UNIVERSITY OF CALIFORNIA, SAN DIEGO

Contributions to Research on Machine Translation

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

David Kauchak

Committee in charge:

Professor Charles Elkan, Chair
Professor Garrison W. Cottrell
Professor Sanjoy Dasgupta
Professor Jeff Elman
Professor Andy Kehler

2006

The dissertation of David Kauchak is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____
Chair

University of California, San Diego

2006

TABLE OF CONTENTS

Signature Page . . . iii

Table of Contents . . . iv

List of Figures . . . vii

List of Tables . . . ix

Acknowledgements . . . xi

Vita and Publications . . . xii

Abstract of the Dissertation . . . xiii

1 Introduction . . . 1
1.1 Translation Improvement . . . 2
1.2 Learning Example Usefulness . . . 5
1.3 Improving Automatic Evaluation Methods . . . 7
1.4 Outline . . . 9

2 Background . . . 10
2.1 Translation Systems . . . 10
2.1.1 Rule-based approaches . . . 10
2.1.2 Data-driven approaches . . . 11
2.1.3 Machine translation evaluation . . . 14
2.1.4 Corpora . . . 15
2.1.5 Software resources . . . 17

3 Learning Word-Level Correction Rules . . . 19
3.1 Introduction . . . 19
3.2 Algorithm . . . 20
3.2.1 Rule format . . . 21
3.2.2 Generating training data . . . 22
3.2.3 Rule learning . . . 26
3.3 Experiments . . . 31
3.3.1 Applying word correction rules . . . 31
3.3.2 Rule precision . . . 32
3.3.3 Using extended word lists . . . 32
3.4 Discussion . . . 34

4 Learning Phrase-Level Correction Rules . . . 36
4.1 Introduction . . . 36
4.2 Related Work . . . 38
4.3 Learning Phrase Rules . . . 39
4.3.1 Rule format . . . 40
4.3.2 Learning an alignment . . . 40
4.3.3 Generating rules . . . 41
4.4 Experiments . . . 42
4.4.1 Experimental setup . . . 42
4.4.2 Improving MT systems . . . 44
4.4.3 Rule filtering analysis . . . 47
4.5 Conclusions and Future Work . . . 48

5 Learning Example Usefulness . . . 51
5.1 Introduction . . . 51
5.2 Related Work . . . 53
5.3 Ranking Examples . . . 55
5.3.1 Generating training data . . . 55
5.3.2 Modeling example contribution . . . 56
5.3.3 Calculating example scores . . . 57
5.3.4 Theoretical justification . . . 57
5.4 Experiments . . . 59
5.4.1 Experimental setup . . . 60
5.4.2 Selecting the most useful examples . . . 60
5.4.3 Analyzing example features . . . 62
5.5 Future Work . . . 64
5.6 Conclusion . . . 66

6 Paraphrasing for Automatic Evaluation . . . 67
6.1 Introduction . . . 67
6.2 Related Work . . . 70
6.3 Algorithm . . . 71
6.4 Experiments . . . 73
6.4.1 Experimental setup . . . 74
6.4.2 Impact of paraphrases on machine translation evaluation . . . 77
6.4.3 Evaluation of paraphrase quality . . . 78
6.5 Conclusion and Future Work . . . 81

7 Contributions and Future Research Directions 83
7.1 Summary of Contributions 83
7.2 Future Research Directions 85

Bibliography 87

LIST OF FIGURES

Figure 1.1 Example output from translating an English word list to Spanish and then back to English. 3
Figure 1.2 Example commercial translations corrected by context-independent phrase rules learned by our method. Changed phrases are in *italics* and replaced by those in **bold**. 4
Figure 1.3 Three foreign/English parallel bilingual examples from Canadian parliamentary proceedings. The first line is French and second English. The third example is an English quote and is therefore the same in French and English. 5
Figure 1.4 Process for generating scored example subsets to use as training data for ranking example usefulness. Random subsets of the examples are selected. Each of these subsets is used to train a translation system, which is then automatically evaluated. 6
Figure 1.5 Four human translations of the same Chinese headline. 7
Figure 1.6 Paraphrases of human reference translations found by our proposed paraphrasing algorithm. The first line is the reference and second paraphrased reference. Paraphrases are shown in **bold**. 8

Figure 2.1 Translation method categorization pyramid. Translation occurs by transferring language information in the foreign language from the left to English language information on the right. 11

Figure 3.1 Example application of the rule g(alquitrán, alquitrán, []) $\rightarrow$ tar. The first sentence is the original Foreign sentence to be translated (in this case, Spanish). The second sentence is the translation made by the commercial system. The final sentence is the translation after the rule is applied. The changed word is in **bold**. 22
Figure 3.2 Outline of algorithm to learn rules to improve foreign-to-English translation. The preprocessing steps generate the initial data for use in learning the rules. The following three sets of steps describe the algorithms for learning the context-independent and context-dependent rules. 23

Figure 4.1 Example French sentence with machine translation (a) and human translation (b). 37
Figure 4.2 Algorithm for determining the cost of the best cost monotone alignment between machine translated sentence S and human translated sentence E. 41

Figure 4.3 Word alignment learned by our method between the human-translated sentence (on top) and machine-translated sentence. Aligned words are denoted by ‘ | ’. 41
Figure 4.4 BLEU scores for the commercial system (a) and the phrase-based system (b) for the 10 different test sets. For each of the systems the translation improved system and phrase rule improved system are also shown. 46
Figure 4.5 BLEU score (a), proportion of rule corrections that are correct changes (b), and the number of rule changes (c) on the first 10,000 test set for different rule filtering thresholds. 49

Figure 5.1 Three foreign/English parallel bilingual examples. The first line is French and second English (reprint of figure 5.1). 52
Figure 5.2 Individual test scores for “best” and “random $\{$1-10$\}$” for the ten different test sets. For clarity, all random scores are marked with the same symbol. 62

LIST OF TABLES

Table 3.1 Exhaustive list of the different possible translation equalities/inequalities given a word (w), the foreign translation of that word ($f(w)$) and the translation back to English ($f'(f(w))$). 24
Table 3.2 Patterns for generating training data for learning rules to improve the foreign-to-English system. For each different translation list equality option, we learn that when "input word" is translated as "current translation" then it should be translated as one of the "correct translations". Most of the information for improving the foreign-to-English translation system is learned from the English words list (English/foreign/English examples), but some information can also be learned from a foreign word list if available (foreign/English/foreign examples). 27
Table 3.3 Summary of results for word correction rules generated from a word list with 45,192 entries applied to the PAHO data set. 32
Table 3.4 Proportion of correct rule changes for both context-independent and context-dependent rules as measured by a native English speaker on 600 random changes. 32
Table 3.5 Summary of results for word correction rules generated using a general word list with 45,192 entries plus 419 learned words and 5,215 domain specific words applied to the PAHO data set. 33

Table 4.1 Sample phrase rules extracted for correcting the commercial system trained on one million sentences. A rule $\tilde{p} \rightarrow p$ changes the phrase $\tilde{p}$ to p. 42
Table 4.2 Average BLEU score and computation time over ten tests for the two different translation systems improved with the two correction methods. "translation" is a trained English to human English statistical phrase-based translation improvement method. "phrase rules" is improvement based on the phrase rules generated by our method. All BLEU scores are significantly different based on a paired t-test over the ten tests, except "phrase none" vs. "phrase translation". 45
Table 4.3 Example sentences corrected by the learned rules. Changed phrases are in *italics* and replaced by those in **bold**. 46
Table 4.4 The number of changes made, number of rules used and proportion of correct and incorrect rule changes all averaged over ten 10,000 sentence test sets. 47

Table 5.1 Average score over the 10 test sets and the paired t-test significance for the "best" system compared to 10 random systems. One, two and three triangles indicate significance at the 95%, 99%, 99.9% confidence level respectively. 61
Table 5.2 Average feature scores for "best" and "random 1". Significantly different averages are shown in **bold** and moderately significant differences in *italics*. 65

Table 6.1 A reference sentence and corresponding machine translation from the NIST 2004 MT evaluation. The two sentences share only auxiliary words. 68
Table 6.2 A reference sentence and a corresponding machine translation. Candidate paraphrases are in bold. 72
Table 6.3 Sample of paraphrasings produced by each method based on the corresponding system translation. Paraphrased words are in **bold** and filtered words underlined. 75
Table 6.4 Pearson adequacy correlation scores for rewriting using one and two references, averaged over ten runs. 77
Table 6.5 Paired t-test significance for all methods compared to BLEU and our method for one reference. Two triangles indicates significant at the 99% confidence level, one triangle at the 95% confidence level and X not significant. Triangles point towards the better method. 78
Table 6.6 Scores and the number of substitutions made for all 1788 segments, averaged over the different MT system translations 79
Table 6.7 Accuracy scores by two human judges and the Kappa coefficient of agreement. 79
Table 6.8 Confusion matrix for the context filtering method on a random sample of 200 examples labeled by the first judge. 80

ACKNOWLEDGEMENTS

There are many people who have helped and supported me through my graduate career. I hope that those who have been there for me through this process already know how much I appreciate them, so I will keep these acknowledgements short (this also avoids the awkward complication of forgetting someone :).

My family and friends were always there to help me through the rough times and to make sure that I had plenty of distractions, which were crucial for me. I can't thank you all enough. I'd also like to thank the many faculty and collaborators, including my committee, that helped me expand my understanding of computer science and learn to be a better researcher. And, of course, I'd like to thank my advisor who served as my guide during grad school.

Throughout my graduate career, I was fortunate to find funding from a wide range of sources. I thank both those organizations as well as those managing the funding who deal with the many bureaucratic issues that allowed me to complete this work.

The text of Chapter 3, in part, is a reprint of the material in:
D. Kauchak and C. Elkan. Learning rules to improve a machine translation system. In *Proceedings of ECML*, pages 205–216, 2003.

and the text in Chapter 6, in part, a reprint of the material in:
D. Kauchak and R. Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of HLT/NAACL*, pages 455–462, 2006.

The dissertation author was the primary researcher and author and the co-authors listed in these publications supervised the research which forms the basis for these chapters.

VITA

2000	B. S., University of Utah
2002	M. S., University of California San Diego
2006	Ph. D., University of California San Diego

PUBLICATIONS

David Kauchak and Regina Barzilay (2006). Paraphrasing for Automatic Evaluation. In *Proceedings of HLT-NAACL*, pages 455–462.

Rasmus E. Madsen, David Kauchak and Charles Elkan (2005). Modeling Word Burstiness Using the Dirichlet Distribution. In *Proceedings of ICML*, pages 489–498.

David Kauchak and Francine Chen (2005). Feature-Based Segmentation of Narrative Documents. In *Proceedings of the ACL Workshop on Feature Engineering and Machine Learning in NLP*, pages 32–39.

David Kauchak, Joseph Smarr and Charles Elkan (2004). Sources of Success for Boosted Wrapper Induction. In *Journal of Machine Learning Research*, pages 499–527.

David Kauchak and Sanjoy Dasgupta (2003). An Iterative Improvement Procedure for Hierarchical Clustering. In *Proceedings of NIPS*, pages 481–488.

David Kauchak and Charles Elkan (2003). Learning Rules to Improve a Machine Translation System. In *Proceedings of ECML*, pages 205–216.

David Kauchak, Joseph Smarr and Charles Elkan (2002). Sources of Success for Information Extraction Methods. Technical Report No. CS2002-0696, January 2002, UCSD.

ABSTRACT OF THE DISSERTATION

Contributions to Research on Machine Translation

by

David Kauchak
Doctor of Philosophy in Computer Science

University of California San Diego, 2006

Professor Charles Elkan, Chair

In the past few decades machine translation research has made major progress. A researcher now has access to many systems, both commercial and research, of varying levels of performance. In this thesis, we describe different methods that leverage these pre-existing systems as tools for research in machine translation and related fields.

We first examine techniques for improving a translation system using additional text. The first method uses a monolingual corpus. Discrepancies are identified by translating a word list to a foreign language and back again. Entries where the original word and its double translation differ are used to learn word-level correction rules. The second method uses parallel bilingual data consisting of source language/target language sentence pairs. The source sentences are translated using a translation system, and a partial alignment is identified between the machine-translated sentences and the corresponding human-translated sentences in the target language. This alignment is used to generate phrase-level correction rules. Experimentally, both word-level and phrase-level correction rules result in improved translation performance. The learned word-level correction rules make 24,235 corrections on 20,000 Spanish to English translated sentences, with high accuracy. The learned phrase-level rules improve the translation performance (as measured by BLEU) of a French to English commercial system by 30%, and of a state of the art phrase-based system in a statistically significantly way.

To train current statistical machine translation systems, bilingual examples of parallel sentences are used. Generating this data is costly, and currently feasible only in limited domains and languages. A fundamental question is whether every potential example is equally useful. We describe a ranking method for examples that scores individual sentence pairs based on the performance of translation systems trained on random subsets of the examples. When used to train a translation system, the top ranking examples result in a significantly better performing system than random selection of examples. Given these ranked examples, a model of example usefulness can potentially be learned to select the most useful unlabeled examples. Initial experiments show two previously used example features are good candidates for identifying useful examples.

In the last part of this thesis we describe how automatic paraphrasing methods can be used to improve the accuracy of evaluation measures for machine translation. Given a human-generated reference sentence and a machine-generated translated sentence, we present a method that finds a paraphrase of the reference sentence that is closer in wording to the machine output than the original reference is. We show that using paraphrased reference sentences for evaluating a translation system output results in better correlation with human judgement of translation adequacy than using the original reference sentences.

1

Introduction

Machine translation systems are commonplace. A search of the web reveals many commercial translation systems available in a wide array of languages[1]. In addition, over a dozen different research systems are currently being developed[2]. In this thesis, we examine the use of these pre-existing machine translation (MT) systems as research tools for improving and analyzing MT and related fields.

All translation systems have the same goal: translate text in one language into text in a second language. The methods by which they accomplish this goal are different. Most commercial systems use a set of dictionaries containing translation, semantic, syntactic and morphological information in combination with human-generated rules to translate. This results in efficient, general-purpose translation systems. Recently, data-driven methods have become increasingly popular. Given a set of parallel bilingual sentences, these methods learn a probabilistic translation model. Given a foreign sentence, the translation process is a search for the most likely English sentence according to the model.

Most research in machine translation focuses on building a better probabilistic model. Each year adaptations to previous models are suggested based on experimental inade-

[1] Afrikaans, Bulgarian, Chinese, Dutch, English, Finnish, French, German, Greek, Hungarian, Icelandic, Italian, Japanese, Malay, Norwegian, Korean, Polish, Portuguese, Russian, Serbian, Slovenian, Spanish, Swahili, Swedish, Tswana, Ukrainian and Welsh.

[2] http://www.nist.gov/speech/tests/mt/mt05eval_official_results_release_20050801_v3.html

quacies and linguistic intuition. In this thesis, we take a different approach. We leverage existing translation systems both to build better systems and to explore research questions related to translation. Initially, we only assume that we can translate foreign text with a system. Using additional text resources, we explore two different methods for learning correction rules. Research systems are trainable on bilingual data. Using this additional functionality, we then investigate example usefulness for training a translation system. Finally, we use the output of many translation systems to examine paraphrasing for automatic evaluation measures. In this chapter we give a brief overview of the problems and methods discussed throughout this thesis.

1.1 Translation Improvement

One use of a translation system is to identify current translation mistakes. This data can then be used to improve the performance of that system. By translating text where the correct translation is known, differences between the machine translation and this ground truth point to possible mistakes made by the translation system. The advantage of this type of approach is that it does not rely on knowledge of how the system translates and only assumes access to the translation system.

We examine the translation improvement problem for both commercial systems and statistical phrase-based systems. Although in some domains commercial systems tend to produce inferior translations, they have other benefits. Commercial translation systems are general-purpose and work well in many domains where little training data is available for statistical systems. Also, commercial systems are very efficient and translate orders of magnitude faster. Finally, commercial systems tend to be more robust than research systems, which can fail to translate problematic texts. Because of the varied uses of translation systems, we consider improving both commercial and statistical systems.

The first improvement method we examine only uses monolingual text, one of the most prevalent natural language resources. Given English text, we generate an English

English	translated foreign	translated English
dog	perro	dog
scroll	rollo	rollo
abstractness	abstractness	abstractness
metro	metro	meter
cupful	taza	cup

Figure 1.1 Example output from translating an English word list to Spanish and then back to English.

word list. We then translate this list to a foreign language and back to English. This results in triplets of English, foreign and double-translated English. Figure 1.1 shows example triplets translated using the SDL International translation system. Depending on the knowledge in the translation systems, different scenarios arise. Only the first example has both the original and double-translated words equal. Entries where the original word is not the same as the double-translated word suggest possible mistakes. Using these entries, we generate a list of foreign words and possible English translation options. Then, using an English corpus, we learn word-level correction rules. In cases where only one English translation option exists or one option is predominant, we learn a context-independent rule. For ambiguous words, words that co-occur with the alternate translation options are identified using a likelihood ratio significance test. Context-dependent rules are constructed using these significant words to select between the different translation options.

This method learns correction rules based on knowledge differences between the foreign to English translation system and the English to foreign translation system. By itself, monolingual data does not contain translation information. Translation information is available in parallel bilingual data, which consists of foreign sentences and the English translations of those sentences. This data is the building block for statistical translation systems.

Parallel bilingual data can also be used to identify and correct the mistakes of a translation system. By translating the foreign text to English, we again obtain a parallel

Commercial:	The *meeting begins again* at 8 hours.
Rule corrected:	The **House resumed** at 8 hours.
Commercial:	I find interesting to note that certain members of *the American Congresses* seem to divide this opinion, but contrary.
Rule corrected:	I find interesting to note that certain members of **Congress** seem to divide this opinion, but contrary.
Commercial:	Is the *Room lends to decide*?
Rule corrected:	Is the **House ready for the question**?

Figure 1.2 Example commercial translations corrected by context-independent phrase rules learned by our method. Changed phrases are in *italics* and replaced by those in **bold**.

data set with machine-translated English and human-translated English. Differences between these sentence pairs identify possible mistakes. Given these pairs, we generate a partial monotone alignment between the two sentences where only lexically identical words are aligned. An alignment specifies portions of the English text that are translation of portions of the foreign text, and is commonly used in training statistical translation systems. A partial alignment only specifies alignments for some portions of the text and a monotone alignment does not allow crossing alignments. Using this alignment, we extract the unaligned phrase pairs as candidate phrase-level correction rules. These candidate rules are scored and filtered to generate the final context-independent phrase-level correction rules. The learned correction rules are applied to unseen translated examples to improve the translation performance.

