ABSTRACT

Title of dissertation: THE CIRCLE OF MEANING: FROM TRANSLATION TO PARAPHRASING AND BACK

Nitin Madnani, Doctor of Philosophy, 2010

Dissertation directed by: Professor Bonnie Dorr
Department of Computer Science

The preservation of meaning between inputs and outputs is perhaps the most ambitious and, often, the most elusive goal of systems that attempt to process natural language. Nowhere is this goal of more obvious importance than for the tasks of machine translation and paraphrase generation. Preserving meaning between the input and the output is paramount for both, the monolingual vs bilingual distinction notwithstanding. In this thesis, I present a novel, symbiotic relationship between these two tasks that I term the "circle of meaning".

Today's statistical machine translation (SMT) systems require high quality human translations for parameter tuning, in addition to large bi-texts for learning the translation units. This parameter tuning usually involves generating translations at different points in the parameter space and obtaining feedback against human-authored reference translations as to how good the translations. This feedback then dictates what point in the parameter space should be explored next. To measure this feedback, it is generally considered wise to have multiple (usually 4) reference

translations to avoid unfair penalization of translation hypotheses which could easily happen given the large number of ways in which a sentence can be translated from one language to another. However, this reliance on multiple reference translations creates a problem since they are labor intensive and expensive to obtain. Therefore, most current MT datasets only contain a single reference. This leads to the problem of reference sparsity—the primary open problem that I address in this dissertation—one that has a serious effect on the SMT parameter tuning process.

Bannard and Callison-Burch (2005) were the first to provide a practical connection between phrase-based statistical machine translation and paraphrase generation. However, their technique is restricted to generating phrasal paraphrases. I build upon their approach and augment a phrasal paraphrase extractor into a sentential paraphraser with extremely broad coverage. The novelty in this augmentation lies in the further strengthening of the connection between statistical machine translation and paraphrase generation; whereas Bannard and Callison-Burch only relied on SMT machinery to extract phrasal paraphrase rules and stopped there, I take it a few steps further and build a full English-to-English SMT system. This system can, as expected, "translate" any English input sentence into a new English sentence with the same degree of meaning preservation that exists in a bilingual SMT system. In fact, being a state-of-the-art SMT system, it is able to generate n-best "translations" for any given input sentence. This sentential paraphraser, built almost entirely from existing SMT machinery, represents the first 180 degrees of the circle of meaning.

To complete the circle, I describe a novel connection in the other direction.

I claim that the sentential paraphraser, once built in this fashion, can provide a solution to the reference sparsity problem and, hence, be used to improve the performance a bilingual SMT system. I discuss two different instantiations of the sentential paraphraser and show several results that provide empirical validation for this connection.

THE CIRCLE OF MEANING: FROM TRANSLATION TO PARAPHRASING AND BACK

by

Nitin Madnani

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor Bonnie Dorr, Chair/Advisor
Professor Philip Resnik
Professor Chris Callison-Burch
Professor Lise Getoor
Professor William Idsardi

For Papa

Wish You Were Here

Acknowledgments

I owe a deep debt of gratitude to all the people who have made this thesis possible and whose help, advice and friendship made my graduate student career at Maryland a life experience that I will never forget.

Most graduate students hope to get a good advisor. I was lucky enough to be surrounded by advisors who wished nothing but the best for me and provided invaluable guidance. First and foremost, I want to thank my official advisor, Professor Bonnie Dorr, whose graduate class in 2004 was the first ever contact I had with the field of Natural Language Processing. That elective class I took as an MSEE student transformed my graduate life into one full of research challenges that I have really enjoyed solving. Bonnie has always had time for me whether it be for professional or personal reasons. I have learned so many things from her and not just about NLP. It has been an absolute pleasure to work with Bonnie. I could not have found a better advisor even if my entry into the field of NLP had not been a stroke of serendipity.

I would also like to thank Richard Schwartz of BBN Technologies who essentially served as my co-advisor for the last two and a half years. I really learned a lot from Rich about the challenges and opportunities that the field of statistical machine translation provides. It will not be an exaggeration to say that without his help and guidance, the work in this thesis could not have been completed. I also want to express my gratitude towards other BBN folks who taught me a lot: Antti, Spyros, Jacob, Jinxi and Libin.

