Jane J. Robinson. A dependency structures and transformational rules. *Language*, 46(2): 259–285, June 1970.

Kay Rottmann and Stephan Vogel. Word reordering in statistical machine translation with a pos-based distortion model. In *Proceedings of TMI-11*, pages 171–180, Sweden, 2007.

Binyamin Rozenfeld, Ronen Feldman, and Moshe Fresko. A systematic cross-comparison of sequence classifiers. In *Proceedings of the SDM*, pages 563–567, Bethesda, MD, USA, April 2006.

Alberto Sanchis, Alfons Juan, and Enrique Vidal. Estimation of confidence measures for machine translation. In *Proceedings of the MT Summit XI*, Copenhagen, Denmark, 2007.

Libin Shen, Jinxi Xu, and Ralph Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Libin Shen, Jinxi Xu, Bing Zhang, Spyros Matsoukas, and Ralph Weischedel. Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 72–80, Singapore, August 2009. Association for Computational Linguistics.

Hideki Shima, Ni Lao, Eric Nyberg, and Teruko Mitamura. Complex Cross-lingual Question Answering as Sequential Classification and Multi-Document Summarization Task. In *Proceedings of NTCIR-7 Workshop*, Japan, 2008.

Advaith Siddharthan. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109, June 2006.

David Smith and Jason Eisner. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 23–30, New York City, June 2006. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA'06*, pages 223–231, August 2006.

Radu Soricut and Abdessamad Echihabi. Trustrank: Inducing trust in automatic translations via ranking. In *Proceedings of the 48th ACL*, pages 612–621, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Lucia Specia, Zhuoran Wang, Marco Turchi, John Shawe-Taylor, and Craig Saunders. Improving the confidence of machine translation quality estimates. In *Proceedings of the MT Summit XII*, Ottawa, Canada, 2009.

Lucia Specia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. Predicting machine translation adequacy. In *Proceedings of the MT Summit XIII*, Xiamen, China, 2011.

Andreas Stolcke. SRILM – An extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 901–904, Denver, 2002.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. Cross-lingual information extraction system evaluation. In *Proceedings of COLING '04*, page 882, Geneva, Switzerland, 2004. Association for Computational Linguistics.

Christoph Tillman. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL'04*, pages 101–104, 2004.

Christoph Tillmann. Efficient dynamic programming search algorithms for phrase-based SMT. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 9–16, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

Roy Tromble and Jason Eisner. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August 2009. Association for Computational Linguistics.

Jenine Turner and Eugene Charniak. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 290–297. Association for Computational Linguistics, 2005.

Nicola Ueffing and Hermann Ney. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40, 2007. ISSN 0891-2017.

Ashish Venugopal and Stephan Vogel. Considerations in maximum mutual information and minimum classification error training for statistical machine translation. In *Proceedings of EAMT-05*, Budapest, Hungary, 2005.

Ashish Venugopal, Andreas Zollmann, and Vogel Stephan. An efficient two-pass approach to synchronous-CFG driven statistical MT. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 500–507, Rochester, New York, April 2007. Association for Computational Linguistics.

David Vilar, Jia Xu, Luis F. DH́aro, and Hermann Ney. Error analysis of statistical machine translation output. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 697–702, Genova, Italy, 2006.

Stephan Vogel. SMT decoder dissected: Word reordering. In *Proceedings of NLP-KE'03*, pages 561–566, Bejing, China, Oct. 2003.

Stephan Vogel. PESA: Phrase pair extraction as sentence splitting. In *Proceedings of the Tenth Machine Translation Summit (MTSummit-X)*, Phuket, Thailand, August 2005. International Association for Machine Translation.

Chao Wang, Michael Collins, and Philipp Koehn. Chinese syntactic reordering for statistical machine translation. In *Proceedings of EMNLP'07*, pages 737–745, 2007.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of the EMNLP-CoNLL*, pages 764–773, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

W.M. Watanabe, A.C. Junior, V.R. Uzêda, R.P.M. Fortes, T.A.S. Pardo, and S.M. Aluísio. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM International Conference on Design of communication*, pages 29–36. ACM, 2009.

Kristian Woodsend and Mirella Lapata. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. In *Computational Linguistics*, volume 23(3), pages 377–403, 1997.

Yuanbin Wu, Qi Zhang, Xuangjing Huang, and Lide Wu. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1541, Singapore, August 2009. Association for Computational Linguistics.

Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of COLING'04*, pages 508–514, Geneva, Switzerland, August 2004.

Fei Xia and Martha Palmer. Converting dependency structures to phrase structures. In *Proceedings of the first international conference on Human Language Technology research (HLT'01)*, San Diego, 2001. Association for Computational Linguistics.

Deyi Xiong, Min Zhang, and Haizhou Li. Error detection for statistical machine translation using linguistic features. In *Proceedings of the 48th ACL*, pages 604–611, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of NAACL-HLT'09*, pages 245–253, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Hirofumi Yamamoto, Hideo Okuma, and Eiichiro Sumita. Imposing constraints from the source tree on ITG constraints for SMT. In *Proceedings of ACL-08:HLT, SSST-2*, pages 1–9, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Victor H Yngve. A framework for syntactic translation. *Mechanical Translation*, 4(3), July 1958.

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. Reordering constraints for phrase-based statistical machine translation. In *Proceedings of COLING'04*, pages 205–211, Geneva, Switzerland, August 2004.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Ying Zhang. Structured language models for statistical machine translation. In *PhD Thesis*, Pittsburgh, USA, 2009. Language Technologies Institute, Carnegie Mellon University.

Ying Zhang and Nguyen Bach. Virtual babel: Towards context-aware machine translation in virtual worlds. In *Proceedings of the Twelfth Machine Translation Summit*

(MTSummit-XII), Ottawa, Canada, August 2009. International Association for Machine Translation.

Ying Zhang and Stephan Vogel. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of EAMT'05*, Budapest, Hungary, May 2005. The European Association for Machine Translation.

Ying Zhang, Stephan Vogel, and Alex Waibel. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 2051–2054, 2004.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A monolingual tree-based translation model for sentence simplification. In *Proceedings of The 23rd International Conference on Computational Linguistics*, pages 1353–1361, Beijing, China, Aug 2010.

Andreas Zollmann and Ashish Venugopal. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June 2006. Association for Computational Linguistics.

Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1145–1152, Manchester, UK, August 2008. Coling 2008 Organizing Committee.