This type of approach is most useful on commercial systems which are often rule based and general domain and therefore receive the most benefit from data-driven correction. Even statistical translation systems trained on the same parallel data benefit. Translation from foreign to English introduces additional regularity that may not be accessible in the original foreign/English data. Applying these rule correction approaches to commercial systems retains their computational and robustness benefits, while moving the translation performance towards that of statistical systems.

Figure 1.2 shows corrected sentences using learned phrase-level rules. Using the

Monsieur l'Orateur, ma question est simple. Mr. Speaker, my question is simple. M. Roch La Salle (Joliette) propose: Mr. Roch La Salle (Joliette) moved: ...beauty is life when life unveils her holy face. ...beauty is life when life unveils her holy face.

Figure 1.3 Three foreign/English parallel bilingual examples from Canadian parliamentary proceedings. The first line is French and second English. The third example is an English quote and is therefore the same in French and English.

Systran translation system, we identify over 70,000 correction rules. When applied to a test corpus, these rules improve the BLEU score, the standard measure of translation performance, of the test data by 30%. Using the same rule learning procedure, we also statistically significantly improve the translation performance of a state of the art phrase-based statistical translation system.

1.2 Learning Example Usefulness

The methods we have discussed have only assumed that we can translate text using the translation system. Research systems are also available where, given bilingual examples of parallel sentences, a translation system is learned. These examples are time-consuming to generate and are only available in limited domains. An important question is whether all these examples are equally useful for training a translation system.

Figure 1.3 shows three different bilingual examples. All of these examples show inferior characteristics. The first example is a straightforward, literal translation. However, the example does not provide new information in the context of other training examples. In a sample of 50,000 examples, 4,285 contain the phrase 'Mr. Speaker', 279 contain the phrase 'my question is' and 93 contain the word 'simple'. The second and third examples suffer from a different lack of information. In both cases, the English

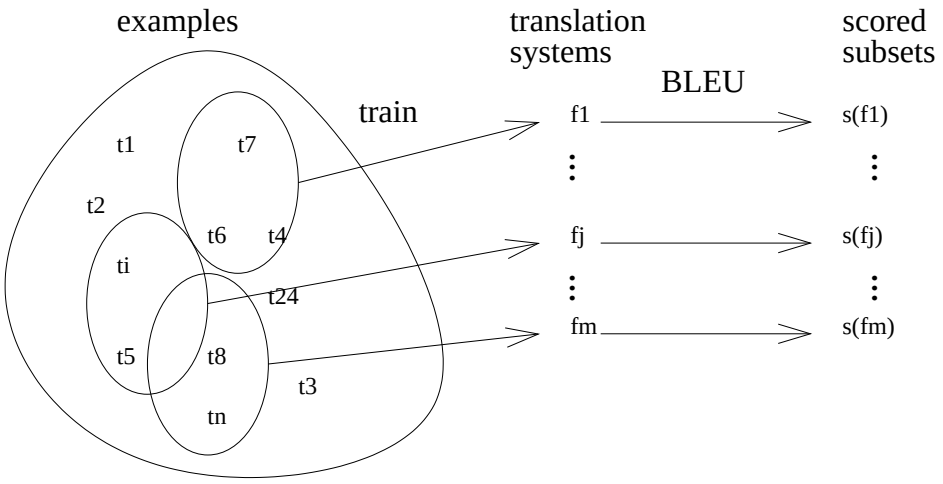


Figure 1.4 Process for generating scored example subsets to use as training data for ranking example usefulness. Random subsets of the examples are selected. Each of these subsets is used to train a translation system, which is then automatically evaluated.

and French are similar. In the second, this occurs because proper names are often the same in languages with similar character sets. The third example is an English quote and is therefore the same in both languages.

The examples in Figure 1.3 highlight just a few causes for example inferiority. In this thesis, we describe a framework that given a set of examples empirically ranks the examples based on usefulness. Because no prior example rankings exist, an important step is generating usefulness data for training. We cannot directly obtain individual example scores, but can obtain usefulness scores for sets of examples. Given a set of examples, random example subsets are selected and used to train translation systems. These systems are automatically evaluated resulting in scores for the random subsets. Figure 1.4 shows this process. These subset/score pairs define the training data for learning usefulness scores for the individual examples.

We assume that the score for an example subset is a linear combination of the example scores in that subset. For phrase-based systems, where each example is broken into a finite number of phrases, this assumption is reasonable. These random subset/score pairs then define a set of linear constraints. We calculate an example's score as the aver-

No Signal from ESA's Beagle 2 Probe Since Its Landing on Mars
Contacts Lost with Europe's Mars Probe Beagle 2 After Landing
No News from The European Mars Probe - Beagle 2 after It Landed
European Mars Probe Beagle 2 Remains Silent after Landing

Figure 1.5: Four human translations of the same Chinese headline.

age of the subset scores that example occurs in. This solution has the advantage of being efficient to calculate and theoretical analysis shows that the solution is near correct.

The examples are then ranked based on the learned usefulness scores. Given this ranking, we show that on a large test set a system trained on the highest-ranking examples consistently performs better than systems trained on a random selection of examples. An important application for the ranked examples is to build a model of example usefulness. This model can potentially be used to identify useful example characteristics and for selecting unlabeled examples that are most useful. As a first step towards building this model, we examine features previously suggested for translation confidence estimation for correlation with the learned ranking.

1.3 Improving Automatic Evaluation Methods

One of the challenges for many natural language applications is that there are many correct solutions to the same problem. This complicates training and evaluation. Figure 1.5 shows four different human translations of the same Chinese headline. Each of the translations conveys the same meaning: Europe sent a probe that landed on Mars, the probe is named Beagle 2 and since landing on Mars, the probe has been out of communication contact. However, the words used to convey this information are different between the different sentences.

Natural language applications such as machine translation and summarization rely on automatic evaluation measures that compare a method's output to a human-generated reference example. There have been a number of different proposed methods for mak-

Ref:	*Filipino* Communists Refuse Talks with Arroyo's Government and Launch New Guerrilla Attacks
Para:	**Philippine** Communists Refuse Talks with Arroyo's Government and Launch New Guerrilla Attacks
Ref:	The economy in our *country* continued to maintain a nice growth trend.
Para:	The economy in our **nation** continued to maintain a nice growth trend.
Ref:	As at end of 2003, Mongolia had 255.6 thousand camels, 1.9583 million horses, 1.7843 million *cows*, 10.7062 million sheep and 10.6034 million goats.
Para:	As at end of 2003, Mongolia had 255.6 thousand camels, 1.9583 million horses, 1.7843 million **cattle**, 10.7062 million sheep and 10.6034 million goats.

Figure 1.6 Paraphrases of human reference translations found by our proposed paraphrasing algorithm. The first line is the reference and second paraphrased reference. Paraphrases are shown in **bold**.

ing this comparison, but they all rely on an analysis of n-gram overlap between the machine-generated text and the human-generated text. As we see in the example, even among human translations, there are still many gaps in the overlap between sentences. A comparison of 10,728 human reference translation pairs used in the NIST 2004 machine translation evaluation reveals only 21 (less than 0.2%) that are identical and 60% that differ in at least 11 words.

Because of this flexibility, human references rarely capture the full range of correct solutions. The use of multiple references has been suggested as a solution to this problem, but generating this data is expensive and only results in a partial solution. We explore the use of paraphrasing to address this problem. Given a human-generated reference sentence and a machine-generated sentence, we seek to find a word-level paraphrase of the reference sentence that is closer in wording to the machine output than the original reference. For all words in the reference sentence that do not occur in the machine-generated sentence, candidate paraphrases are suggested using existing lexico-semantic resources, such as WordNet. These candidate word paraphrases represent context-independent choices and are not appropriate in all sentences. To address this, for each candidate paraphrase, we learn a model of the contexts that word occurs in.

The candidate paraphrase is only substituted if the likelihood of the word in the context of the reference sentence is acceptable according to the learned model.

Figure 1.6 shows reference sentences and the paraphrases of those sentences generated using our proposed method. All of the reference paraphrases increase the overlap with the machine translation output. We show experimentally that using the paraphrased references increases the evaluation performance of automatic evaluation methods. We also show a connection between paraphrase quality and automatic evaluation performance: better paraphrases result in better automatic evaluation performance.

1.4 Outline

Before discussing these methods in more detail, in Chapter 2 we first discuss background material and data and software resources. In Chapter 3 we use monolingual data to learn word-level correction rules to improve a translation system. We continue the translation improvement problem in Chapter 4 by learning phrase-level correction rules using parallel bilingual data. In Chapter 5 we switch to the problem of ranking bilingual examples based on usefulness. Finally, in Chapter 6 we analyze the effect of paraphrasing on automatic evaluation measures. We conclude with a summary of the key findings in this thesis and suggested future research directions.

2

Background

2.1 Translation Systems

Many different types of machine translation systems exist. In this section we briefly overview current translation systems available. Classifying these different systems is problematic, particularly for commercial systems[1]. We loosely divide the systems into rule-based systems and data-driven approaches.

2.1.1 Rule-based approaches

During the inception of machine translation, methods were linguistically driven, rule-based approaches. Today, most research methods are data-driven and only commercial translation systems still use rule-based approaches. Commercial systems can be purchased for home or office use and are publicly available through a number of web portals. These systems are efficient and general-purpose. Most commercial systems translate by combining translation dictionaries, idiomatic expressions, semantic dictionaries and homograph resolution with human generated rules. [27] discusses a number of commercial systems in detail and individual papers can also be found for some systems, for example [60].

[1]Traditionally, commercial systems were rule-based. Recently, there have been a number of commercial systems introduced that have been based on statistical translation approaches.

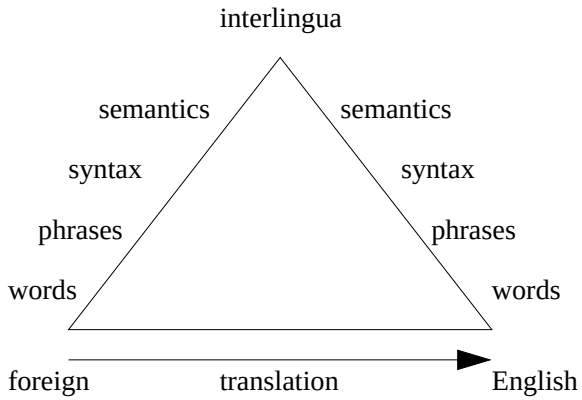


Figure 2.1 Translation method categorization pyramid. Translation occurs by transferring language information in the foreign language from the left to English language information on the right.

Rule-based approaches can be subdivided based on the type of information that is used when translating between languages. Figure 2.1 shows the archetypal translation method pyramid. At the very top are interlingua approaches. To translate from a foreign language to English, the foreign text is first translated into a language independent knowledge representation. From this knowledge representation, the English text is then generated [45]. This framework proved difficult and general-purpose interlingua methods have not emerged.

Moving down the pyramid are transfer methods, both semantic and syntactic [27]. Rather than translating the foreign text into a language independent representation, language specific semantic or syntactic knowledge is extracted from the text. Rules are then applied that convert this foreign representation into an English representation. From this English representation, the final English text is generated. As with interlingua approaches, general-purpose transfer approaches have not been successful.

2.1.2 Data-driven approaches

In the last 10 years, research systems have made substantial progress. This progress can be attributed to a shift from rule-driven approaches to data-driven approaches. These

data-driven approaches have been greatly assisted by increases in computational power and data availability. The two data-driven approaches that are currently popular are example-based methods and statistical methods.

Example-based systems Given a large bilingual data set of foreign sentences with associated English translations, example-based translation methods translate a new foreign text fragment by finding the fragment in the data set that is most similar to the new foreign sentence [11, 63]. Example-based machine translation methods differ in the size of the fragment, post-processing after matching and the matching criteria, which may incorporate syntactic information [28] or other linguistic information. [63] gives a good historical review of example-based translation methods.

Statistical systems Given the same bilingual data set, statistical machine translation methods take a different approach. A probabilistic model is learned, $p(e|f)$, that describes the process of translating a foreign sentence to English. Given a new foreign sentence f, translation occurs by finding the most likely English sentence, given that foreign sentence

$$\arg\max_e p(e|f)$$

The foundation of statistical machine translation research is the noisy channel model [9]. Rather than directly modeling $p(e|f)$ we apply Bayes rule:

$$p(e|f) = \frac{p(f|e)p(e)}{p(f)}$$

where $p(f|e)$ is the translation model, $p(e)$ is the language model of the English sentence and $p(f)$ is ignored since f is given and does not affect e. Each of these models are trained individually and then combined during translation. The translation model describes how a foreign sentence gets translated to an English sentence. The language model describes what English sentences look like. The combination of these two models results in a much more robust translation system than trying to model the entire translation process indivisibly.

In practice, an extension of the original noisy channel model, a parameterized log-linear model, is used:

$$\log p(e|f) = \lambda_1 \log(p(f|e)) + \lambda_2 \log(p(e)) + \lambda_3 g_1(f,e) + ... + \lambda_n g_n(f,e)$$

where g_i are additional feature functions such as the sentence length, number of words found in a bilingual dictionary, etc. and λ_i are model weighting parameters. These feature functions complicate the translation process (i.e. finding the most likely English sentence according to the model), but allow additional criteria to be included in the translation process.

For the language model, an n-gram model is most commonly used. The probability of a sentence is broken down into individual word probabilities [41]. We assume that the probability of a word occurring is only dependent on a small number of the previous words. For example, a bigram language model models the probability of a word in a sentence given only the previous word. Specifically, the probability of a sentence $p(e_1, e_2, ..., e_n) = p(e_1|\langle start \rangle)p(e_2|e_1)...p(e_n|e_{n-1})p(\langle end \rangle|e_n)$. These individual probabilities are estimated from a corpus. For those translation models that include syntactic information, a syntax-based language model is used in addition to the n-gram model [12]. These syntax-based models determine the probability of an entire syntax tree by similarly decomposing the tree into smaller components.

The main differentiating factor between statistical translation approaches is the translation model. The first models were word-level translation models [9]. Each word in the foreign sentence is translated to zero or more English words with word reordering. To obtain reasonable performance, these word level models tend to be complicated. The problem is that translation is rarely word for word. To model this, some words are translated to multiple different words, some words disappear during translation and some words randomly appear during translation. All these scenarios complicate the translation model.

The best performing models are phrase-based models. Phrase-based models take a similar approach to word models, but the translation component is the phrase. A foreign

sentence is broken into phrases and each of those phrases is translated to an English phrase with reordering [50, 33, 42, 65]. Because the translation is phrase-level, many of the complications seen in the word-level models do not arise.

Recently, models incorporating syntactic information have been suggested. [67, 68, 24] suggest probabilistic models that transform a syntactic tree in the foreign language into a syntactic tree in English. A number of "tree-to-string" models have also been proposed that translate a foreign sentence to an English syntactic tree [69, 24, 22]. Models that do not depend on explicit syntactic structure, but involve hierarchical structure have also received some interest [13].

2.1.3 Machine translation evaluation

Evaluating the performance of natural language methods is difficult. Traditionally, machine translation methods were evaluated by human judges. Though human judging still occurs, for most uses, it is too slow and expensive. Instead, automatic evaluation measures are used that judge the performance of a machine-generated translation with respect to a human-generated reference translation [46, 43, 53].

Throughout this thesis, we use the BLEU evaluation method [53]. BLEU is the geometric average of the n-gram precisions of the machine-translated sentences with respect to the corresponding human-translated reference sentences, times a brevity penalty. The BLEU score is computed as

$$\begin{aligned} BLEU &= B \cdot \sqrt[4]{\prod_{n=1}^{4} p_n} \\ B &= \min(1, e^{1-r/c}), \end{aligned}$$

where p_n is the n-gram precision, c is the number of words in the machine-generated text and r is the number of words in the shortest reference text. The n-gram precision is the proportion of n-grams in the machine-translated sentence that are found in the reference sentence. These precisions are calculated over the entire test set. The brevity

penalty B penalizes translations that are too short by discounting those translations that are shorter than the shortest reference translation. This prevents "gaming" of the metric, for example just outputting "the" which tends to have a high precision.

t-test An important question for any empirical study that compares two different methods is whether the difference between the performance of the methods is significantly different. In this thesis, we often use the paired t-test to determine to what extent this is true. The t-test asks whether two paired sets are significantly different *assuming* individual scores are independent and normally distributed.

Let X and Y be sets of data points where each point in X corresponds to a unique point in Y. In the case of algorithmic differentiation, these points represent two different methods trained on the same data and scored using an evaluation metric. The t-score for the difference between these two sets is

$$t = (\overline{X} - \overline{Y}) \sqrt{\frac{n(n-1)}{\sum_{i=1}^{n}((\overline{X} - X_i) - (\overline{Y} - Y_i))^2}}$$

where $\overline{X}$ and $\overline{Y}$ are the means of the sets and n is the number of pairs. Given the t-score, the probability that the two sets are significantly different can be looked up in a t-table under $(n-1)$ degrees of freedom.

The t-test assumes that the points are independent. As [16] point out, this is rarely the case for when using the t-test to compare different algorithms. [16] suggests a number of alternative tests; however, for machine translation, the training/testing setup is different than most supervised scenarios and the alternate tests suggested are not appropriate. In practice, non-independence leads to fewer degrees of freedom.

2.1.4 Corpora

We use a number of different corpora in this thesis for training and evaluation. All of these corpora are publicly available, though some require membership to the Linguistic

Data Consortium[2] (LDC) or to have participated in a particular event.

Hansard corpus The Hansard corpus is a Canadian French/English parallel bilingual corpus. The corpus contains 2.87 million aligned sentence pairs consisting of 70 million words transcribed from the Canadian parliamentary proceedings. Although the corpus only contains parliamentary proceedings, a wide variety of topics are discussed. The corpus was obtained through the LDC.

PAHO corpus The Pan American Health Organization (PAHO) Conferences and General Services Division parallel texts consists of 616 thousand words divided into 180 pairs of documents in English and Spanish [51]. The 180 documents were automatically segmented into 20 thousand sentences, identified by periods (minus a number of abbreviations). The sentences were NOT aligned. This corpus is available online[3].