Last but certainly not the least among all my advisors, I want to acknowledge Professor Philip Resnik. I came up with the fundamental idea underlying this thesis after one of those sessions with Philip where he served as a fountain of ideas and I, the listener, simply tried to gather all I could. Even though he was not required to advise me in any official capacity, Philip always took the time to listen to my ideas (some of them embarrassingly naive) and provided constructive feedback that only made them better.

I also want to thank all the other faculty members (Amy, Doug, Jimmy, Judith, Mary and Louiqa) associated with the CLIP lab at Maryland for all their help and guidance over the years. They were generous with them whenever I asked and also whenever I didn't but it was obvious to them that I could use some.

As a member of the NLP community, I have been extremely fortunate in managing to forge productive professional relationships with people outside Maryland. These relationships have proven to be extremely useful and instructive and I hope that they will continue to be so for the rest of my professional career. One such relationship that deserves explicit mention is the one I share with Chris Callison-Burch at Johns Hopkins University. CCB—as he is usually disambiguated in light of the inherent propensity for NLP research exhibited by people named Chris—is not just one of the nicest people that I have met in my professional life but also one of the smartest. I am glad to have him on my committee.

As a member of the CLIP lab, I have been extremely lucky to have worked with and, perhaps more importantly, hung out with some of the other smartest people on the planet. The list is long (Adam, Asad, ChrisD, Christof, David, Eric, Fazil,

Jordan, Matt, Michael, Okan, Saif, Yuval, Vlad) but incomplete as I am certain to have inadvertently left out several others. All these colleagues have willingly worn several hats for me over the years: bouncing boards for half-baked ideas, audience for countless practice talks, co-authors for papers, and above all, good friends.

Finally, I want to thank my family even though it is not really possible to do so in any words that I know. Mamma, Didi and Niroop have always been my pillars of strength and I am glad to have them in my life. Mulan Uncle, Padma Aunty, Reena and Ravi made me feel like I was part of their family and at home even though I was thousands of miles away from my real home. I hope I have made all of you proud and pray for your blessings and good wishes as I move into the next phase of my life.

Table of Contents

List of Figures

1 Introduction

———— * ————

I have in mind present day machines that do not possess a semantic organ. The situation will change in the not too distant future.

–Yehoshua Bar-Hillel (1953)

NLP System: *I haven't got a semantic organ ... only probabilistic straw.*
User: *How can you process language if you haven't got a semantic organ?*
NLP System: *I don't know... But some systems without semantic organs do an awful lot of processing... don't they?*
User: *Yes, I guess you're right.*

–adapted from `http://www.imdb.com/title/tt0032138/quotes?qt0409916`

———— * ————

The most ambitious, and often the most elusive, goal of systems that attempt to process natural language is the preservation of meaning between input and output. Nowhere is this goal of more obvious importance than in a machine translation system. Such a system must, ideally, "understand" the utterance in the source language in order to accurately translate it into a semantically equivalent and fluent utterance in the target language. Even in the years just after its inception, it was clear that the biggest obstacle to machine translation lay in meaning preservation which led to influential researchers calling for the explicit exhibition of semantic structure of input sentences as *sine qua non* for high quality machine translation (Bar-Hillel, 1970).

Meaning preservation is also paramount for the task of paraphrase generation. This task has not received as much focused attention in the community, especially at the sentence level, as the task of machine translation. The overarching goal of meaning preservation between the input and the output remains the same, even though they are now both in the same language. In fact, recent work has attempted to solidify this conceptual connection between translation and paraphrasing into a tangible one by generating paraphrases for English phrases using bilingual parallel text generally employed for translation (Bannard and Callison-Burch, 2005).

This thesis seeks to answer several questions that will serve to carve out a symbiotic relationship between the tasks of translation and paraphrase generation:

1. Is it possible to extend the existing work on paraphrase generation to the sentential level by, in fact, casting this problem as one of English-to-English translation?

2. How should this English-to-English translation model be constructed and defined in order to maximize meaning preservation?

3. Modern, state-of-the-art translation systems learn to translate from multiple **reference** translations for each input sentence. Given the expense of asking humans to create these translations, most new datasets only contain a single reference, leading to reference sparsity and, ultimately lower quality translation. Is it possible to create additional, artificial references by paraphrasing the single reference using the paraphraser built in (1) above and improve the translation quality?

Orig	Alcatel added that the company's whole year earnings would be announced on February 4.
Para	Alcatel said that the company's total annual revenues would be released on February 4.
Orig	He was now preparing a speech concerning the US policy for the upcoming World Economic Forum.
Para	He was now ready to talk with regard to the US policies for the forthcoming International Economic Forum.

Figure 1.1: Examples of paraphrases generated by the sentential paraphraser. **Orig** denotes the original sentence and **Para** its generated paraphrase. The sentences were chosen manually.

4. What characteristics should an artificial reference have in order for the translation system to learn as effectively as it might do with a human-authored reference translation?

Figure 1.1 shows example paraphrases that are generated by the English-to-English translation system. Figure 1.2 shows a Chinese sentence and two possible translations. The first was produced by a system that learned to translate by inspecting other Chinese sentences with one human reference translation. The second was produced by a system that learned by inspecting the same Chinese sentences but with both the human reference and its paraphrase as generated by my sentential paraphraser. For comparison, the figure also shows how a particular human translates the same Chinese sentence.

The sections below will provide additional motivation for the thesis problems described above in terms of two motifs —translation and paraphrase generation— and their symbiotic connection, followed by an outline of the thesis and specific research contributions.

Source	传北韩授权美国代表团走访宁边核子设施
Human	North Korea Reported to Allow US Delegation to Visit Nuclear Facilities in Yongbyon
Translation 1	Authorized by the North Korean delegation to visit the nuclear facilities in the United States
Translation 2	US delegation has been allowed to visit the Yongbyon nuclear facilities in North Korea

Figure 1.2: An example of how Chinese-English machine translation can be improved by the addressing the reference sparsity problem using the automatic sentential paraphraser. **Translation 1** is produced by a system that does not use the paraphrases while **Translation 2** is produced by the system that does. The sentences were chosen manually.

1.1 Motivation

Statistical Machine Translation—the approach assumed in this thesis and abbreviated as "Machine Translation" (or SMT)—relies heavily on machine-learning methods. This approach to automatic translation is distinguished by the heavy use of machine learning methods and has proven to be extremely successful. Almost all SMT techniques apply a learning algorithm to a large body of previously translated text, known as a *parallel corpus* or a *bitext*. It is assumed that the learner can generalize from these already translated examples and *learn* how to translate unseen sentences. Almost all modern SMT methods have been influenced, in one way or another, by the translation approach first proposed by Brown et al. (1990; 1993) at IBM. The *IBM Models*, as these models are referred to collectively, represent the earliest statistical translation models and operate at the word level. The next iteration of SMT methods were *phrase-based* in that they used models that were designed to translate contiguous sequences of words together as a unit (Marcu and Wong, 2002; Koehn et al., 2003).

Another concomitant change was the use of discriminative translation models,

where the posterior probability of the translation is directly modeled, instead of the earlier generative approach. Using phrases allowed learning of local reorderings, translations of multi-word expressions, or insertions and deletions that are sensitive to local context. However, it was observed by Koehn et al. (2003) that phrases longer than three words do not give any significant gains in translation performance for training with bitexts as large 20 million words. Even though this finding did not pass the test of time, it was felt that the data may generally be too sparse to learn longer fully-lexicalized phrases. In addition, the challenge of finding a good reordering of the translated phrases still remained. Chiang (2007) proposed a solution to this problem that built directly upon the strengths of the phrase-based approach: since phrases are good for learning reorderings of words, they can be used to learn reorderings of phrases as well. This was achieved by the use of *hierarchical phrases*, i.e., a phrase that contains placeholders that can be filled by other phrases. Hierarchical-phrase based models represent the current state-of-the-art. For a more comprehensive survey of contemporary SMT, the reader is referred to (Lopez, 2008) and (Koehn, 2010).

Over the years, SMT techniques have made good progress towards the goal of creating semantically equivalent translations. They have done so by learning a generalized model of linguistic correspondence between the source and target languages from millions of actual meaning-preserving human translation examples. In fact, as the translation models have become more complex, the reliance on such examples has increased. In addition to the large bitexts that are used to extract lexical, phrasal and hierarchical correspondence units, today's SMT systems also

require *additional* high quality human translations for parameter tuning. Such tuning is usually carried out by means of machine learning algorithms that intelligently traverse the very large multi-dimensional parameter space (Ostendorf et al., 1991; Och, 2003). Translations are produced for each explored point in the parameter space and a quantitative estimate of the error in said translations is computed by comparing them against human *reference* translations. The magnitude of this error is then used to guide the search to a more useful point in the parameter space.