Europarl corpus The European Parliament Proceedings Parallel Corpus contains 28 million words of paragraph-aligned transcriptions in 11 European languages (French, Italian, Spanish, Portuguese, English, Dutch, German, Danish, Swedish, Greek and Finnish) [31]. The text was automatically extracted from the parliamentary proceedings from 1996-2003. This corpus is available online[4].

NIST 2004 The National Institute of Standards and Technology (NIST) performs yearly evaluation of submitted machine translation systems. These systems include commercial and research systems. We use the Chinese portion of the 2004 data set which consists of 200 Chinese documents subdivided into 1788 segments. Each segment is translated to English by 10 machine translation systems and by four human translators. A quarter of the machine translated segments are scored by human evaluators on a one to five scale along two dimensions: adequacy and fluency. Adequacy

[2] http://www.ldc.upenn.edu
[3] http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat
[4] http://people.csail.mit.edu/koehn/publications/europarl/

measures how well the content is preserved while fluency measures the quality of the English. The corpus is available to participants of the 2004 NIST evaluations.

North American News Text Corpus The North American News Text Corpus contains English news articles on a wide range of topics from the Los Angeles Times, Washington Post, New York Times, Reuters and Wall Street Journal published from 1994-1997 and contains 350 million words of text. The corpus was obtained through the LDC.

2.1.5 Software resources

Much of the work done in this thesis relies on previously developed machine translation systems. These systems are composed of a combination of different software components. All of the software used in this thesis is publicly available.

Pharaoh For the statistical phrase-based translation system we use the Pharaoh training algorithm[5] and decoder [32]. This is a state of the art system that performs competitively in the yearly NIST evaluations. Included in this package is an implementation of maximum BLEU discriminative training for learning model parameters [48].

GIZA++ GIZA++ is a method for learning word-level statistical translation models [49]. Given a parallel corpus, the procedure learns a probabilistic alignment between the words in the aligned sentences using the EM algorithm [9]. The most likely word alignment is then used as input to the phrase-based translation system.

SRILM toolkit For the statistical systems, a language model is also required. We used the SRI language modeling toolkit. The toolkit is an n-gram language model package with many different smoothing techniques implemented[6].

[5] http://www.iccs.informatics.ed.ac.uk/~pkoehn
[6] http://www.speech.sri.com/projects/srilm/

Carmel Carmel is a finite state transducer program[7]. Given a trained translation model from the phrase-based system, Carmel is used to generate the best n translations of a foreign sentence under that model. This n-best list of translations is then used for optimizing model parameters [48].

BoosTexter BoosTexter is a classification program that learns a boosted set of decision stumps [59]. BoosTexter is particularly well suited for many natural language classification tasks since it is very efficient and can handle large data sets with many features.

Commercial translation systems We used two different commercial translation systems in our experiments. In Chapter 3 we use translations provided online from SDL International[8]. These translations were obtained in February, 2003. In Chapter 4, we examine Systran Professional version 4.0.0.

[7] http://www.isi.edu/licensed-sw/carmel/
[8] http://www.freetranslation.com

3

Learning Word-Level Correction Rules

Most machine translation research requires bilingual data consisting of parallel sentences. The produced systems have good translation performance, but rely on an expensive data source. Parallel bilingual data is time-consuming to generate and is only available in limited domains. In this chapter, we propose a method for improving existing translation systems using monolingual data, which is abundantly available in many domains.

3.1 Introduction

Machine translation systems are often available in both directions of a language pair. In commercial settings, these systems are developed semi-independently and the dictionaries used by each are different. This results in a difference in the knowledge built into each system. By analyzing translations made by the systems in both directions, these differences can be identified and used to learn correction rules.

To improve a foreign-to-English translation system, we start with an English word list. We translate the words in the list to the foreign language and back again to English. The original English word list defines a ground truth for the double-translated list of words. Deviations from this ground truth point to cases where the system can be improved. From these translated lists, we generate a list of foreign words and possible

English translation options. For those foreign words where only one translation option exists, context-independent rules are learned. For foreign words with multiple possible translations, a corpus is used to identify words that significantly co-occur with each translation option using a likelihood ratio test. These significant words are used to define context-dependent rules that disambiguate between the possible English translation options.

There are many advantages to learning word correction rules over other types of approaches. Rule learning approaches have proved successful in other natural language problems and produce efficient and understandable rules [8]. Although the rules only change the translation output one word at a time, word-level translation occurs in most commercial systems and word-based research systems are competitive. Also, [34] show that 90% of the words in a corpus can be translated using word for word translation.

In Section 3.2 we discuss the rule learning algorithm including the rule format, the data generation process and the method for dealing with ambiguous translations. To evaluate the performance of the learned rules, in Section 3.3 we apply the learned rules to a commercial system translation of a Spanish corpus and examine the number of changes made as well as the precision of these changes. We conclude in Section 3.4 with future work.

3.2 Algorithm

A machine translation system translates from one natural language to a second. We define f to be a translation system that translates from English text to foreign text and f' a system that translates in the reverse direction. We assume that we have unlimited access to the translation systems, but not to the details of how the systems operate. We also assume that we have a large amount of monolingual text available in the languages that the machine translation systems translate between.

The input to our algorithm consists of the two translation systems (foreign-to-English

and English-to-foreign) and text resources: an English word list, a foreign word list[1] and an English corpus. The output is word-level correction rules that improve the foreign-to-English translation system. Figure 3.2 outlines the steps for learning these correction rules. In this section, we look at each of the steps in more detail.

3.2.1 Rule format

We learn word-level rules of the form:

$$g(s, t, context(r)) = r$$

where s is a foreign input word, t is a system translated English word and $context(r)$ is an English context. We use a bag of words representation for this context. Although this loses positional information, it is a simple representation that reduces the number of parameters required during learning.

A context-independent rule is one that does not contain a context (denoted []) and applies anytime s occurs in the foreign sentence and is translated as t. The application of the rule changes t to r. A context-dependent rule adds the restriction that the translated sentence must also contain one of the words in the learned context of the rule to apply. Figure 3.1 shows an example application of the context-independent rule:

$$g(\text{alquitrán, alquitrán, []}) \rightarrow \text{tar}$$

which changes “alquitrán” to “tar”.

The context-dependent rules have the possibility of including both an input context and an output context. In practice, only context in the input or output language is necessary. In our case, for foreign-to-English improvement, English text is more readily available, so only the output contexts are learned. These English contexts are different than the actual contexts used when applying the rules, which consist of foreign words

[1] If unavailable, the foreign word list is not required for our method. The addition of a foreign word lists allows us to learn additional correction rules, but a majority of the rules are learned using only English resources.

foreign:
El contenido de **alquitrán** en los cigarrillos de tabaco negro sin filtro es mayor que en los restantes tipos de cigarrillos y son aquellos precisamente los de mayor consumo en la población, lo que aumenta la potencialidad del tabaquismo como factor de riesgo.

Original translation:
The content of **alquitrán** in the black cigarettes of tobacco without filter is greater that in the remaining types of cigarettes and are those precise the of greater consumption in the population, what enlarges the potencialidad of the tabaquismo as factor of risk.

Improved translation:
The content of **tar** in the black cigarettes of tobacco without filter is greater that in the remaining types of cigarettes and are those precise the of greater consumption in the population, what enlarges the potencialidad of the tabaquismo as factor of risk.

Figure 3.1 Example application of the rule $g(\text{alquitrán, alquitrán, []}) \rightarrow \text{tar}$. The first sentence is the original Foreign sentence to be translated (in this case, Spanish). The second sentence is the translation made by the commercial system. The final sentence is the translation after the rule is applied. The changed word is in **bold**.

translated to English by the system. In our experiments, the benefit of a large English data set outweighs the downsides of using translated English contexts. Also, the bag of words representation minimizes problems due to irregular contexts by not considering higher order n-grams or positional information.

3.2.2 Generating training data

Monolingual resources, such as a word list, do not contain translation information. We use the word list to extract translation information from the English-to-foreign and foreign-to-English translation systems. Given translation systems f and f' and an English word list, we calculate translations $f(w)$ and $f'(f(w))$ for every word w in the list. By translating words, rather than larger text fragments, we avoid the alignment problem of determining which translated words are associated. From these translations, we obtain a list of foreign words and possible English translation options, which is then used for rule learning.

We examine the SDL International translation system that translates in both directions between English and Spanish. Table 3.1 shows a summary of the data generated

Algorithm Input/Output

Input:

English word list

foreign word list (optional)

English corpus

Output:

context-independent and context-dependent correction rules to improve the foreign-to-English translation system

Generate training data

- Translate English word list to the foreign language and back to English
- Translate foreign word list to English and back to the foreign language
- Generate *input word* s (foreign), *current translation* t and *correct translation* r (English) triplets using the rules in Table 3.2
- For all words w in the corpus, generate frequency counts, $count(w)$
- Let $option_1, option_2, ..., option_n$ be all possible English *correct translations* for a given *input word*

Learn context-independent rules for non-ambiguous words

- Identify non-ambiguous words by finding all *input words* with only a single translation option (i.e. $n = 1$)
- Generate context-independent rules of the form:

$g(s, t, []) \rightarrow r$

Learn context-independent rules for k-dominant words

- Identify all k-dominant *input words* where

$count(option_p) > k$ and $count(option_q) = 0$ for all $p \neq q$

- Generate context-independent rules of the form:

$g(s, t, []) \rightarrow option_p$

Learn context-dependent rules for ambiguous words

- Get the possible context words w for each $option_p$ for the remaining ambiguous *input words*:
  - In the English corpus, find sentences where $option_p$ appears
  - Get all possible context words w as the words surrounding $option_p$
- For each $option_p$, generate the context, $context(option_p)$, as all w that pass the significance level α threshold for the likelihood ratio test
- Learn context-dependent rules of the form:

$g(s, t, context(option_p)) \rightarrow option_p$

Figure 3.2 Outline of algorithm to learn rules to improve foreign-to-English translation. The preprocessing steps generate the initial data for use in learning the rules. The following three sets of steps describe the algorithms for learning the context-independent and context-dependent rules.

Table 3.1 Exhaustive list of the different possible translation equalities/inequalities given a word (w), the foreign translation of that word ($f(w)$) and the translation back to English ($f'(f(w))$).

	Occurrences	Example w, $f(w)$, $f'(f(w))$
$w = f'(f(w)) \neq f(w)$	9,330	dog, perro, dog
$w = f(w) \neq f'(f(w))$	278	metro, metro, meter
$w \neq f(w) = f'(f(w))$	8,785	scroll, rally, rollo
$w = f(w) = f'(f(w))$	11,586	abstractness, abstractness, abstractness
$w \neq f(w) \neq f'(f(w)) \neq w$	14,523	cupful, taza, cup

from translations in February 2003 of 45,192 English words [56]. A partition (i.e. non-overlapping, exhaustive set) of the possible outcomes is shown. In this section, we examine each of these cases and describe the information generated for improving the translation systems. For most machine translation systems, the default behavior when the translation for a word w is unknown is to translate the word as w (i.e. $f(w) = w$). We assume that equality implies that the system could not translate the word. A message or flag issued by the system could be used instead, if available.

- $w = f'(f(w)) \neq f(w)$

  w is translated to a different string, $f(w)$, in the foreign language and then $f(w)$ is translated back to the original word w. In this situation the machine translation system is likely translating these words correctly. Mistakes can still occur here if there are complementary mistakes in the translation system lexicons. In either case, we do not learn any information.

- $w = f(w) \neq f'(f(w))$

  w is translated to the same string in the foreign language. However, it is then translated to a different string when it is translated back to the original language. This happens when w is a word in both languages (possibly with different meanings) and the English-to-foreign system (f) does not know the correct translation (for example, w = arena, $f(w)$ = arena, $f'(f(w))$ = sand). From these examples, we learn that the translation system f should translate $f'(f(w))$ to $f(w)$. This

information may or may not be useful. We can query f to see if this information is already known.

- $w \neq f(w) = f'(f(w))$
  w is translated to a different word in the foreign language, but it is then translated as the same word when translated back to English. There are two cases where this happens.

  1. The most likely situation is that there is a problem with the foreign-to-English system (f'). In this case, two pieces of information are learned. First, if $f(w)$ is seen on the input and is translated to $f'(f(w))$ then a mistake has occurred. We can also suggest the correct translation. Given a sentence $\overline{s}$, if word s is translated to s and $s = f'(f(w))$, then s was incorrectly translated and the correct translation is w.
  2. The second case, which is less likely, is that $f(w)$ is a word that, when translated back to English, is the same string (this is similar to case 2 below of $w = f(w) = f'(f(w))$). For example, w = abase, $f(w)$ = degrade (present subjunctive form of degradar, to degrade), $f'(f(w))$ = degrade. We learn that $f(w)$ is an ambiguous word that can be translated as either w or $f'(f(w))$.

- $w = f(w) = f'(f(w))$
  In this case, all the words are the same. There are two situations where this can occur:

  1. If the translation for w in the second language is w then the translation is correct. This is common with proper names (for example, w = Madrid, $f(w)$ = Madrid, $f'(f(w))$ = Madrid). In this case, no information is learned.
  2. If the English-to-foreign system (f) is unable to translate w, then $w = f(w)$. If this is the case, it is unlikely that w will actually be a valid word in the foreign language (as shown above, this does happen 278 out of 45,192 times, where the $f(w)$ is translated to something different by f'). Since w is not a

valid foreign word, it is again translated as w from foreign-to-English (for example, w = matriarchal, $f(w)$ = matriarchal, $f'(f(w))$ = matriarchal). In this case, the translation system f makes a mistake on word w.

- $w \neq f(w) \neq f'(f(w)) \neq w$
  There are two situations that can cause this to happen: w is a synonym for $f'(f(w))$ or there is at least one error in the translation systems. If we assume that the knowledge in the translation systems is accurate, then both w and $f'(f(w))$ are appropriate translations for $f(w)$. These two cases can be disambiguated using contextual information.

One last piece of information can be obtained when $f(w) \neq f'(f(w))$. In these cases, some translation was done by f'. We assume that $f'(f(w))$ is a word in the original language and can extend the word list in the original language.

Table 3.2 summarizes the information that is used to generate the training data from these translated word lists. The *input words* are foreign words. The *current translations* are the words expected to be seen in the output of the foreign-to-English translation system being improved. Finally, the *correct translations* indicate which word the *current translations* should be changed to.

3.2.3 Rule learning

Using the method described in the previous section, we obtain a list of foreign words and possible English translations for each foreign word. In this section, we describe how to use this data to learn correction rules to improve the foreign-to-English translation system.

Non-ambiguous words

For non-ambiguous foreign words where only one translation option exists, a context-independent rule of the form $g(s, t, []) = r$ is learned, where s = foreign input word, t = current English translation and r = correct translation. Almost all non-ambiguous words

Table 3.2 Patterns for generating training data for learning rules to improve the foreign-to-English system. For each different translation list equality option, we learn that when "input word" is translated as "current translation" then it should be translated as one of the "correct translations". Most of the information for improving the foreign-to-English translation system is learned from the English words list (English/foreign/English examples), but some information can also be learned from a foreign word list if available (foreign/English/foreign examples).

Case	input word	current translation	correct translation
English foreign English $w \neq f(w) = f'(f(w))$ $f(w)$ is **not** an English word	$f(w)$	$f'(f(w))$	w
English foreign English $w \neq f(w) = f'(f(w))$ $f(w)$ **is** an English word	$f(w)$ $f(w)$	$f'(f(w))$ $f'(f(w))$	w $f'(f(w))$
English foreign English $w \neq f(w) \neq f'(f(w))$	$f(w)$ $f(w)$	$f'(f(w))$ $f'(f(w))$	w $f'(f(w))$
foreign English foreign $w = f(w) \neq f'(f(w))$ $f'(f(w)) = f(f'(f(w)))$	$f'(f(w))$	$f(f'(f(w)))$	$f(w)$
foreign English foreign $w = f(w) \neq f'(f(w))$ $f'(f(w)) \neq f(f'(f(w))) \neq f(w)$	$f'(f(w))$ $f'(f(w))$	$f(f'(f(w)))$ $f(f'(f(w)))$	$f(w)$ $f(f'(f(w)))$

are generated from the case when $w \neq f(w) = f'(f(w))$. In this case the English-to-foreign system knows that w should be translated as $f(w)$, but this knowledge is absent from the foreign-to-English system.

Ambiguous words

Many of the entries in Table 3.2 are inherently ambiguous, such as when $w \neq f(w) \neq f'(f(w))$. For the remaining ambiguous foreign words where at least two correct translations for the same foreign word exist, we must decide between the possible translation options. We suggest two methods that both leverage an English corpus to distinguish between translation options.

k-dominant words We would like to identify as many non-ambiguous words in the data as possible, since these result in simpler rules. The frequencies in a corpus of the translation options gives some indication about the likelihood of the options. We define a foreign word as k-dominant if one translation option occurs at least k times in the corpus and all other options do not appear at all. When a word is k-dominant, it is reasonable to assume that the input word should always be translated as the dominant option. For all k-dominant options, we learn a context-independent rule that always changes the current translation to the dominant translation option.

Determining significant context words For the remaining foreign words where multiple translation options exists and no one option is k-dominant, context is used to disambiguate between the options. Given an ambiguous input word with $option_1, ..., option_n$ as possible correct translations, the goal is to learn a context for each translation option that disambiguates it from the other options. For each translation option, we collect all words w from the English corpus that occur in the same sentences as the option. We then determine which of these context words significantly co-occur with the option. These significant words then become contextual triggers for discriminating between the different translation options for the context-dependent rules: one of the words in the learned context must occur in the sentence to be corrected for the context-dependent rule to fire.

Given a translation option $option_i$ and a context word w we want to determine if w occurs significantly in the same sentences as the option. Many methods have been proposed for discovering co-occurrences such as frequency counts, mean and variance tests, t-test, χ^2 test and likelihood ratio test [41]. The likelihood ratio test has been suggested as the most appropriate for this problem since it does not assume a normal distribution like the t-test nor does it make assumptions about minimum frequency counts like the χ^2 test [17].

Let S_i be the set of sentences that contain the translation option $option_i$ and $\overline{S_i}$ be the set of sentences that do not contain the translation option. For each context word w in the sentences belonging to S_i, we use the likelihood ratio test to determine whether or not that word significantly occurs in S_i. If w passes this test, then it significantly co-occurs with the translation option and is added to the context-dependent rule for that option.

The likelihood ratio test tests an alternate hypothesis against a null hypothesis. The null hypothesis is that the occurrence counts of w in the two sets (sentences with and without $option_i$) come from the same distribution. The alternate hypothesis is that the occurrence counts are different in the two sets. We also impose the further constraint that w must be more likely to occur in sentences of S_i, since we are interested in positively correlated co-occurrences.

For each set, the occurrence of w is modeled using the binomial distribution. For both hypotheses, the likelihood equation is $l = p(S_i; \theta_i)p(\overline{S_i}; \theta_2)$. For the null hypothesis, the assumption is that the counts come from the same distribution and $\theta_1 = \theta_2 = \theta$. The maximum likelihood estimates of the parameters are used

$$\theta = \frac{count(w, S_i \cup \overline{S_i})}{|S_i| + |\overline{S_i}|} \quad \theta_1 = \frac{count(w, S_i)}{|S_i|} \quad \theta_2 = \frac{count(w, \overline{S_i})}{|\overline{S_i}|}$$

where $count(w, S)$ is the number of times w occurs in S (for space, below we abbreviate this as $C(w, S)$). Using these parameter estimations and the binomial assumption, the

likelihood ratio can be calculated in a similar fashion to [17]:

$$\begin{aligned}
\lambda &= \frac{p(S_i;\theta_1)p(\overline{S_i};\theta_2)}{p(S_i;\theta)p(\overline{S_i};\theta)} \\
&= \frac{\binom{|S_i|}{C(w,S_i)}\theta_1^{C(w,S_i)}(1-\theta_1)^{(|S_i|-C(w,S_i))}\binom{|\overline{S_i}|}{C(w,\overline{S_i})}\theta_2^{C(w,\overline{S_i})}(1-\theta_2)^{(|\overline{S_i}|-C(w,\overline{S_i}))}}{\binom{|S_i|}{C(w,S_i)}\theta^{C(w,S_i)}(1-\theta)^{(|S_i|-C(w,S_i))}\binom{|\overline{S_i}|}{C(w,\overline{S_i})}\theta^{C(w,\overline{S_i})}(1-\theta)^{(|\overline{S_i}|-C(w,\overline{S_i}))}} \\
&= \frac{\theta_1^{C(w,S_i)}(1-\theta_1)^{(|S_i|-C(w,S_i))}\theta_2^{C(w,\overline{S_i})}(1-\theta_2)^{(|\overline{S_i}|-C(w,\overline{S_i}))}}{\theta^{C(w,S_i)}(1-\theta)^{(|S_i|-C(w,S_i))}\theta^{C(w,\overline{S_i})}(1-\theta)^{(|\overline{S_i}|-C(w,\overline{S_i}))}} \\
&= \frac{\theta_1^{C(w,S_i)}(1-\theta_1)^{(|S_i|-C(w,S_i))}\theta_2^{C(w,\overline{S_i})}(1-\theta_2)^{(|\overline{S_i}|-C(w,\overline{S_i}))}}{\theta^{C(w,S_i\cup\overline{S_i})}(1-\theta)^{(|S_i|+|\overline{S_i}|-C(w,S_i\cup\overline{S_i}))}}
\end{aligned}$$

In practice, $-2\log\lambda$ is used since it follows the χ^2 distribution asymptotically:

$$\begin{aligned}
-2\log\lambda = {} & C(w,S_i)\log\theta_1 + (|S_i| - C(w,S_i))\log(1-\theta_1) \\
& + C(w,\overline{(}S_i))\log\theta_2 + (|\overline{S_i}| - C(w,\overline{S_i})\log(1-\theta_2) \\
& - C(w,S_i+\overline{S_i})\log\theta - (|S_i| + |\overline{S_i}| - C(w,S_i+\overline{S_i})\log(1-\theta)
\end{aligned}$$

We compare this value with a significance level, α, to make a decision about the significance of the co-occurrence. We do this for all words in sentences that contain the translation option. A separate context-dependent rule is generated for each translation option. All words that pass the significance test are added to the set of words that define the rule context, $context(option_i)$. For our experiments, a significance level α = 0.001 is used. Intuitively, there is a one in a thousand chance of a candidate word being misclassified as significant.

To improve the generality of the contexts learned, we perform the test on stemmed versions of the words and generate context-dependent rules using these stemmed words. The Porter stemmer [54] is used to stem the words. For the remainder of the chapter, the results provided are for the stemmed versions.

3.3 Experiments

To evaluate the performance of our rule learning method, we examined a commercial Spanish-to-English translation system. We learned rules using a 45,192 English word list [56], a 29,977 word Spanish word list and 5.1 million English sentences from the North American News Text Corpus. We learn context-independent rules for non-ambiguous words and k-dominant words with k = 5. For ambiguous words, we learn context-dependent rules with $\alpha = 0.001$ used to test the significance of the context words.

We applied the learned rules to the Spanish portion of the Pan American Health Organization (PAHO) Conferences and General services parallel corpus. We evaluated the impact of the rules using two measures. To evaluate the coverage of the rules, we measure the number of rule changes made over the corpus. To evaluate the accuracy of the rules, 600 random rule firings were manually evaluated by a native English speaking judge. The judge was asked to choose between the original translation and the corrected translation for each rule firing.

3.3.1 Applying word correction rules

Each Spanish sentence in the PAHO corpus is translated using the commercial system to get the initial translation. Then, the rules learned using the algorithm in Section 3.2 are applied to correct the sentences. Table 3.3 shows the number of rules learned using our method. We learned 6,783 context-independent rules where only a single translation option existed and an additional 809 context-independent rules for k-dominant words. To correct the ambiguous foreign words, we learned 1,355 context-dependent rules. Table 3.3 also shows the results from applying these rules to the translations of the Spanish sentences. The rules change 22,206 words in 14,952 or 74% of the sentences from the translated PAHO data set.

Table 3.3 Summary of results for word correction rules generated from a word list with 45,192 entries applied to the PAHO data set.

Rule type	Rules learned	Avg. # words in context	Rules used	Words changed
Context-independent	6,783	-	701	5,022
Context-independent, dominant k=5	809	-	191	4,768
Context-dependent, signif. = .001	1,355	5	301	12,416

Table 3.4 Proportion of correct rule changes for both context-independent and context-dependent rules as measured by a native English speaker on 600 random changes.

Rule type	Proportion correct
Context-independent	0.99
Context-dependent	0.79

3.3.2 Rule precision

A native English speaking judge manually evaluated a random sample of 600 rule changes. For each change, the judge was asked to determine if the original machine translation or the corrected translation is correct. These options were presented randomly to the judge to avoid any bias. Figure 3.4 shows the proportion of correct rule firings for the context-independent and context-dependent rules. The context-independent rules have very high precision. This occurs since a majority of the context-independent rules represent changes where the original system does not know any English translation. In addition, the context-independent rules are correcting those words that were found to be non-ambiguous by the rule learning system. For the context-dependent rules, the rules must choose between multiple translation options, resulting in lower precision.

3.3.3 Using extended word lists

The methods in this chapter use an English word list to generate training data. In this section, we present two methods for extending this word list. One of the advantages of the rule learning method described above is that it is robust to erroneous words in the

Table 3.5 Summary of results for word correction rules generated using a general word list with 45,192 entries plus 419 learned words and 5,215 domain specific words applied to the PAHO data set.

Rule type	Rules learned	Avg. # words in context	Rules used	Words changed
Context independent	7,155	-	903	6,526
Context independent, dominant k=5	816	-	200	5,038
Context dependent, signif. = .001	1,444	5	327	12,671

word list. If the system does not recognize a word in the word list then it will not get translated, as is the case where $w = f(w) = f'(f(w))$. No learning is done in this case, so erroneous words are filtered by translation system.

When translating w to $f(w)$ and back to the original language as $f'(f(w))$, if $f(w) \neq f'(f(w))$ then some translation was done between $f(w)$ and $f'(f(w))$. We assume that if the machine translation system translates $f(w)$ to $f'(f(w))$, then $f'(f(w))$ is an English word. Using this method, 419 additional words not in the original English word list are learned.

In many circumstances, translation systems are to be used in a specific domain (for example medicine, politics, public health, etc.). The PAHO data set contains documents in the public health domain. To improve the recall of the machine translation system we can incorporate more rules that contain terminology that is relevant to this domain. For the PAHO data set, the English translations of the Spanish documents are available. Using this English data we add an additional 5,215 new words to the original English word list.

Table 3.5 shows the rule results with the original 45,192 words plus the additional 419 learned words and the 5,215 domain specific words. The additional words add 468 new rules. Although these new rules only constitute a small fraction of the total rules (5%) they account for over 8% of the changes. In particular, the domain specific context-independent rules fire over four times more often than the rules learned from the generic word list. Because these additional rules are learned using domain specific words, they

are much more likely to apply for translating text in that particular domain. With the addition of these new rules, 78% of the sentences are corrected by the rules.

3.4 Discussion

In this chapter, we have examined a technique for improving a machine translation system using only monolingual text. Unlike parallel bilingual data, monolingual data is available in a wide range of domains [40]. Our method only requires access to the translation system and makes no assumptions about the type of translation system.

By translating English words to a foreign language and back to English, differences in information between the foreign-to-English and English-to-foreign translation systems are isolated. Using this information, correction rules are learned. Context-independent rules are learned where only a single translation option exists. When there is ambiguity about the correct translation, the likelihood ratio test is used to identify words that co-occur significantly with each translation option.

Using the learned rules, over 24,235 words are changed on a corpus of 616,000 translated words. On a random sample, 99% and the context-independent rule firings were correct. The context-independent rules achieved a lower percision of 79% even though a significance level of 0.001 was used. The main reason for this lower precision is that the likelihood ratio can suggest co-occurrences that are significant, but that are not useful for ambiguity resolution. This is attenuated when the counts are very small or when the ambiguous translation is common and the counts are high. In these cases, common words such as “is”, “that”, “it”, “have”, etc. can be identified as significant.

Another problem is the rule representation chosen. The context-dependent rules use a bag of words representation. This representation loses information such as word order and does not include additional information such as syntax or semantics, which can be useful for discriminating significant co-occurrences. For example, when deciding between “another” and “other” in the sentence fragment “Another important coordination type...”, the location of “type” and the fact that it is singular suggests “another” as the

correct translation, but our existing method cannot detect this.

One final problem is that stemming can cause undesired side effects in the contexts learned. As seen in the sentence fragment above, plurality is important, particularly when deciding between two translations that only differ by plurality. Unfortunately, stemming, in attempting to improve generality, removes the plurality of a word. The combination of these problems leads to a lower precision for the context-dependent rules. Future research should be directed towards employing alternate rule representations and alternate collocation techniques such as in [35].

Acknowledgements

The text of this chapter, in part, is a reprint of the material in:
D. Kauchak and C. Elkan. Learning rules to improve a machine translation system. In *Proceedings of ECML*, pages 205–216, 2003.

The dissertation author was the primary researcher and author and the co-author listed in this publication supervised the research.

4

Learning Phrase-Level Correction Rules

In the previous chapter, we described a method for improving a translation system using only monolingual data. Monolingual data is widely available, but does not contain any novel translation information. Parallel bilingual data contains sentence fragments in two languages and is commonly used to train statistical translation systems. Parallel data is more expensive to generate, but is available in many languages and domains. In this chapter, we continue to examine translation system improvement and describe a method for learning phrase-level correction rules using this bilingual data.

4.1 Introduction

Figure 4.1 shows a foreign sentence, and machine and human translations of that sentence. The machine translation is a literal, word for word translation, but the meaning is not well preserved. In this chapter, we learn phrase-level correction rules to correct the machine translation output. In the example, *sitting is opened* should be changed to *House met* and *hours* should be changed to *p.m.* Rather than training on the foreign/English pairs, we instead train on the machine translation/human translated pairs. Training on these English-only pairs allows us to apply methods that utilize lexical sim-

La séance est ouverte à 2 heures.
(a) The sitting is opened at 2 hours.
(b) The House met at 2 p.m.

Figure 4.1 Example French sentence with machine translation (a) and human translation (b).

ilarity by leveraging the translation capabilities of existing translation systems.

Given a translation system to be improved and a parallel bilingual corpus, training data is generated by translating the foreign sentences using the system. This results in a new corpus consisting of machine translated and human translated sentence pairs. Ignoring paraphrasing effects, differences in these sentences point to mistakes in the translation system.

To identify these mistakes, a partial alignment is learned between lexically identical portions of the machine translated and human translated sentences. Unaligned phrase pairs are extracted based on this alignment as candidate phrase-level correction rules. Statistics are gathered about these candidate rules in the corpus and used to filter the rules based on their frequency and accuracy. Given a new translated sentence, the learned rules change one phrase to another phrase. Although paraphrases, rather than corrections, can be identified by our proposed procedure, they are often filtered out by the filtering steps and those that do pass tend to be high-quality, context-independent paraphrases and do not alter the quality of the translation.

The example highlights two reasons for applying an improvement method to machine translation. First, general performance improvements can be learned by training on an additional bilingual corpus. This is particularly important for commercial systems that are often rule-based rather than data-driven, but is also true for statistical systems where improvements are also achieved. Second, training on bilingual data in a different domain than the original system was trained allows domain-specific knowledge, such as *House met*, to be incorporated into the end system.

Although the method we present is applicable to any machine translation system, the biggest benefits are seen on commercial systems, which are not trained on the same data

used for rule correction. In many settings, commercial systems produce lower quality translations, but they are not without their benefits. A commercial system can translate 10,000 sentences in 1 minute. The same sentences take 83 minutes to translate using a phrase-based system (Pharaoh, [32]). Also, commercial systems tend to be more robust and fail more gracefully than research systems.

In this chapter, we first discuss related work. In Section 4.3 we describe the rule learning algorithm, including the alignment procedure and rule generation. In Section 4.4 we compare the results of a commercial translation system and a phrase-based system improved with both the learned context independent rules and a baseline translation improvement method. Finally, we conclude with a summary of contributions and future work.

4.2 Related Work

Many research projects involve learning rules from text data. In Chapter 3 we use monolingual resources to learn word correction rules for translation systems ([30]). [5] use documents translated from the same source to learn paraphrases for use in other applications such as interpretation and generation of natural language. They iterate between learning phrase rules and context rules in a co-training framework. For generalization, the rules include parts of speech in addition to the lexical component. This rule generality was designed to cope with the wide variations encountered in the different human translations. In our work, the translations are much more similar and allow for alignment based techniques. More general-purpose rule learning frameworks have also been suggested [8].

Our work is related to many aspects of current statistical machine translation research. One of the foundational components of statistical translation systems is an alignment of parallel text. An alignment specifies pairs of words or phrases in the parallel sentences that are translations. These alignments are used to calculate word translation probabilities [9], phrase translation probabilities [50] or probabilistic syntax rules [22].

We learn a partial alignment model that is related to phrase alignment models [42]. Most of the alignment models rely on some form of EM training to learn the sentence alignments [42, 9]. Because we pose our problem as a same-language learning problem, we can use simpler techniques to learn the alignments.

Besides alignments, our approach has a similar construction to phrase-based translation models [50]. These approaches learn a probabilistic translation model with reordering. Ignoring reordering, the generative process of a phrase-based system can be seen as replacing foreign phrases with English phrases. In our work, we learn deterministic rules that replace English phrases.

Finally, there has been some work on improving translation systems to maximize translation performance. [48] and [62] optimize model parameters on a development set to maximize the BLEU score. These methods are only appropriate for statistical translation systems where multiple models are combined in a probabilistic or log-linear model. We are interested in improvement procedures that do not rely on particular translation system characteristics.

4.3 Learning Phrase Rules

The input to the algorithm is a set of aligned bilingual pairs (s_i, e_i) and a machine translation system f, to be improved. Using this translation system, we first generate a set of training examples by translating the foreign sentences. This results in English sentence pairs of the same foreign sentence, ($f(s_i)$, e_i), where one translation was generated by the translation system and the other by a human.

Differences between these pairs point to possible places where the translation system is not translating correctly. To identify these differences, we first learn a partial alignment between the machine-generated and human-generated English translations. From this alignment, we extract candidate context-independent phrase rules. These candidate rules are then scored and filtered to obtain the final set of of correction rules.

4.3.1 Rule format

We learn context-independent phrase-level correction rules of the form $\tilde{p} \rightarrow p$, where both $\tilde{p}$ and p are non-empty strings. Given a translated sentence, a rule fires if $\tilde{p}$ occurs in the sentence and replaces $\tilde{p}$ with p. For example, applying the rule $\tilde{p} \rightarrow p$ to the sentence $t_1, t_2, ..., \tilde{p}, ..., t_n$, results in the corrected sentence $t_1, t_2, ..., p, ..., t_n$. The rules are context-independent since every occurrence of $\tilde{p}$ is replaced, regardless of the context it occurs in. This context independence assumption simplifies the learning procedure and tends to work well in practice.

4.3.2 Learning an alignment

A key advantage of learning from sentence pairs that are in the same language is that lexical cues can be used to assist the alignment algorithm. If a word occurs in both sentences, it is very likely that those two occurrences should be aligned. We leverage this information to generate the alignment between the sentences. For each sentence pair in the machine translated/human translated English training corpus, we learn the best partial monotone alignment where only lexically equal words are aligned. A monotone alignment is an alignment where no two aligned pairs cross. Specifically, given the machine translated sentence $t_1, t_2, ..., t_n$ and the human translated sentence $e_1, e_2, ..., e_m$, if t_i is aligned to e_j, then all words t_k such that $k > i$ can only be aligned to e_l such that $l > j$ and all words t_k such that $k < i$ can only be aligned to e_l such that $l < j$. The best alignment is the alignment that aligns the most words between the two sentences.

For sentences of length n, the best monotone alignment can be calculated optimally using a dynamic programming method in $O(n^3)$ time. Figure 4.2 outlines the algorithm to calculate the cost of the best alignment. As with most dynamic programming methods, to calculate the actual alignment, backpointers must be kept and retraced once the best cost has been found [14]. Figure 4.3 shows the alignment produced by this algorithm for the sentences in Table 4.1.

$cost$[0][j] = 0
$cost$[i][0] = 0

for i = 1 to length(S)
  for j = 1 to length(E)
    $cost[i][j]$ = **max**$\{cost$[i-1][j], $1 + cost[i][k]\}$
                $\forall k : S_i = E_k$ and $k \leq j$

Figure 4.2 Algorithm for determining the cost of the best cost monotone alignment between machine translated sentence S and human translated sentence E.

```
The         House met at 2    p.m.
 |                      |  |       |
The sitting is opened at 2 hours .
```

Figure 4.3 Word alignment learned by our method between the human-translated sentence (on top) and machine-translated sentence. Aligned words are denoted by ' | '.