Since a given source sentence can be translated into the target language in a multitude of ways, it is generally considered wise to have multiple (usually four) reference translations; with only a single reference, it is possible that an entirely correct translation hypothesis may be judged incorrect due to low n-gram overlaps and, therefore, the feedback to the search process may be misguided. However, this reliance on multiple reference translations creates a problem, because reference translations are labor intensive and expensive to obtain. For example, producing reference translations at the Linguistic Data Consortium, a common source of translated data for MT research, requires undertaking an elaborate process that involves translation agencies, detailed translation guidelines, and quality control processes (Strassel et al., 2006). Therefore, most recent datasets produced for use in SMT tend to contain only a single reference translation. This leads to the problem of *reference sparsity*—the primary open problem that I address in this thesis—one that has a serious effect on the SMT parameter tuning process. This thesis posits the first direct *paraphrase-based* solution to this problem.

Everyone is familiar with the notion of a *paraphrase* in its most fundamental

sense. The concept of *paraphrasing* is most generally defined by the principle of semantic equivalence, i.e., a paraphrase is an alternative surface form in the *same language* expressing the same semantic content as the original form. While the task of automatically generating lexical, phrasal and sentential paraphrases has been employed to improve the performance of several NLP applications, it has not historically received as much focused attention as machine translation has. However, the relationship between these two tasks is quite strong as obvious from the above definition; paraphrase generation can simply be seen as a monolingual version of machine translation.

Resnik (2004) presents a detailed treatment of the general idea that monolingual semantic knowledge can be considered inherent in parallel bilingual corpora, i.e., looking at two languages in parallel translation provides a way to "triangulate" on semantics without having to commit to overt semantic representations. The relationship between paraphrase generation and machine translation can be considered to be an instantiation of this idea. Bannard and Callison-Burch (2005) were the first to provide a practical connection between phrase-based statistical machine translation and paraphrase generation. However, their technique was restricted to generating phrasal paraphrases. This thesis builds upon their approach and augments a phrasal paraphrase extractor into a sentential paraphraser with extremely broad coverage. The novelty in this augmentation lies in the further strengthening of the connection between statistical machine translation and paraphrase generation. Whereas Bannard and Callison-Burch (2005) only rely on SMT machinery to extract phrasal paraphrase rules and stop there, this thesis takes the additional step

of building a full English-to-English SMT system. This system can, as expected, "translate" any English input *sentence* into a new English sentence with the same degree of meaning preservation that exists in a bilingual SMT system. In fact, as a state-of-the-art SMT system, it is able to generate n-best "translations" for any given input sentence.

The description above characterizes a novel one-sided connection between translation and paraphrasing: using statistical machine translation as the basis for constructing a *sentential* paraphraser. To complete the circle of meaning, this thesis also proposes a new connection in the other direction, claiming that the sentential paraphraser, once built in this fashion, can provide a solution to the reference sparsity problem and, hence, be used to improve the performance a bilingual SMT system. The solution is simply to create multiple artificial references by paraphrasing the available single human reference with the sentential paraphraser. Two different instantiations of the sentential paraphraser are described. Both automatic translation metrics and human judgments are used to empirically validate this proposed connection. Therefore, the tasks of machine translation and sentential paraphrase generation can now be thought of as participating in a symbiotic relationship. Any developments that improve the core SMT machinery will also help in building an improved sentential paraphraser. An improved paraphraser can, in turn, help an SMT system better overcome the reference sparsity problem.

1.2 Outline of the Dissertation

Setting the stage for the research reported in this thesis, Chapter 2 surveys the current state of the art in corpus-based paraphrase generation techniques, sets the current work in context and provides the first up-to-date and comprehensive overview of the field. Over the last two decades, there has been significant research on paraphrase generation within every NLP research community so as to improve the specific application with which that community is concerned. This has led to a fragmented research pattern: paraphrase generation is used in different forms and with different names in the context of different applications (e.g., synonymous collocation extraction, query expansion). This usage pattern does not allow researchers in one community to share the lessons learned with those from other communities. Chapter 2 brings together research on paraphrase generation from these different sub-communities and draws broad and useful connections among researchers working on related problems. Subsequent chapters rely on this chapter as a foundation.