4.3.3 Generating rules

Once the alignment is learned, we extract phrase pairs occurring between aligned words as candidate correction rules. Given the sentence pairs $\{(t_1, t_2, ..., t_n), (e_1, e_2, ..., e_m)\}$, aligned words t_i with e_j and t_k with e_l and the unaligned phrase $t_{i+1}...t_{j-1}$, we extract the candidate correction rule $t_{i+1}...t_{j-1} \rightarrow e_{k+1}...e_{l-1}$. For example, given the above alignment, we extract the two candidate correction rules "House met" $\rightarrow$ "sitting is opened" and "p.m" $\rightarrow$ "hours".

After generating the candidate correction rules, the final rule set is obtained by applying two filtering criteria to eliminate inappropriate rules. First, the candidate rules are filtered based on the number of times the phrase pair occurred aligned in the training data, i.e. whether $count(\tilde{p} \rightarrow p) > t_c$, t_c is the count threshold. Filtering by count assures that the phrases are not spuriously aligned and that the accuracy of the other scoring criterion is meaningful. We also filter the rules based on an estimate of the

Table 4.1 Sample phrase rules extracted for correcting the commercial system trained on one million sentences. A rule $\tilde{p} \rightarrow p$ changes the phrase $\tilde{p}$ to p.

Rule	score	count
how much costs some the ACDI, → at what cost to CIDA (	1.0	98
author of thelowest tender obtain → low bidder awarded	1.0	22
offices secondary → sub-offices maintained	1.0	15
for la1re → the first	0.959	304
meeting begins again → House resumed	0.895	334
At 10 hours → Ten o'clock	0.867	13
With the order! → Order, please.	0.593	73
Dirty → Sales	0.286	2
, this evening, → tonight	0.094	8
financial → fiscal	0.092	586

correction accuracy. This estimate is

$$score(\tilde{p} \rightarrow p) = \frac{count(\tilde{p} \rightarrow p)}{count(\tilde{p})}$$

where $count(\tilde{p})$ is the number of times $\tilde{p}$ occurs in the translated sentences. Figure 4.1 shows sample learned rules along with the rule count and score.

4.4 Experiments

To evaluate the effectiveness of the phrase rule learning method we applied the algorithm to a modern commercial system and a state of the art phrase-based system ([32]). We measure improvement based on the BLEU evaluation metric and also examine rule accuracy, coverage and computation time.

4.4.1 Experimental setup

Data We used the Hansard Canadian French/English bilingual corpus in our experiments. The first 1 million sentences (20.1 million words) were used as training data for

the phrase correction rule learning. A 100 sentence development set was used during training for parameter selection and ten 10,000 sentence data sets were used for testing.

Translation systems We learn correction rules to improve a modern commercial translation system and a phrase-based translation system. For the phrase-based system, we used the Pharaoh training algorithm[1] and decoder [32] and trained on the *same* 1 million sentences used for rule learning. We selected model parameters using maximum BLEU discriminative training [48]. For both the commercial system and phrase-based system, we measured translation performance and computation time. For the statistical system, there has been some research investigating greedy methods that trade off translation performance for faster translation times [67, 23]. For our experiments, we are interested in higher translation quality and internal decoding parameters were selected appropriately.

Alternate improvement method Given pairs of system and human translated English sentences we learn a set of improvement rules. For comparison, we also trained a phrase-based "machine translated English to human translated English" translation system on this same data. The one million translated English/human English sentence pairs used for learning rules are used as training data. As above, we used the Pharaoh translation system with maximum BLEU discriminative training. To correct an unseen machine translated sentence, the learned statistical model is used to "translate" the sentence.

Evaluation metrics Our goal is to improve the translation output of a machine translation system. We use BLEU to measure to what extent this is accomplished. One of the main advantages of using commercial translation systems is they translate much faster than research systems. This is important in many data-driven and real-time applications. For all of the methods, we measured the translation time and the correction time.

To better understand the behavior of the rule learning method, we also measured the number of changes made, the fraction of those changes that are known correct and incorrect, and the number of rules used. The number of correct changes is calculated as

[1] http://www.iccs.informatics.ed.ac.uk/~pkoehn

the number of rule changes where the changed phrase is found in the human translation. Similarly, the number of incorrect changes is calculated as the number of rule changes where the original phrase is found in the human translation, and incorrectly changed by the rule. In practice, there are still many changes that do not fall into either of these categories for which we cannot determine the correctness based on the human translation.

Parameter estimation The last step in the phrase learning algorithm is to filter the rules by occurrence count and by score. By varying these thresholds the coverage and the accuracy of the rule set is affected. For our experiments, we are interested in maximizing the BLEU score. The BLEU score was calculated for all threshold values for occurrence count = $\{2, 3, 4, 5\}$ and for the score threshold = $\{0, .05, .1, ..., 1\}$ on an independent 100 sentence development set. The best performing pair of parameters was used during testing. For the commercial system, this was count ≥ 2 and score ≥ 0.1 and for the phrase-based system count ≥ 2 and score ≥ 0.25.

4.4.2 Improving MT systems

Translation performance Table 4.2 shows the average BLEU scores for the different methods. For both the commercial system and phrase-based system the learned phrase rules improve the translation performance. For the commercial system, more than 5 BLEU points improvement is achieved; a 30% improvement. Even on the phrase-based system, which is trained on the exact same data, the phrase rules result in a statistically significant improvement based on a paired t-test over the 10 test sets. For both translation systems, the phrase rules perform better than the more complicated and time consuming "translation" improvement process. On the phrase-based system, learning an improvement translation system fails to improve on the original system. Table 4.3 shows example corrections made to the commercial system.

The individual test scores for each of the 10 test sets are shown in Figure 4.4. For the commercial system, there is a clear separation between the original system, the

Table 4.2 Average BLEU score and computation time over ten tests for the two different translation systems improved with the two correction methods. "translation" is a trained English to human English statistical phrase-based translation improvement method. "phrase rules" is improvement based on the phrase rules generated by our method. All BLEU scores are significantly different based on a paired t-test over the ten tests, except "phrase none" vs. "phrase translation".

MT system	correction method	BLEU	time
commercial	none	0.170	57s
	translation	0.204	16,387s
	phrase rules	0.221	8s
phrase	none	0.250	5013s
	translation	0.250	16,524s
	phrase rules	0.252	6s

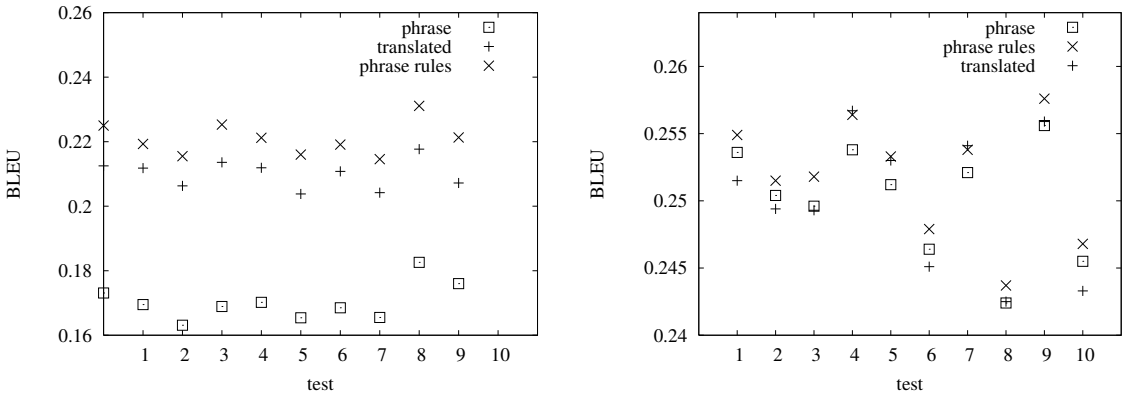
translation improved system and the phrase rule improved system. A statistical test of significance with a paired t-test over the 10 tests confirms these differences. For the phrase-based system, the separation is not as clear. Using the learned phrase rules, the improvement is consistent and significant. However, the translation improvement method only improves the BLEU score for 3 of the 10 tests and fails to perform better on average.

Correction time Table 4.2 also shows the computation times for the different algorithms. Besides resulting in better translations, the phrase rules are very efficient to apply. In both the commercial and the phrase-based system, applying the correction rules takes only a few seconds. Correcting the systems using the statistical translation method took substantially longer. In fact, using "translation" for correction tends to take longer than translating from foreign to English.

One of the reasons for using a commercial translation system is the computation advantage. The translation time of the commercial system is orders of magnitude faster than the phrase-based system. By applying the correction rules, with just a few additional seconds of computation time, the translation performance of the commercial sys-

Table 4.3 Example sentences corrected by the learned rules. Changed phrases are in *italics* and replaced by those in **bold**.

Human translated:	Mr. Speaker, I will be very brief.
Commercial:	*Mister the* Speaker, I will be very *short*.
Rule corrected:	**Mr.** Speaker, I will be very **brief**.
Human translated:	I really did not say that, even by implication.
Commercial:	I really didnot *make it clear* it either.
Rule corrected:	I really didnot **suggest** it either.
Human translated:	Again I say they cannot have it both ways.
Commercial:	I *repeat it*, it is one or the other.
Rule corrected:	I **say**, it is one or the other.



(a) (b)

Figure 4.4 BLEU scores for the commercial system (a) and the phrase-based system (b) for the 10 different test sets. For each of the systems the translation improved system and phrase rule improved system are also shown.

Table 4.4 The number of changes made, number of rules used and proportion of correct and incorrect rule changes all averaged over ten 10,000 sentence test sets.

System improved	known correct	known incorrect	number of changes	number of rules
commercial	0.482	0.076	14130	3410
phrase	0.635	0.067	194.1	85

tem is improved drastically, approaching the performance of the statistical phrase-based system.

Rule statistics To better understand the behavior of the phrase rule improvement method, Table 4.4 shows statistics for our method applied to both the commercial system and the phrase-based system. The rule correction method makes substantially more changes to the commercial system with many more rules than on the phrase-based system. With 14,130 changes over 10,000 sentences, the method averages more than one correction per sentence.

In both systems, the proportion of changes known to be correct is around 0.5 to 0.6. The BLEU score still improves, though, since the number of incorrect changes is low. For the commercial system, 48% are known correct, 7.6% are known incorrect and for the other 40% it is unknown whether they are correct or incorrect. These unknown changes occur when neither the original phrase nor the corrected phrase is found in the reference sentence. These unknown changes have only a minor impact on the BLEU score.

4.4.3 Rule filtering analysis

For the above experiments, we selected the threshold parameters to maximize the BLEU score on a development set. Depending on the desired application, these parameters can be set manually, for example to increase the rule accuracy. To analyze the relationship between the parameter settings and the improvement performance, Figure

4.5 show the BLEU score, correct proportion and number of changes for the commercial system on the first test set[2]. Similar trends were seen for correction rules learned for the phrase-based system.

For all of the different measures, there is only a small difference between the varied count threshold values. Increasing the rule count increases the rule accuracy and decreases the number of changes. The parameter that has the most dramatic effect on the results is the score threshold. In Figure 4.5(a) we see that the highest BLEU score for this test set is at 0.1, which is the value learned on the development set. Even though the accuracy of the rules is lower at this level, we see in Figure 4.5(c) that the actual number of changes is much higher for lower thresholds. As the threshold increases, the number of changes decays exponentially. This exponential factor offsets the increase in precision, resulting in lower BLEU scores.

For most of the range, as the threshold increases, the rule accuracy also increases. If high precision is desired, the score threshold can be adjusted. A higher threshold only changes the original translation system's translation when there is high confidence in the correction rule. For example, selecting a score threshold of 0.4 results in 80% of the rules making known correct choices, while still obtaining a 3 BLEU point increase (17% improvement).

4.5 Conclusions and Future Work

In this chapter we proposed a method for learning phrase-level correction rules to improve the translation performance of a translation system using a parallel bilingual corpus. Aligned machine translated/human translated English training data is generated by translating the foreign sentences to English. Using this data, we learn a partial monotone alignment. Based on this alignment, we learn correction rules. These learned rules improve the performance of a commercial system by 30% with only minimal extra computation time. Statistically significant improvements are also seen on a state of the

[2]For graph clarity, the score threshold of 0 is not shown, but does continue the trends seen in the current graphs.

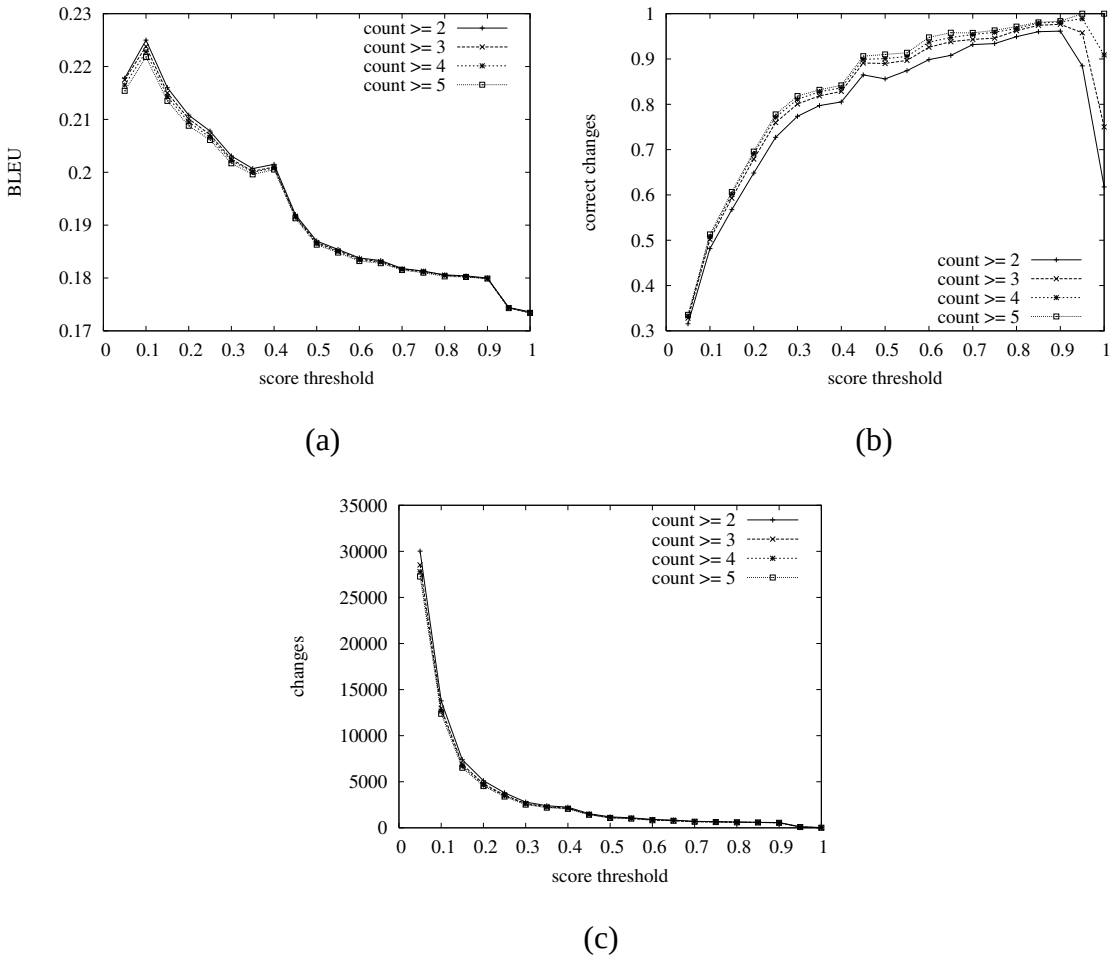


Figure 4.5 BLEU score (a), proportion of rule corrections that are correct changes (b), and the number of rule changes (c) on the first 10,000 test set for different rule filtering thresholds.

art phrase-based statistical system.

One possible improvement of the rule learning method is in the rule construction. We learn context-independent rules. While these are efficient to learn, the context independence assumption is often not true. For example, “hours” should not always be changed to “p.m.”. [30] propose contextual rules based on significantly co-occurring context words, which could be added to the learned phrase rules. Alternatively, rules incorporating syntactic information could be used [5]. Post-correction based on syntax is one method to retain the performance of a phrase-based system, while incorporating the theoretically well motivated syntactic information [22].

In this chapter, we have only looked at phrase-level correction rules. Examining translated and human English pairs can also be useful for other types of correction. One of the main criticisms with phrase-based translation models is the simplistic reordering model. Rather than attempt to modify this model internally, one solution is to post-correct the sentence ordering based on additional information, such as syntax. The advantage of this type of approach is that there are many more tools available when dealing exclusively with English, such as parsers and taggers.

5

Learning Example Usefulness

So far, we have examined two different methods for improving black box machine translation systems. The only assumption made was that we are allowed to query the translation of text using the machine translation system. In this chapter, we include the additional ability to train translation systems based on parallel bilingual examples. This additional assumption allows us to analyze the usefulness of individual examples for training a translation system.

5.1 Introduction

Natural language processing techniques are becoming more and more data-driven. Researchers rely on large data sets that often require many man hours of annotation at a considerable cost. Given this cost, an important question is whether all examples are equally useful.

In this chapter, we examine this question for machine translation, though the methods we discuss are applicable in many other applications. Statistical machine translation methods are trained on parallel bilingual examples consisting of pairs of sentences in two languages. For popular languages, training data sets contain on the order of hundreds of millions of words. Even in these cases, though, data is rarely available in domains outside of news or government proceedings and one of the language pair is

Monsieur l'Orateur, ma question est simple.
Mr. Speaker, my question is simple.

M. Roch La Salle (Joliette) propose:
Mr. Roch La Salle (Joliette) moved:

...beauty is life when life unveils her holy face.
...beauty is life when life unveils her holy face.

Figure 5.1 Three foreign/English parallel bilingual examples. The first line is French and second English (reprint of figure 5.1).

almost always English. For less popular languages, there is little data readily available. Given this, along with the cost of generating training data, identifying the most useful examples is a crucial first step in minimizing the amount of training data required.

In Chapter 1, we examined three parallel bilingual examples. (For convenience, these examples are shown again in Figure 5.1.) Qualitatively these examples are poor choices for training a translation system: the examples do not provide additional information in the context of other examples. Many other example properties can also result in inferior examples. For machine translation, like other natural language processing applications, a large amount of data preprocessing is required, for example tokenization, sentence splitting and sentence alignment. These preprocessing steps introduce noise in the data. Human translator performance also gives rise to lower quality examples: translators make mistakes. This can occur due to human error or lack of language familiarity and results in poorly phrased translations or inappropriate sentence orderings. All these factors can reduce the usefulness of an example for use during training.

Given a set of training examples $t_1, t_2, ..., t_n$ our goal is to order these examples based on their usefulness when used to train a statistical machine translation system. We accomplish this by first automatically evaluating multiple random subsets of the examples. Given these example subsets and associated scores, the problem then becomes a parameter estimation problem where the goal is to assign example scores that best explain the subset scores. The examples are then ranked based on these scores.

In this chapter, we make three main contributions:

All examples are not equally useful We describe an algorithm for ranking examples based on their usefulness for training a machine translation system. When used to train a translation system, the top ranked examples perform significantly better than random selection of examples.

Learning framework No ranking or score information is readily available for example usefulness. We describe a framework for generating training data for learning example scores based on automatically scored random subsets of the examples.

MT example feature analysis Using the learned ranking, we compare feature values for the top ranking examples to a random set of examples. From this comparison, we identify salient features for discriminating good labeled and unlabeled examples.

Identifying the usefulness of examples builds an important foundation for many avenues of future research. Given useful examples, a model can be built and used to identify those unlabeled examples (e.g. untranslated sentences) worth labeling in a framework similar to active learning. Also, given ranked examples, we can identify features of good and bad examples. These features can provide useful insights into language characteristics and help analyze the translation system learning method. Finally, a model of example usefulness can also be used for outlier detection to remove those examples that are inappropriate.

In this chapter, we first discuss previous research. We then describe our method including theoretical analysis. In Section 5.4 we compare machine translation systems trained using the most useful examples with those trained on randomly selected examples. We conclude with a brief analysis of applications of this research.

5.2 Related Work

Although the question of example usefulness has not previously been explicitly addressed, there are many related problems that have been examined.

Active learning Active learning research attempts to find those examples that are most useful *given* the current trained model. In many different domains and applications, researchers have shown that given a partially trained model, some examples are more useful. Active learning techniques commonly use the uncertainty of the model on unseen examples as a selection criterion [38]. Alternatively, methods have been proposed that select examples with maximal disagreement to reduce the version space [21] or select examples to minimize an evaluation function on a test set [58]. Because of the complexity of current machine translation methods, these methods are not applicable. Our approach differs from most active learning research because we are interested in finding universally good examples for a given training method, rather than for a partially trained model.

Boosting Boosting continually reweights examples during training to focus on those examples that are more difficult [20]. As the boosting algorithm continuously reweights the examples, the learning algorithm is forced to focus on problematic examples. While the example weightings do provide a ranking, this ranking is for example difficulty and it is not clear how this relates to example usefulness. As with active learning techniques, boosting reweights examples with respect to the current model. Also, boosting requires the training algorithm to accept weighted examples, which is not easily done for MT.

Feature selection Feature selection methods attempt to reweight features [37], rank features [19] or find good subsets of features [7]. If we view training examples as features, many of the approaches suggested in the feature selection literature are relevant. However, the techniques applied and data available for feature selection is not appropriate for the example ranking problem. These methods rely on additional information, such as class labels, or involve a search of the feature space, which is not tractable for machine translation.

Hypergraph approximation Hypergraphs are an extension of traditional graphs where hyperedges describe the weighting between two or more vertices in the graph [1]. Since

most graph algorithms function on traditional graphs, a common problem is to find a traditional graph approximation of a given hypergraph. If we view the set of examples as graph edges and the selected subsets as defining hyperedges in this graph, our proposed framework is similar to a hypergraph approximation problem, where the goal is to find the edge weights between individual vertices. Our proposed solution is similar to the commonly used Clique Expansion method.

5.3 Ranking Examples

The goal is to rank the examples $t_1, t_2, ..., t_n$ based on their usefulness in training a translation system. Because of the time required to train and evaluate a translation system, methods that involve searching the space of example combinations are not tractable [7]. Instead, we rank the examples using a single pass method that only requires an initial set of translation systems to be trained and evaluated.

5.3.1 Generating training data

One of the key challenges of this problem is that there is no a priori example rankings or example scores to learn from. Information can, however, be obtained about the performance of sets of examples. Given a set of examples, a machine translation system is trained. This system is then evaluated on a development set to obtain a score for that set.

To generate the training data we select m random subsets of l examples, $f_1, f_2, ..., f_m$, from the n original examples. A machine translation system is trained on each of these subsets and evaluated using an automatic evaluation measure (e.g. BLEU using a development set) to generate a score for each subset, $s(f_j)$. To evaluate the general usefulness of the examples, the development set must be large enough to minimize increases in the score due to spurious overlap between the randomly selected subsets and the development set. These subset/score pairs $\{f_1, s(f_1)\}$, $\{f_2, s(f_2)\}$, ..., $\{f_m, s(f_m)\}$ define the basic training data. This framework only requires that a model can be trained and auto-

matically evaluated and is therefore applicable to a wide range of applications besides machine translation.

5.3.2 Modeling example contribution

Rather than directly learning a ranking of the examples, for each example we learn a score, $s(t_i)$. We then sort the examples based on this score to obtain the ranking. Given the training data generated as described above, the one remaining component required is a model of how training examples contribute to the performance of the trained system. Specifically, we need a model of how the example scores relate to the automatic evaluation score of a system trained using those examples.

Because of the complexity of the statistical translation process and because multiple different models are combined during translation, it is not obvious how a single example contributes to the final translation system score. There has been some research on the computational complexity of translation [66, 23], but we are interested in the example contribution during training, not runtime complexity.

We assume that the subset score is a linear combination of the example scores:

$$s(f_j) = \sum_{t_i \in f_j} s(t_i)$$

This linearity assumption is reasonable for phrase-based translation models. A number of similar phrase translation models have been discussed that align a foreign phrase with a distinct English phrase [42, 50]. During training, each example in a phrase-based system is decomposed into a finite number of phrases. Ignoring distortion effects, each of these phrases can be seen as contributing to the end performance of the translation system by adding a new, unseen phrase, or by increasing the model precision for a seen phrase pair. In either case, the contribution of a single example is at most a constant amount.

5.3.3 Calculating example scores

The model described above, in combination with the training data, $\{f_1, s(f_1)\}$, $\{f_2, s(f_2)\}$, ..., $\{f_m, s(f_m)\}$, defines a constraint satisfaction problem. We calculate an example's score as the average of the subset scores that example is contained in

$$\tilde{s}(t_i) = \frac{\sum_{f_j \in T_i} s(f_j)}{|T_i|}$$

where T_i is the set of subsets that contained t_i and $|T_i|$ is the number of subsets that contain t_i. This solution is intuitive, efficient to calculate and can be analyzed to better understand the performance trade-offs for different model parameters. Other methods do exist for solving this problem, such as linear programming or perceptron learning, however, initial investigation into these other methods proved inferior or intractable.

5.3.4 Theoretical justification

In this section, we show that assuming each subset score, $s(f_j)$, is a linear combination of the example scores contained in f_j, as the number of random subsets increases the approximate example score, $\tilde{s}(t_i)$, approaches the actual example score, $s(t_i)$, plus a constant and a small error factor that is dependent on the ratio of the subset size to the total number of examples: as $m \to \infty$, $\tilde{s}(t_i) \to s(t_i) + C + \varepsilon$. Since we are only interested in ranking the examples, the only factor that distinguishes the estimated ranking from the actual ranking is the ε.

Starting with our example score approximation, we can derive the following:

$$\begin{aligned}
\tilde{s}(t_i) &= \frac{\sum_{f_j \in T_i} s(f_j)}{|T_i|} \\
&= \frac{\sum_{f_j \in T_i} \sum_{t_k \in f_j} s(t_k)}{|T_i|} \\
&= \frac{|T_i| s(t_i) + \sum_{f_j \in T_i} \sum_{t_k \in f_j : k \neq i} s(t_k)}{|T_i|} \\
&= s(t_i) + C_i
\end{aligned}$$

where C_i is an example specific constant. The first step is derived by applying the linearity assumption. In the second step, since we know that each subset in T_i contains the example t_i, we can separate the score contribution of t_i from each of the subsets.

Although each example score approximation contains a different constant, we can show that these constants differ by only a small value. Specifically, as $m \rightarrow \infty$, the difference between any two constants reduces to a small error value, that is $|C_p - C_q| \rightarrow \varepsilon$.

Since the subsets are generated by randomly selecting examples, according to the law of large numbers, as the number of subsets m increases, the proportion of subsets that a given example occurs in will approach the distribution mean, $\frac{l}{n}$. The constant then reduces to

$$\begin{aligned} C_i &= \frac{|T_i|\frac{l}{n}\sum_{t_k:k\neq i} s(t_k)}{|T_i|} \\ &= \frac{l}{n}\sum_{t_k:k\neq i} s(t_k) \end{aligned}$$

and the difference between any two constants is

$$\begin{aligned} |C_p - C_q| &= |\frac{l}{n}\sum_{t_k:k\neq p} s(t_k) - \frac{l}{n}\sum_{t_k:k\neq q} s(t_k)| \\ &= |\frac{l}{n}(-s(t_p) + \sum_{t_k} s(t_k) - (-s(t_q) + \sum_{t_k} s(t_k)))| \\ &= |\frac{l}{n}(s(t_q) - s(t_p))| \equiv \varepsilon \end{aligned}$$

Since $l < n$ the $\frac{l}{n}$ reduces the possible error between the actual and approximate scores. This error factor quantifies an intuitive trade-off between the precision of the example score approximation and the number of subsets containing that example. For small l, the $\frac{l}{n}$ reduces the constant variability, thereby increasing precision of the example score approximation. However, since the training sizes are small, the example

occurs in fewer subsets. Furthermore, training and evaluating each individual subset becomes less accurate since training a machine translation system becomes less accurate as the training size decreases. On the other hand, for larger l, the examples occur in many more subsets and subset evaluation is more accurate. But, estimating the example score is more difficult. Subsets contain many examples and it becomes more difficult to associate the subset score with the many example scores in that subset.

5.4 Experiments

In Section 5.3, we describe a method for ranking examples based on usefulness. Without previous knowledge about example performance, determining the quality of this ranking is similar to determining the ranking itself. Instead, we examine the quality of the most useful examples, as determined by the ranking. If the ranking is good, training on these examples should result in a superior performing translation system than a random selection of examples. This formulation also allows us to answer the question of whether all examples are equally useful.

We compare the translation performance of the most useful examples to ten translation systems trained on randomly selected subsets of the examples. For consistency, both the random subsets and the most useful examples contain the same number of *words*. This is critical since longer examples tend to perform better than shorter examples, but require more effort to translate. The important question is whether two training sets, which took similar effort to construct, perform differently.

For any given test set, there are some training examples that will result in better performance due to phrasal overlap. We are not interested in these local effects and want to find examples that perform generally better on different test sets. To accomplish this, we tested on multiple large test sets.

We conclude this section with one application of the example ranking. Once “good” examples have been learned, we can identify characteristics of these good examples. We compare feature values for the top ranking examples and a random selection of examples

for previously proposed features.

5.4.1 Experimental setup

We selected the first $n = 47{,}282$ unique sentence pairs (1.03 million words) from the Hansard corpus as the set of examples to rank. $m = 500$ translation systems were generated using 200,000 word (approximately 9500 examples) training subsets selected randomly and scored on a 10,000 sentence development set with BLEU. This data was then used to generate the example ranks using the method described in Section 5.3.

After ranking, the best 200,000 word training set was created from the top ranked examples. This was then used to generate a translation system we denote "best". For comparison, we generated ten random systems trained on random example subsets of 200,000 words, denoted "random $\{$1-10$\}$".

We used the Pharaoh training algorithm[1] and decoder [32] as the translation system in all phases of the experiments. We selected model parameters using maximum BLEU discriminative training [48] run on the 47K sentence pairs and those parameters were used for all of the translation systems.

5.4.2 Selecting the most useful examples

We tested the performance of the different translation systems on ten different test sets of 10,000 sentences. If our ranking method performs well, then the system trained on the top ranked examples should perform better than those trained on randomly selected examples. Table 5.1 shows the average BLEU scores over the ten test sets for "best" and the random systems.

The system trained on the top ranking examples performs 0.18 BLEU points higher than any of the random systems and on average 0.34 BLEU points higher. Given the small training sizes used for these systems, these represent substantial differences. More importantly, these differences are significant. Table 5.1 also shows the significance re-

[1] http://www.iccs.informatics.ed.ac.uk/~pkoehn

Table 5.1 Average score over the 10 test sets and the paired t-test significance for the "best" system compared to 10 random systems. One, two and three triangles indicate significance at the 95%, 99%, 99.9% confidence level respectively.

	average score	t-test vs. best
best	0.1800	-
random 1	0.1750	△△△
random 2	0.1782	△△△
random 3	0.1777	△△
random 4	0.1764	△△△
random 5	0.1765	△
random 6	0.1760	△△△
random 7	0.1745	△△△
random 8	0.1781	△△
random 9	0.1769	△△△
random 10	0.1766	△△

sults of a paired t-test between "best" and the random systems over the 10 different test sets. "best" performs significantly better than all of the random systems.

Figure 5.2 shows a plot of all of the individual test scores for the different systems. Since we are only interested in comparing the top ranking examples to a random selection, the random systems are all plotted using the same symbol. In almost all of the test sets, the "best" score is the highest score for the test set. Over the 100 different random test scores (10 test sets with 10 random systems), the "best" method performed better on 97.

The "best" and the random systems are trained on 200,000 word training sets. To attempt to quantify how much better the "best" examples are than random, we repeated the random experiments on training sets of 215,000 and 220,000 word data sets. For 215,000 the average random test score is 0.1784 and for 220,000 the average score is 0.1804. The "best" example data set of size 200,000 words performed better than a random selection using 7.5% more data and only slightly worse than a random selection of examples 10% larger.

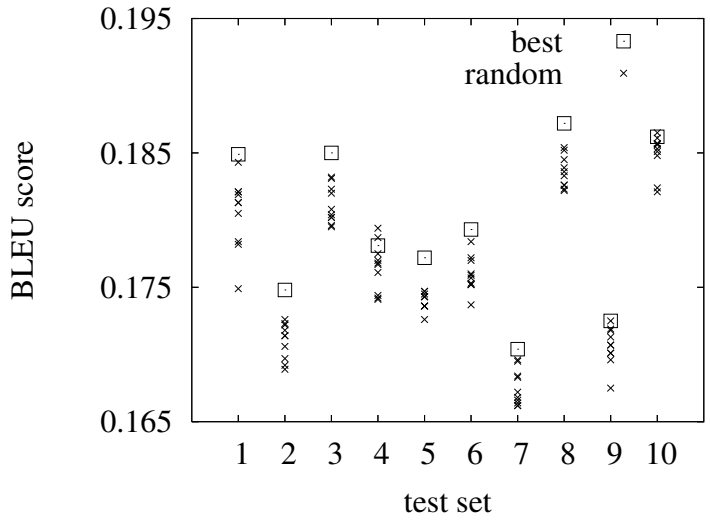


Figure 5.2 Individual test scores for "best" and "random {1-10}" for the ten different test sets. For clarity, all random scores are marked with the same symbol.

5.4.3 Analyzing example features

Once the examples are ranked, this information can be used for a number of different applications. The learned example scores can be used to predict example usefulness on *unlabeled* examples. Features can be extracted from both the ranked examples and unlabeled examples. These features can then be used in a wide range of learning frameworks to identify unlabeled examples that most closely resemble the useful examples. Also, extracted features are useful as an analytical tool for characterizing examples and for better understanding statistical models. Finally, examining features from different portions of the ranking provides assurance that our ranking method is accomplishing something non-trivial.

For each of the examples, we extracted eight different features. These features have previously been suggested for use in confidence estimation for MT [6]. We compared the feature values of the top ranking set of examples and a randomly selected set of examples. If a feature is a good predictor of example usefulness, then the value of that feature should be different between these two sets.

We analyzed eight features that highlight a wide range of example characteristics.

We intentionally tried to select those features that rely only on the foreign portion of the example, so that they may be of future use in selecting unlabeled examples. Only the length ratio and overlap proportion feature use both the foreign and English portion of the example.

Some of the features characterize intrinsic properties of the examples such as the length or the similarity between the foreign and English text. The language model probability features try and quantify the fluency of the text. The n-gram quartile features capture the frequency characteristics of the foreign words. Finally, some of the features assess how likely a given example is with respect to a machine translation model. We trained a model on the 47K training examples and used this translation system to translate the foreign portion of the examples and to gather various translation statistics.

Length ratio Number of words in the English sentence divided by the number of words in the foreign sentence.

Overlap proportion Proportion of the words in the foreign sentence that occur in the English sentence.

Foreign length Number of words in the foreign sentence.

Foreign language model score Log probability of the foreign sentence based on a foreign trigram language model.

Translated sentence language model score Log probability of the translated sentence based on an English trigram language model.

Translation model probability Product of the translation rule probabilities for the phrase rules used to translate the foreign sentence[2].

Phrase rules used Number of phrase rules used to translate the foreign sentence.

n-gram quartiles Foreign word frequencies were counted on a corpus. Words only occurring once were removed. The words were then sorted based on their frequency

[2]This is different from the true translation model probability since the phrase reording probabilities are not included.

and divided into four quartiles. The first quartile contains the quarter least frequent words, second quartile the next quarter most frequent words, etc. Given an example, the proportion of words in the foreign sentence in each quartile is recorded as "Quartile $\{$1-4$\}$".

Table 5.2 shows the average example feature scores for the top ranked French/English examples and a random set of examples both containing 200,000 words. Only the translation rule probability is significantly different between the two sets. One cause for this difference could be a length difference in examples. Longer examples on average require more rules to be used, resulting in lower overall probability. However, since the lengths and the number of rules are similar between the subsets, there is an inherent difference in the translation model probability: the top ranking examples are more likely.

Though not significantly different, the other analyzed features may still provide useful information. The second quartile of words was moderately ($p < .25$) different for the two sets. Also, we only examined these features individually. Future analysis is required to see if a combination of these features proves more useful.

One final motivation for examining example features is to verify the quality of the ranking algorithm. A good ranking algorithm should rank the examples based on multiple dimensions. It is therefore reassuring that none of the individual features correlated strongly with the learned ranking.