Chapter 3 describes the first 180 degrees of the circle of meaning: how to build a sentential paraphraser using nothing but widely available SMT machinery. The architecture of such a paraphraser is described along with examples and empirical evaluations of its output.

In Chapter 4, a first attempt at completing the circle is described, i.e., addressing the reference sparsity problem for an SMT system by using the sentential paraphraser presented in Chapter 3. Both automatic and manual evaluation results are presented for translations produced by this SMT system for four different source

languages. The results show that while using artificial references does lead to significant improvements in translation performance, the improvements diminish as more and more paraphrases are used. An error analysis is provided to explain why this is the case.

Chapter 5 presents the creation of a new version of the sentential paraphraser that approaches the reference sparsity problem from a different angle. This paraphraser makes very focused, targeted changes to the original reference in contrast to the paraphraser presented in Chapter 3. Automatic and manual evaluation results are provided for the same experimental conditions as those used in Chapter 4 and this paraphraser is also shown to produce significant gains when used for tuning. Furthermore, the gains are shown to increase monotonically as more paraphrases are added.

Chapter 6 takes an overarching view of the characteristics of reference translations insofar as they are related to the SMT parameter tuning process. Three types of reference translations are compared: human references, untargeted paraphrases of human references computed using the approach described in Chapter 4, and targeted paraphrases of human references computed using the approach described in Chapter 5. The comparison is undertaken in terms of three qualities associated with reference translations that are essential for effective parameter tuning: correctness, reachability, and focus. An approach to measuring the degree to which a set of reference translations possess these qualities is presented. Such a detailed characterization of reference translations has not previously been undertaken. This chapter then presents possible avenues of future work and concludes with a reiteration of

the questions posed above along with the answers that this dissertation provides.

Appendix A compares the actual translations produced by the various SMT systems used in this dissertation and provides an intuitive picture of the empirical gains in translations quality reported in Chapters 4 and 5.

1.3 Research Contributions

Through the research conducted in this thesis, I have made the following research contributions:

- A new general sentential paraphraser architecture is developed, in the form of an English-to-English SMT system. It is built entirely using bilingual SMT machinery and by extending previous research on phrasal paraphrase generation. Some components of the architecture are simply adaptations of the corresponding bilingual components whereas others are entirely novel.

- The first (ever) automatic approach to addressing the reference sparsity problem in SMT is implemented in the form of a sentential paraphraser that creates artificial references. The solution is elegant in that it is entirely bootstrapped using almost nothing except what is already available to the bilingual SMT system.

- A second SMT-specific instantiation of the sentential paraphraser is developed, wherein paraphrases are generated in a focused, *targeted* fashion. This second instantiation is driven by two new additions —a targeting feature and a self-paraphrase bias–that interact with each other. A novel search method has

been implemented to determine the configuration of these additions that is most effective for the paraphrasing process.

- Both instantiations of the sentential paraphraser, when used to create artificial references for parameter tuning, are shown to induce statistically significant gains as measured by automatic MT evaluation metrics (Madnani et al., 2007, 2008a,b). In addition, the same gains are demonstrated when Amazon Mechanical Turk is used to enlist human subjects to evaluate the translation output.

- The first (ever) detailed characterization of a reference translation is undertaken in terms of three essential qualities: correctness, reachability, and focus. A theoretical and empirical analysis is provided wherein both human and artificial references are compared according to these qualities.

- A comprehensive overview of paraphrasing approaches and their application is presented, wherein the hitherto fragmented research on data-driven paraphrase generation has been brought together and broader philosophical connections among related efforts has been drawn (Madnani and Dorr, 2010).

It has been well understood that the tasks of machine translation and paraphrase generation are intimately connected and existing research has taken promising preliminary steps towards substantiating this connection, e.g., by showing that it is possible to induce monolingual semantics at the phrase level from bilingual parallel text. However, the work in this thesis takes this connection much farther by establishing a fully symbiotic relationship between translation and paraphrase

generation. As part of carving out this relationship, my work yields two novel and important research contributions: (1) a general sentential paraphrasing architecture, modeled as English-to-English translation and built entirely from components usually employed to build bilingual translation systems and, (2) a *direct* paraphrase-driven solution to the reference sparsity problem faced by a state-of-the-art machine translation; one that is able to provide statistically significant improvements in the quality of produced translations.