5.5 Future Work

There are a number of future directions for this research. Given knowledge of what good examples are, this information can be used as a stepping stone for other applications. One of the main motivations for investigating the question of example usefulness was to incorporate this knowledge in an active learning framework. Given a ranking of examples, a model of useful examples can be learned based on extracted features [6, 47]. Given this model, unlabeled examples with similar characteristics can then be identified.

Throughout this chapter, we focused on training examples for machine translation.

Table 5.2 Average feature scores for "best" and "random 1". Significantly different averages are shown in **bold** and moderately significant differences in *italics*.

	best	random 1
Length ratio	1.025	1.021
Overlap	0.1375	0.1381
Foreign length	21.98	21.92
Foreign LM	-42.31	-42.07
Translated LM	-43.50	-43.37
Translation prob	**0.0014**	**0.0012**
Rules used	7.123	7.110
Quartile 1	0.0098	0.0093
Quartile 2	*0.0047*	*0.0043*
Quartile 3	0.0143	0.0143
Quartile 4	0.9713	0.9721

Our learning framework for ranking the examples only requires that random subsets of the examples can be used to train a model and that the resulting model can be evaluated automatically. There are many applications both in natural language processing and others that satisfy these requirements. An interesting question is how well this method will perform on these different applications.

Besides applications of this method, there are still many open questions about the performance of the method itself. On the development set, the best ranking examples score 0.1845, the worst ranking examples 0.1727 and all of the 500 random subsets used for training fall in between these two scores. On the 10 tests sets, this ordering is not as well preserved. As we saw in section 5.4, the best examples are significantly better than all of the randomly subsets. However, the worst ranking examples are only significantly worse than 6 of the 10 random subsets, with an average score of 0.176 on the test sets.

There are many possible explanations for this behavior. Theoretical analysis of the method showed that the approximation accuracy depended on two factors: 1) the number of subsamples trained and evaluated and 2) the ratio of the training subset size to the total number of examples. More analysis is needed, to determine empirically how different parameter settings affect performance.

Another explanation are the model assumptions. We suggest a linear model for approximating the contribution of individual examples. This linearity assumption is reasonable for phrase-based systems, but not perfect. The linear model assumes independence between training examples, which is not the case. A model that maintains the linearity assumption, but better models example overlap could improve ranking performance.

5.6 Conclusion

In this chapter we have suggested a new framework for determining the usefulness of examples based on the evaluated performance of random subsets of those examples. For machine translation, we showed that all examples are NOT equally useful. Using the ranking obtained by our method, the most useful examples are identified. When used to train a new translation system these examples perform significantly better than a random selection of examples on a large test set. We also provide theoretical justification for our method that shows as the number of example subsets increases, the performance of our method is reasonable and predictable. Finally, we provided an analysis of a number of features that identified both the translation probability and n-gram quartiles as a possible discriminating features for identifying “good” examples.

6

Paraphrasing for Automatic Evaluation

An important component for many natural language tasks is evaluation. Until recently, evaluation of machine translation was done by hand. In the last few years, a number of automatic evaluation measures have been proposed that correlate well with human evaluations. These evaluation measures have become crucial components for many stages in development. In this chapter, we examine the benefit of using paraphrasing to assist these evaluation measures.

6.1 Introduction

The use of automatic methods for evaluating machine generated text is quickly becoming mainstream in natural language processing. The most notable examples in this category include measures such as BLEU and ROUGE which drive research in the machine translation and text summarization communities. These methods assess the quality of a machine generated output by considering its similarity to a reference text written by a human. Ideally, the similarity would reflect the semantic proximity between the two. In practice, this comparison breaks down to n-gram overlap between the reference and the machine output.

Table 6.1 A reference sentence and corresponding machine translation from the NIST 2004 MT evaluation. The two sentences share only auxiliary words.

(a) However, Israel's reply failed to completely clear the U.S. suspicions.
(b) However, Israeli answer unable to fully remove the doubts.

Consider the human translation and the machine translation of the same Chinese sentence shown in Table 6.1. While the two translations convey the same meaning, they share only auxiliary words. Clearly, any measure based on word overlap will penalize a system for generating such a sentence. The question is whether such cases are common phenomena or infrequent exceptions. Empirical evidence supports the former. Analyzing 10,728 reference translation pairs[1] used in the NIST 2004 machine translation evaluation, we found only 21 (less than 0.2%) that are are identical. Moreover, 60% of the pairs differ in at least 11 words. These statistics suggest that without accounting for paraphrases, automatic evaluation measures may never reach the accuracy of human evaluation.

As a solution to this problem, researchers have suggested using multiple references to refine automatic evaluation. [53] shows that expanding the number of references reduces the gap between automatic and human evaluation. However, very few human annotated sets are augmented with multiple references and those that are available are relatively small in size. Moreover, access to several references does not guarantee that the references will include the same words that appear in machine generated text.

In this chapter, we explore the use of paraphrasing methods for refinement of automatic evaluation techniques. Given a reference sentence and a machine generated sentence, we seek to find a paraphrase of the reference sentence that is closer in wording to the machine output than the original reference. For instance, given the pair of sentences in Table 6.1, we automatically transform the reference sentence (a) into

However, Israel's *answer* failed to completely *remove* the U.S. suspicions.

[1] Each pair included different translations of the same sentence, produced by two human translators.

Thus, among many possible paraphrases of the reference, we are only interested in those that use words appearing in the system output. Our paraphrasing algorithm is based on the *substitute in context* strategy. First, the algorithm identifies pairs of words from the reference and the system output that could potentially form paraphrases. We select these candidates using existing lexico-semantic resources such as WordNet. Next, the algorithm tests whether the candidate paraphrase is admissible in the context of the reference sentence. Since even synonyms cannot be substituted in any context [18], this filtering step is necessary. We predict whether a word is appropriate in a new context by analyzing its distributional properties in a large body of text. Finally, paraphrases that pass the filtering stage are used to rewrite the reference sentence.

We apply our paraphrasing method in the context of machine translation evaluation. Using this strategy, we generate a new sentence for every pair of human and machine translated sentences. This synthetic reference then replaces the original human reference in automatic evaluation.

The key findings of our work are as follows:

Automatically generated paraphrases improve the accuracy of the automatic evaluation methods. Our experiments show that evaluation based on paraphrased references gives a better approximation of human judgments than evaluation that uses original references.

The quality of automatic paraphrases determines their contribution to automatic evaluation. By analyzing several paraphrasing resources, we found that the accuracy and coverage of a paraphrasing method correlate with its utility for automatic MT evaluation.

Our results suggest that researchers may find it useful to augment standard measures such as BLEU and ROUGE with paraphrasing information thereby taking more semantic knowledge into account.

In the following section, we provide an overview of existing work on automatic paraphrasing. We then describe our paraphrasing algorithm and explain how it can be used in an automatic evaluation setting. Next, we present our experimental framework and data and conclude by presenting and discussing our results.

6.2 Related Work

Automatic Paraphrasing and Entailment Our work is closely related to research in automatic paraphrasing, in particular, to sentence level paraphrasing [4, 52, 55]. Most of these approaches learn paraphrases from a parallel or comparable monolingual corpora. Instances of such corpora include multiple English translations of the same source text written in a foreign language, and different news articles about the same event. For example, Pang et al. [52] expand a set of reference translations using syntactic alignment, and generate new reference sentences that could be used in automatic evaluation.

Our approach differs from traditional work on automatic paraphrasing in goal and methodology. Unlike previous approaches, we are not aiming to produce *any* paraphrase of a given sentence since paraphrases induced from a parallel corpus do not necessarily produce a rewriting that makes a reference closer to the system output. Thus, we focus on words that appear in the system output and aim to determine whether they can be used to rewrite a reference sentence.

Our work also has interesting connections with research on automatic textual entailment [15], where the goal is to determine whether a given sentence can be inferred from text. While we are not assessing an inference relation between a reference and a system output, the two tasks face similar challenges. Methods for entailment recognition extensively rely on lexico-semantic resources [25, 26], and we believe that our method for contextual substitution can be beneficial in that context.

Automatic Evaluation Measures A variety of automatic evaluation methods have been recently proposed in the machine translation community [46, 43, 53]. All these metrics compute n-gram overlap between a reference and a system output, but measure the overlap in different ways. Our method for reference paraphrasing can be combined with any of these metrics. In this chapter, we report experiments with BLEU due to its wide use in the machine translation community.

Recently, researchers have explored additional knowledge sources that could enhance automatic evaluation. Examples of such knowledge sources include stemming

and TF-IDF weighting [2, 3]. Our work complements these approaches: we focus on the impact of paraphrases, and study their contribution to the accuracy of automatic evaluation.

6.3 Algorithm

The input to our method consists of a reference sentence $R = r_1 \dots r_m$ and a system-generated sentence $W = w_1 \dots w_p$ whose words form the sets $\mathcal{R}$ and $\mathcal{W}$ respectively. The output of the model is a synthetic reference sentence S_{RW} that preserves the meaning of R and has maximal word overlap with W. We generate such a sentence by substituting words from R with contextually equivalent words from W.

Our algorithm first selects pairs of candidate word paraphrases, and then checks the likelihood of their substitution in the context of the reference sentence.

Candidate Selection We assume that words from the reference sentence that already occur in the system generated sentence should not be considered for substitution. Therefore, we focus on unmatched pairs of the form $\{(r, w)|r \in \mathcal{R} - \mathcal{W}, w \in \mathcal{W} - \mathcal{R}\}$. From this pool, we select candidate pairs whose members exhibit high semantic proximity. In our experiments we compute semantic similarity using WordNet, a large-scale lexico-semantic resource employed in many NLP applications for similar purposes. We consider a pair as a substitution candidate if its members are synonyms in WordNet.

Applying this step to the two sentences in Table 6.2, we obtain two candidate pairs (**home**, **place**) and (**difficult**, **hard**).

Contextual Substitution The next step is to determine for each candidate pair (r_i, w_j) whether w_j is a valid substitution for r_i in the context of $r_1 \dots r_{i-1} \square r_{i+1} \dots r_m$, where '$\square$' denotes the location of r_i and possible substitution location for w_j in the sentence. This filtering step is essential because synonyms are not universally substitutable[2]. Con-

[2]This can explain why previous attempts to use WordNet for generating sentence-level paraphrases [4, 55] were unsuccessful.

Table 6.2 A reference sentence and a corresponding machine translation. Candidate paraphrases are in bold.

(a) It is **hard** to believe that such tremendous changes have taken **place** for those people and lands that I have never stopped missing while living abroad.
(b) For someone born here but has been sentimentally attached to a foreign country far from **home**, it is **difficult** to believe this kind of changes.

sider the candidate pair (**home**, **place**) from our example (see Table 6.2). Words **home** and **place** are paraphrases in the sense of "habitat", but in the reference sentence "**place**" occurs in a different sense, being part of the collocation "take place". In this case, the pair (**home**, **place**) cannot be used to rewrite the reference sentence.

We formulate contextual substitution as a binary classification task: given a context $r_1 \dots r_{i-1} \square r_{i+1} \dots r_m$, we aim to predict whether w_j can occur in this context at position i. For each candidate word w_j we train a classifier that models contextual preferences of w_j. To train such a classifier, we collect a large corpus of sentences that contain the word w_j and an equal number of randomly extracted sentences that do not contain this word. The former category forms positive instances, while the latter represents the negative. For the negative examples, a random position in a sentence is selected for extracting the context. This corpus is acquired automatically, and does not require any manual annotations.

We represent context by n-grams and local collocations, features typically used in supervised word sense disambiguation. Both n-grams and collocations exclude the word w_j. An n-gram is a sequence of n adjacent words appearing in $r_1 \dots r_{i-1} \square r_{i+1} \dots r_m$. A local collocation also takes into account the position of an n-gram with respect to the target word. To compute local collocations for a word at position i, we extract all n-grams ($n = 1 \dots 4$) beginning at position $i - 2$ and ending at position $i + 2$. To make these position dependent, we prepend each of them with the length and starting position.

Once the classifier[3] for w_j is trained, we apply it to the context $r_1 \ldots r_{i-1} \square r_{i+1} \ldots r_m$. For positive predictions, we rewrite the string as $r_1 \ldots r_{i-1} w_j r_{i+1} \ldots r_m$. In this formulation, all substitutions are tested independently.

For the example from Table 6.2, only the pair (**difficult**, **hard**) passes this filter, and thus the system produces the following synthetic reference:

> For someone born here but has been sentimentally attached to a foreign country far from home, it is **hard** to believe this kind of changes.

The synthetic reference keeps the meaning of the original reference, but has a higher word overlap with the system output.

One of the implications of this design is the need to develop a large number of classifiers to test contextual substitutions. For each word to be inserted into a reference sentence, we need to train a separate classifier. In practice, this requirement is not a significant burden. The training is done off-line and only once, and testing for contextual substitution is instantaneous. Moreover, the first filtering step effectively reduces the number of potential candidates. For example, to apply this approach to the 71,520 sentence pairs from the MT evaluation set (described in Section 6.4.1), we had to train 2,380 classifiers.

We also discovered that the key to the success of this approach is the size of the corpus used for training contextual classifiers. We derived training corpora from the English Gigaword corpus, and the average size of a corpus for one classifier is 255,000 sentences. We do not attempt to substitute any words that have less than 10,000 appearances in the Gigaword corpus.

6.4 Experiments

Our primary goal is to investigate the impact of machine generated paraphrases on the accuracy of automatic evaluation. We focus on automatic evaluation of machine

[3] In our experiments, we used the publicly available BoosTexter classifier [59] for this task.

translation due to the availability of human annotated data in that domain. The hypothesis is that by using a synthetic reference translation, automatic measures better approximate human evaluation. In Section 6.4.2, we test this hypothesis by comparing the performance of BLEU scores with and without synthetic references.

Our secondary goal is to study the relationship between the quality of paraphrases and their contribution to the performance of automatic machine translation evaluation. In Section 6.4.3, we present a manual evaluation of several paraphrasing methods and show a close connection between intrinsic and extrinsic assessments of these methods.

6.4.1 Experimental setup

We begin by describing the data set and the alternative paraphrasing methods considered in our experiments. BLEU is the basic evaluation measure that we use in our experiments. To augment BLEU with paraphrasing information, we substitute each reference with the corresponding synthetic reference.

Data

We use the Chinese portion of the 2004 NIST MT dataset. This portion contains 200 Chinese documents, subdivided into a total of 1788 segments. Each segment is translated by ten machine translation systems and by four human translators. A quarter of the machine-translated segments are scored by human evaluators on a one-to-five scale along two dimensions: adequacy and fluency. We use only adequacy scores, which measure how well content is preserved in the translation.

Alternative paraphrasing techniques

To investigate the effect of paraphrase quality on automatic evaluation, we consider two alternative paraphrasing resources: Latent Semantic Analysis (LSA), and Brown et al. clustering [10]. These techniques are widely used in NLP applications, including language modeling, information extraction, and dialog processing [25, 61, 44]. Both

Table 6.3 Sample of paraphrasings produced by each method based on the corresponding system translation. Paraphrased words are in **bold** and filtered words underlined.

Reference	The monthly magazine “Choices” has won the deep trust of the residents. The current Internet edition of “Choices” will give full play to its functions and will help consumers get quick access to market information.
System	The public has a lot of faith in the “Choice” monthly magazine and the Council is now working on a web version. This will enhance the magazine’s function and help consumer to acquire more up-to-date market information.
WordNet	The monthly magazine “Choices” has won the deep **faith** of the residents. The current Internet **version** of “Choices” will give full play to its functions and will help consumers **acquire** quick access to market information.
ContextWN	The monthly magazine “Choices” has won the deep trust of the residents. The current Internet **version** of “Choices” will give full play to its functions and will help consumers **acquire** quick access to market information.
LSA	The monthly magazine “**Choice**” has won the deep trust of the residents. The current **web** edition of “**Choice**” will give full play to its functions and will help **consumer** get quick access to market information.
Brown	The monthly magazine “Choices” has won the deep trust of the residents. The current Internet **version** of “Choices” will give full play to its functions and will help consumers get quick access to market information.

techniques are based on distributional similarity. The Brown clustering is computed by considering mutual information between adjacent words. LSA is a dimensionality reduction technique that projects a word co-occurrence matrix to lower dimensions. This lower dimensional representation is then used with standard similarity measures to cluster the data. Two words are considered to be a paraphrase pair if they appear in the same cluster.

We construct 1000 clusters employing the Brown method on 112 million words from the North American News Text corpus. We keep the top 20 most frequent words for each cluster as paraphrases. To generate LSA paraphrases, we used the Infomap software[4] on a 34 million word collection from the same corpus[5]. We used the default parameter settings: a 20,000 word vocabulary, the 1000 most frequent words (minus a stop-list) for features, a 15 word context window on either side of a word, a 100 feature reduced representation, and the 20 most similar words as paraphrases.

We experimented with several parameter settings for LSA and Brown methods, but do not claim that the selected settings are necessarily optimal. However, these methods present sensible comparison points for understanding the relationship between paraphrase quality and its impact on automatic evaluation.

Table 6.3 shows synthetic references produced by the different paraphrasing methods.

Evaluating evaluation metrics

The standard way to analyze the performance of an automatic evaluation metric in machine translation is to compute the Pearson correlation between the automatic metric and human scores [53, 33, 39, 64]. Let X and Y be two sets of n data points where each point in X corresponds to a unique point in Y. The Pearson correlation between these points is

$$r = \frac{\sum_{i=1}^{n} x_i x_i - \frac{\sum x \sum y}{n}}{\sqrt{(\sum x^2 - \frac{(\sum x)^2}{n})(\sum y^2 - \frac{(\sum y)^2}{n})}}$$

[4] `http://infomap-nlp.sourceforge.net`

[5] For computational reasons, a smaller set was used.

Table 6.4 Pearson adequacy correlation scores for rewriting using one and two references, averaged over ten runs.

Method	1 reference	2 references
BLEU	0.9657	0.9743
WordNet	0.9674	0.9763
ContextWN	**0.9677**	**0.9764**
LSA	0.9652	0.9736
Brown	0.9662	0.9744

Pearson correlation estimates how linearly dependent two sets of values are. The Pearson correlation values range from 1, when the scores are perfectly linearly correlated, to -1, in the case of inversely correlated scores. For evaluating automatic evaluation, the correlation is calculated between the automatic evaluation scores and human evaluation scores.