2 Related Work on Data-Driven Paraphrase Generation and its Applications

———— * ————

Knowledge is out there.
Among bits, chunks and pieces.
It must be gathered.
–Nitin Madnani

———— * ————

While everyone may be familiar with the notion of *paraphrase* in its most fundamental sense, there is still room for elaboration on how paraphrases may be automatically generated or elicited for use in language processing applications. Moreover, the task of automatically generating or extracting semantic equivalences for the various units of language—words, phrases and sentences, is being increasingly employed to improve the performance of several NLP applications. This chapter presents a comprehensive and application-independent overview of data-driven phrasal and sentential paraphrase generation methods is presented, while also conveying an appreciation for the importance and potential use of paraphrases in the field of NLP research. While many paraphrase methods are developed with a particular application in mind, all methods share the potential for more general applicability. Recent work on manual and automatic construction of paraphrase corpora is presented,

strategies used for evaluating paraphrase generation techniques are discussed, and future trends in paraphrase generation are explored.[1]

Related work on furthering the community's understanding of paraphrases has been done by Hirst (2003), wherein a deep analysis of the nature of paraphrase is provided. This chapter focuses instead on delineating the salient characteristics of the various paraphrase generation methods with an emphasis on describing how they could be used in several different NLP applications. Both these treatments provide different but valuable perspectives on paraphrasing. The next section formalizes the concept of a paraphrase and scopes out the coverage of the discussion for the remainder of this chapter. Section 2.2 provides broader context and motivation by discussing applications in which paraphrase generation has proven useful, along with examples. Section 2.3 briefly describes the tasks of paraphrase recognition and textual entailment and their relationship with paraphrase generation and extraction. Section 2.4 is the core of this chapter, wherein various corpus-based techniques for paraphrase generation are examined, organized by corpus type. Section 2.5 examines recent work done to construct various types of paraphrase corpora and to elicit human judgments for such corpora. Section 2.6 considers the task of evaluating the performance of paraphrase generation and extraction techniques. Finally, Section 2.7 provides a brief glimpse of the future trends in paraphrase generation and Section 2.8 concludes the chapter with a summary.

[1]The material presented in this chapter will be published as an upcoming journal article (Madnani and Dorr, 2010).

2.1 Formalization of Paraphrase and Scope of Discussion

The concept of *paraphrasing* is most generally defined on the basis of the principal of semantic equivalence, i.e., a paraphrase is an alternative surface form in the same language expressing the same semantic content as the original form. Paraphrases may occur at several levels as itemized below:

- **Lexical**: Individual lexical items having the same meaning are usually referred to as *lexical paraphrases* or, more commonly, *synonyms*, e.g., $\langle$*hot*, *warm*$\rangle$ and $\langle$*eat*, *feed*$\rangle$. However, lexical paraphrasing cannot be restricted strictly to the concept of synonymy. There are several other forms such as *hyperonyms*, where one of the words in the paraphrastic relationship is either more general or more specific than the other, e.g. $\langle$*reply*, *say*$\rangle$ and $\langle$*landlady*, *hostess*$\rangle$.

- **Phrasal**: The term *phrasal paraphrases* refers to phrasal fragments sharing the same semantic content. While these fragments most commonly take the form of syntactic phrases, e.g. $\langle$*work on*, *soften up*$\rangle$ and $\langle$*take over*, *assume control of*$\rangle$, they may also be patterns with linked variables, e.g. $\langle$*Y was built by X*, *X is the creator of Y*$\rangle$.

- **Sentential**: Two sentences that represent the same semantic content are termed *sentential paraphrases*. For example, $\langle$*I finished my work*, *I completed my assignment*$\rangle$. While it is possible to generate very simple sentential paraphrases by simply substituting words and phrases in the original sentence with their respective semantic equivalents, it is significantly more difficult to gen-

erate more interesting ones, e.g. ⟨*He needed to make a quick decision in that situation*, *The scenario required him to make a split-second judgment*⟩.

The idea of paraphrasing has been explored in conjunction with, and employed in, a large number of natural language processing applications. Given the difficulty inherent in surveying such a diverse topic, an unfortunate but necessary remedy is to impose certain limits on the scope of the discussion. In this chapter, and in the remainder of the thesis, the discussion will be restricted to automatic acquisition of phrasal paraphrases (including paraphrastic patterns) and on generation of sentential paraphrases. More specifically, this entails the exclusion of certain categories of paraphrasing work. However, as a compromise for the interested reader, a relatively comprehensive list of references is included for the work that this chapter does not cover.

For one, paraphrasing techniques that rely primarily on knowledge-based resources are not discussed, e.g., those that rely on dictionaries (Wallis, 1993; Fujita et al., 2004), hand-written rules (Fujita et al., 2007) and formal grammars (McKeown, 1979; Dras, 1999; Gardent et al., 2004; Gardent and Kow, 2005). Work on purely lexical paraphrasing is not included, e.g., approaches that make use of various ways to cluster words occurring in similar contexts (Inoue, 1991; Crouch and Yang, 1992; Pereira et al., 1993; Grefenstette, 1994; Lin, 1998; Gasperin et al., 2001; Glickman and Dagan, 2003; Shimohata and Sumita, 2005).[2] Exclusion of general

[2]Inferring words to be similar based on similar contexts can be thought of as the most common instance of employing *distributional similarity*. The concept of distributional similarity also turns out to be quite important for phrasal paraphrase generation and is discussed in more detail in Section 2.4.1.

lexical paraphrasing methods obviously implies that other lexical methods developed just for specific applications are also excluded (Bangalore and Rambow, 2000; Duclaye et al., 2003; Murakami and Nasukawa, 2004; Kauchak and Barzilay, 2006). Methods at the other end of the spectrum that paraphrase supra-sentential units such as paragraphs and entire documents are also omitted from discussion (Hovy, 1988; Inui and Nogami, 2001; Power and Scott, 2005; Hallett and Scott, 2005).

Finally, the notion of near-synonymy is also not discussed in detail. Near-synonyms are words that are almost synonyms, but not quite (Hirst, 1995; Edmonds and Hirst, 2002). They are not fully substitutable for each other, but vary in their shades of connotation, or in the components of semantic emphasis and grammatical or collocational constraints. For example, the word *foe* emphasizes active warfare more than *enemy* does and *forest* and *woods* differ in terms of a complex combination of size, proximity to civilization, and wildness. Some applications using lexical paraphrases (synonyms), e.g. query expansion, may be agnostic on the distinction between synonyms and near-synonyms, yet this distinction is arguably central to understanding the true nature of paraphrases. More recently, work has focused on automatic elicitation of near-synonyms for database population (Inkpen and Hirst, 2006; Inkpen, 2007).

2.2 Applications of Paraphrase Generation

Before describing the techniques used for paraphrasing, it is essential to examine the broader context of the application of paraphrases. For some of the applica-

tions discussed below, the use of paraphrases in the manner described may not yet be the norm. However, wherever applicable, recent research is cited that promises gains in performance by using paraphrases for these applications. Also note that only those paraphrasing techniques are discussed that can generate the types of paraphrases examined in this chapter: phrasal and sentential.

2.2.1 Query and Pattern Expansion

One of the most common applications of paraphrasing is automatic generation of query variants for submission to information retrieval systems or of patterns for submission to information extraction systems. For example, one of the earliest approaches (Spärck-Jones and Tait, 1984) generated several simple variants for compound nouns, in queries submitted to a technical information retrieval system.

Original : *circuit details*

Variant 1 : *details about the circuit*

Variant 2 : *the details of circuits*

In fact, in recent years, the information retrieval community has extensively explored the task of query expansion by applying paraphrasing techniques to generate similar or related queries (Beeferman and Berger, 2000; Jones et al., 2006; Sahami and Heilman, 2006; Shi and Yang, 2007; Metzler et al., 2007). The generation of paraphrases in these techniques is usually effected by utilizing the *query log* (a log containing the record of all queries submitted to the system) to determine semantic

similarity.