To calculate the Pearson correlation, we create a document by concatenating 300 segments. This results in a document for each of the ten MT systems. This strategy is commonly used in MT evaluation, because of BLEU's well-known problems with documents of small size [53, 33]. For each of the ten MT system translations, the evaluation metric score is calculated on the document and the corresponding human adequacy score is calculated as the average human score over the segments. The Pearson correlation is calculated over these ten automatic evaluation score/human adequacy score pairs [53, 64]. This process is repeated for ten different documents to obtain ten correlation scores. A paired t-test is calculated over these ten correlation scores to compute statistical significance.

6.4.2 Impact of paraphrases on machine translation evaluation

Table 6.4 shows Pearson correlation scores for BLEU and the four paraphrased augmentations, averaged over ten runs.[6] In all ten tests, our method based on contextual

[6]Depending on the experimental setup, correlation values can vary widely. Our scores fall within the range of previous researchers [53, 39].

Table 6.5 Paired t-test significance for all methods compared to BLEU and our method for one reference. Two triangles indicates significant at the 99% confidence level, one triangle at the 95% confidence level and X not significant. Triangles point towards the better method.

Method	vs. BLEU	vs. ContextWN
WordNet	◁◁	△△
ContextWN	◁◁	-
LSA	X	△△
Brown	◁◁	△

rewriting (ContextWN) improves the correlation with human scores over BLEU. Moreover, in nine out of ten tests ContextWN outperforms the method based on WordNet alone. The results of statistical significance testing are summarized in Table 6.5. All the paraphrasing methods except LSA, exhibit higher correlation with human scores than plain BLEU. Our method significantly outperforms BLEU, and all the other paraphrase-based metrics. This consistent improvement confirms the importance of contextual filtering.

The third column in Table 6.4 shows that automatic paraphrasing continues to improve correlation scores even when two human references are paraphrased using our method.

6.4.3 Evaluation of paraphrase quality

In the last section, we saw significant variations in MT evaluation performance when different paraphrasing methods were used to generate a synthetic reference. In this section, we examine the correlation between the quality of automatically generated paraphrases and their contribution to automatic evaluation. We analyze how the substitution frequency and the accuracy of those substitutions contributes to a method's performance.

We compute the substitution frequency of an automatic paraphrasing method by counting the number of words it rewrites in a set of reference sentences. Table 6.6 shows the substitution frequency and the corresponding BLEU score. The substitution

Table 6.6 Scores and the number of substitutions made for all 1788 segments, averaged over the different MT system translations

Method	Score	Substitutions
BLEU	0.0913	-
WordNet	0.0969	994
ContextWN	0.0962	742
LSA	0.0992	2080
Brown	0.0921	117

Table 6.7 Accuracy scores by two human judges and the Kappa coefficient of agreement.

Method	Judge 1 accuracy	Judge 2 accuracy	Kappa
WordNet	63.5%	62.5%	0.74
ContextWN	**75.0%**	**76.0%**	0.69
LSA	30.0%	31.5%	0.73
Brown	56.0%	56.0%	0.72

frequency varies greatly across different methods — LSA is by far the most prolific rewriter, while Brown produces very few substitutions. As expected, the more paraphrases identified, the higher the BLEU score for the method. However, this increase does not translate into better evaluation performance. For instance, our contextual filtering method removes approximately a quarter of the paraphrases suggested by WordNet and yields a better evaluation measure. These results suggest that the substitution frequency cannot predict the utility value of the paraphrasing method.

Accuracy measures the correctness of the proposed substitutions in the context of a reference sentence. To evaluate the accuracy of different paraphrasing methods, we randomly extracted 200 paraphrasing examples from each method. A paraphrase example consists of a reference sentence, a reference word to be paraphrased and a proposed paraphrase of that reference (that actually occurred in a corresponding system translation). The judge was instructed to mark a substitution as correct only if the substitution

Table 6.8 Confusion matrix for the context filtering method on a random sample of 200 examples labeled by the first judge.

	negative	positive
filtered	40	27
non-filtered	33	100

was both semantically and grammatically correct in the context of the original reference sentence.

Paraphrases produced by the four methods were judged by two native English speakers. The pairs were presented in random order, and the judges were not told which system produced a given pair. We employ a commonly used measure, Kappa, to assess agreement between the judges. We found that on all the four sets the Kappa value was around 0.7, which corresponds to substantial agreement [36].

As Table 6.7 shows, the ranking between the accuracy of the different paraphrasing methods mirrors the ranking of the corresponding MT evaluation methods shown in Table 6.4. The paraphrasing method with the highest accuracy, ContextWN, contributes most significantly to the evaluation performance of BLEU. Interestingly, even methods with moderate accuracy, i.e. 63% for WordNet, have a positive influence on the BLEU metric. At the same time, poor paraphrasing accuracy, such as LSA with 30%, does hurt the performance of automatic evaluation.

To further understand the contribution of contextual filtering, we compare the substitutions made by WordNet and ContextWN on the same set of sentences. Among the 200 paraphrases proposed by WordNet, 73 (36.5%) were identified as incorrect by human judges. As the confusion matrix in Table 6.8 shows, 40 (54.5%) were eliminated during the filtering step. At the same time, the filtering erroneously eliminates 27 positive examples (21%). Even at this level of false negatives, the filtering has an overall positive effect.

6.5 Conclusion and Future Work

This chapter presents a comprehensive study of the impact of paraphrases on the accuracy of automatic evaluation. We found a strong connection between the quality of automatic paraphrases as judged by humans and their contribution to automatic evaluation. These results have two important implications: (1) refining standard measures such as BLEU with paraphrase information moves the automatic evaluation closer to human evaluation and (2) applying paraphrases to MT evaluation provides a task-based assessment for paraphrasing accuracy.

We also introduce a novel paraphrasing method based on contextual substitution. By posing the paraphrasing problem as a discriminative task, we can incorporate a wide range of features that improve the paraphrasing accuracy. Our experiments show improvement of the accuracy of WordNet paraphrasing and we believe that this method can similarly benefit other approaches that use lexico-semantic resources to obtain paraphrases.

Our ultimate goal is to develop a contextual filtering method that does not require candidate selection based on a lexico-semantic resource. One source of possible improvement lies in exploring more powerful learning frameworks and more sophisticated linguistic representations. Incorporating syntactic dependencies and class-based features into the context representation could also increase the accuracy and the coverage of the method. Our current method only implements rewriting at the word level. In the future, we would like to incorporate substitutions at the level of phrases and syntactic trees.

Acknowledgments

The text in this chapter, in part, a reprint of the material in:
D. Kauchak and R. Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of HLT/NAACL*, pages 455–462, 2006.

The dissertation author was the primary researcher and author and the co-author listed in this publication supervised the research.

7

Contributions and Future Research Directions

In the last 10 years the performance of machine translation has drastically increased. Even from year to year, performance of the state of the art systems increases noticeably. These improvements come from many dimensions. Every year more and more data becomes available with faster and faster computers. Also, the number of translation systems being developed, both commercially and in research environments, is increasing. In 2004, 12 systems participated in the yearly NIST translation evaluations. In 2005, 20 systems participated. This increasing interest in machine translation has resulted in many translation systems available of differing design and quality. In this thesis, we have examined a number of uses for these existing MT systems for research purposes.

7.1 Summary of Contributions

In Chapters 3 and 4 we suggest a framework for post-correction of machine translation systems using translations made by the system to identify mistakes. By translating data where a correct translation is known, differences between the translation and the ground truth point to possible errors. Using only monolingual data, we learn word correction rules. On a 20,000 sentence Spanish/English data set 24,235 different word

changes are made on 78% of the sentences, with high accuracy.

Given bilingual data, we describe a method that uses a partial alignment between the machine translated sentence and human sentence to learn phrase correction rules. By generating machine translated/human translated English pairs, a simplified alignment algorithm can be used that aligns lexically identical words. Using this alignment, context-independent phrase correction rules are learned. The learned correction rules improve the BLEU score of a commercial system by 30% and statistically significantly improve the performance of a state of the art statistical phrase-based system.

In Chapter 5 we showed quantitatively that all MT examples are *not* equally useful. We describe a method for generating example usefulness training data that is applicable in many domains since it only requires trainability and evaluatability. Random subsets of the examples are used to train translation systems. These systems are evaluated using BLEU on a development set, resulting in score/subset pairs. Given these pairs, we proposed a method that efficiently determines the example rankings based on the average of the subset scores a given example occurs in. Theoretical analysis of this method shows only minor deviations from the correct ranking. Using this method, we ranked 47,282 machine translation examples. The top ranked examples perform significantly better on a large test set than randomly selecting examples. A preliminary study of the most useful examples also shows a number of possible features for discriminating useful examples.

Finally, using the output from many different machine translation systems we analyze the impact of paraphrasing on automatic evaluation measures. We show that paraphrasing does improve automatic evaluation measures. This performance increase results from the ability to identify alternate appropriate words that occur in the machine translation, but not in the reference translations. Besides improving automatic evaluation measures, this problem provides a new quantitative task for evaluating paraphrasing performance. We showed strong correlation between the improvement of the automatic evaluation methods and paraphrase accuracy.

In the process of analyzing these different applications of translation systems, we described two novel methods for determining whether a word is substitutable in the

context of a particular sentence. Given a corpus, we identify those words that co-occur significantly with the word in question using the likelihood ratio test. Given a new sentence, if one of the significant words occurs in that sentence, then the substitution is considered appropriate. The second method trains an individual classifier for each word. Given sentences that a word occurs in and does not occur in, features are extracted based on position dependent and independent occurrences of n-grams. Using this data, a classifier is learned to identify the contexts that the word can occur in.

7.2 Future Research Directions

In each chapter, we suggested continuations of the work presented in that chapter. To conclude, we summarize these suggestions and mention other future research directions that utilize pre-existing translation systems.

Learning an active learner

In Chapter 5 we described a method that, given a set of foreign/English sentence examples, ranked those examples based on their usefulness for training a translation system. One use of this ranking is for learning a model of example usefulness. This model could then be used to identify useful foreign sentences to have translated by a human translator in an active learning framework. The first step for this type of method is to suggest candidate features. [6] suggest features for confidence estimation and [47] for n-best list reranking that can be used here.

In this thesis, we did a preliminary study to identify features that correlated with the example ranking. This idea could be continued in future research to learn a model of example usefulness based on the ranking. Given examples and the extracted features, a ranking or regression method could be applied to learn a model of example usefulness. This model would then be used to identify the most useful foreign sentences to translate. For machine translation, where examples are expensive to annotate and still relatively rare in many languages and domains, an algorithm that can select the most useful foreign

sentences to generate examples from would be invaluable. Also, this type of approach has not previously been explored and may prove useful in many other domains.

Evaluating automatic evaluation measures

One of the challenges in developing automatic evaluation methods such as BLEU is evaluating these evaluation measures. The standard approach in machine translation is to measure the Pearson correlation between the automatic scores and human scores. There are a number of problems with this approach. First, the assumption of linearity is overly restrictive. The key component of an evaluation method is whether it ranks methods appropriately. The distance between scores does provide some information, but this distance does not necessarily need to correlate linearly with human scores. Isotonic regression [57] is an alternative method that assumes monotonicity, but does not require the the relationship is linear. Also, Spearman rank correlation has been suggested as a possibility. Even using Pearson correlation, there are still many free parameters: what size documents are used, how many documents are used, how multiple human scores should be incorporated and how significance should be measured. These experimental variations are rarely discussed in detail in papers, but all affect comparisons.

Improved phrase rules

In Chapter 4 we learned context-independent rules. Because of this context independence assumption, many rules were eliminated in order to obtain reasonable correction accuracy, particularly when improving the statistical phrase-based system. These rules could also be extended with context information. In Chapter 3 this allowed for a larger rule set to be learned. In Chapter 6 a contextual filtering method was used to increase the rule accuracy. In addition to lexical information, rules could also use syntactic or semantic information. For systems that were word or phrase based, this would allow additional information not available to the original system to be incorporated.

Bibliography

[1] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *Proceedings of CVPR*, pages 838–845, 2005.

[2] B. Babych and A. Hartley. Extending the BLEU evaluation method with frequency weightings. In *Proceedings of ACL*, pages 621–628, 2004.

[3] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–72, 2005.

[4] R. Barzilay and L. Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of HLT/NAACL*, pages 16–23, 2003.

[5] R. Barzilay and K. McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of ACL/EACL*, pages 50–57, 2001.

[6] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueng. *Confidence Estimation for Machine Translation*. Final report, JHU/CLSP Summer Workshop, 2003.

[7] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[8] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[9] P. Brown, S. D. Pietra, V. D. Pietra, and R. Mercer. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2), 1993.

[10] P. F. Brown, P. V. deSouza, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

[11] M. Carl and A. Way, editors. *Recent advances in example-based machine translation*. Kluwer Academic Publishers, 2003.

[12] E. Charniak. Immediate-head parsing for language models. In *Proceedings of ACL*, pages 116–123, 2001.

[13] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270, 2005.

[14] T. T. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[15] I. Dagan, O. Glickman, and B. Magnini, editors. *The PASCAL recognizing textual entailment challenge*, 2005.

[16] T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. In *Neural Computation*, pages 1895–1923, 1998.

[17] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[18] P. Edmonds and G. Hirst. Near synonymy and lexical choice. *Computational Linguistics*, 28(2):105–144, 2002.

[19] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1306, 2003.

[20] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[21] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

[22] M. Galley, M. Hopkins, K. Knight, and D. Marcu. What's in a translation rule? In *Proceedings of HLT/NAACL*, 2004.

[23] U. Germann. Greedy decoding for machine translation in almost linear time. In *Proceedings of HLT/NAACL*, pages 1–8, 2003.

[24] D. Gildea. Loosely tree-based alignment for machine translation. In *Proceedings of ACL*, pages 80–87, 2003.

[25] A. Haghighi, A. Ng, and C. Manning. Robust textual inference via graph matching. In *Proceedings of HLT/NAACL*, pages 387–394, 2005.

[26] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of ACL*, pages 274–291, 2001.

[27] W. J. Hutchins and H. L. Somers. *An Introduction to Machine Translation*. Academic Press, London, UK, 1992.

[28] K. Imamura, H. Okuma, T. Watanabe, and E. Sumita. Example-based machine translation based on syntactic transfer with statistical models. In *Proceedings of COLING*, pages 99–105, 2004.

[29] D. Kauchak and R. Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of HLT/NAACL*, pages 455–462, 2006.

[30] D. Kauchak and C. Elkan. Learning rules to improve a machine translation system. In *Proceedings of ECML*, pages 205–216, 2003.

[31] P. Koehn. Europarl: A multilingual corpus for evaluation of machine translation. Unpublished: http://people.csail.mit.edu/koehn/publications/europarl/, 2002.

[32] P. Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA*, pages 115–124, 2004.

[33] P. Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395, 2004.

[34] P. Koehn and K. Knight. Knowledge sources for word-level translation models. In *Proceedings of EMNLP*, pages 27–35, 2001.

[35] B. Krenn. Collocation mining: Exploiting corpora for collocation identification and representation. In *Proceedings of KONVENS*, pages 209–214, 2000.

[36] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174, 1977.

[37] M. H. Law, A. K. Jain, and M. A. T. Figueiredo. Feature selection in mixture-based clustering. In *Proceedings of NIPS*, pages 625–632, 2003.

[38] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR*, pages 3–12, 1994.

[39] C. Lin and F. Och. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of COLING*, pages 501–507, 2004.

[40] E. Macklovitch and M. Hannan. Line'em up: Advances in alignment technology and their impact on translation support tools. In *Proceedings of AMTA*, pages 41–57, 1996.

[41] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[42] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP*, pages 133–139, 2002.

[43] I. D. Melamed, R. Green, and J. P. Turian. Precision and recall of machine translation. In *Proceedings of HLT/NAACL*, pages 61–63, 2003.

[44] S. Miller, J. Guinness, and A. Zamanian. Name tagging with word clusters and discriminative training. In *Proceedings of HLT/NAACL*, pages 337–342, 2004.

[45] S. Nirenburg, J. Carbonell, M. Tomita, and K. Goodman. *Machine Translation: A Knowledge-based Approach*. Morgan Kaufmann Publishers, Los Altos, CA, 1992.

[46] NIST. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. www.nist.gov/speech/tests/mt/doc/ngram-study.pdf, 2002.

[47] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. *Syntax for Statistical Machine Translation*. JHU Summer Workshop on Language Engineering, Center for Language and Speech Processing, Baltimore, MD, USA, 2003.

[48] F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*, pages 295–302, 2002.

[49] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[50] F. J. Och and H. Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, pages 417–449, 2004.

[51] Pan American Health Organization documents. http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat#H8, 2002.

[52] B. Pang, K. Knight, and D. Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of HLT/NAACL*, pages 102–209, 2003.

[53] K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, 2002.

[54] M. Porter. An algorithm for suffix stripping. *Automated Library and Information Systems*, 14(3):130–137, 1980.

[55] C. Quirk, C. Brockett, and W. Dolan. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, pages 142–149, 2004.

[56] Linux, Red Hat 7.2, English word list /usr/dict/words, 2002.

[57] T. Robertson, F. Wright, and R. Dykstra. *Order Restricted Statistical Inference*, chapter 1. John Wiley and Sons, 1988.

[58] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of ICML*, pages 441–448, 2001.

[59] R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000.

[60] J. Senellart, P. Dienes, and T. Váradi. Development of a new generation of translation system. In *MT Summit VIII*, 2001.

[61] R. Serafin and B. D. Eugenio. FLSA: Extending latent semantic analysis with features for dialogue act classification. In *Proceedings of ACL*, pages 692–699, 2004.

[62] L. Shen, A. Sarkar, and F. J. Och. Discriminative reranking for machine translation. In *Proceedings of HLT/NAACL*, pages 177–184, 2004.

[63] H. Somers. Review article: Example-based machine translation. *Machine Translation*, 14:113–157, 1999.

[64] A. Stent, M. Marge, and M. Singhai. Evaluating evaluation methods for generation in the presense of variation. In *Proceedings of CICLING*, pages 341–351, 2005.

[65] C. Tillmann. A projection extension algorithm for statistical machine translation. In *Proceedings of EMNLP*, pages 311–318, 2003.

[66] D. Wu. A polynomial-time algorithm for statistical machine translation. In *Proceedings of ACL*, pages 152–158, 1996.

[67] D. Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, pages 377–403, 1997.

[68] K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proceedings of ACL*, pages 523–530, 2001.

[69] K. Yamada and K. Knight. A decoder for syntax-based statistical MT. In *Proceedings of ACL*, pages 303–310, 2002.