Jacquemin (1999) generates morphological, syntactic as well as semantic variants for phrases in the agricultural domain. For example,

Original : *simultaneous measurements*

Variant : *concurrent measures*

Original : *development area*

Variant : *area of growth*

Ravichandran and Hovy (2002) use semi-supervised learning to induce several paraphrastic patterns for each question type and use them in an open-domain question answering system. For example, for the INVENTOR question type, they generate:

Original : *X was invented by Y*

Variant 1 : *Y 's invention of X*

Variant 2 : *Y , inventor of X*

Riezler et al. (2007) expand a query by generating n-best paraphrases for the same (via a pivot-based sentential paraphrasing model employing bilingual parallel corpora, detailed in Section 2.4) and then using any new words introduced therein as additional query terms. For example, for the query *how to live with cat allergies*, they may generate the following two paraphrases. The novel words in the two

paraphrases are highlighted in bold and are used to expand the original query:

P_1 : **ways** *to live with* **feline allergy**

P_2 : *how to* **deal** *with cat* **allergens**

Finally, paraphrases have also been used to improve the task of relation extraction (Romano et al., 2006). Most recently, Bhagat and Ravichandran (2008) collect paraphrastic patterns for relation extraction by applying semi-supervised paraphrase induction to a very large monolingual corpus. For example, for the relation of "acquisition," they collect:

Original : *X agreed to buy Y*

Variant 2 : *X completed its acquisition of Y*

Variant 3 : *X purchased Y*

2.2.2 Expanding Sparse Human Reference Data for Evaluation

A large percentage of NLP applications are evaluated by having human annotators or subjects carry out the same task for a given set of data and using the output so created as a reference against which to measure the performance of the system. The two applications where comparison against human-authored reference output has become the norm are machine translation and document summarization.

In machine translation evaluation, the translation hypotheses output by a

machine translation system are evaluated against reference translations created by human translators by measuring the n-gram overlap between the two (Papineni et al., 2002). However, it is impossible for a single reference translation to capture all possible verbalizations that can convey the same semantic content. This may unfairly penalize translation hypotheses that have the same meaning but use n-grams that are not present in the reference. For example, the system output S below will not have a high score against the reference R even though it conveys precisely the same semantic content:

S : *We must consider the entire community.*

R : *We must bear in mind the community as a whole.*

One solution is to use multiple reference translations which is expensive. An alternative solution, tried in a number of recent approaches, is to address this issue by allowing the evaluation process to take into account paraphrases of phrases in the reference translation so as to award credit to parts of the translation hypothesis that are semantically, even if not lexically, correct (Zhou et al., 2006a; Owczarzak et al., 2006).

Another way to address the lack of multiple references is to use evaluation metrics that possess an inherent notion of semantic equivalence, such as METEOR (Lavie and Agarwal, 2007) which employs WordNet (Fellbaum, 1998) or TERp (Snover et al., 2009) which uses phrasal paraphrases induced in the manner described in Section 2.4.5. While the previous versions of METEOR only used lexical para-

phrases from WordNet, the current version (Denkowski and Lavie, 2010) is able to take advantage of the same set of automatically induced paraphrases that is used by TERp.

In evaluation of document summarization, automatically generated summaries (*peers*) are also evaluated against reference summaries created by human authors (*models*). Zhou et al. (2006b) propose a new metric called ParaEval that leverages an automatically extracted database of phrasal paraphrases to inform the computation of n-gram overlap between peer summaries and multiple model summaries.

2.2.3 Machine Translation

Besides being used in evaluation of machine translation systems, paraphrasing has also been applied to directly improve the translation process. Callison-Burch et al. (2006b); Marton et al. (2009) use automatically induced paraphrases to improve a statistical phrase-based machine translation system. Such a system works by dividing the given sentence into phrases and translating each phrase individually by looking up its translation from a table. The coverage of the translation system is improved by allowing any source phrase that does not have a translation in the table to use the translation of one of its paraphrases. For example, if a given Spanish sentence contains the phrase *presidente de Brazil* but the system does not have a translation for it, another Spanish phrase such as *presidente brasileño* may be automatically detected as a paraphrase of *presidente de Brazil*; then if the translation table contains a translation for the paraphrase, the system can use the same trans-

lation for the given phrase. Therefore, paraphrasing allows the translation system to properly handle phrases that it does not otherwise know how to translate.

In a similar vein, Buzek et al. (2010) automatically identify portions of the source sentence that are likely to be problematic for MT systems. They then elicit paraphrases for these portions using Amazon Mechanical Turk and. Finally, all new sentences (formed by replacing each specific portion with its paraphrase and the combinations thereof) are translated in addition to the original sentence. Results show that for about a sixth of the input sentences, the MT system is able to produce better translations for one of the paraphrastic source sentences compared to the original source. Although the percentage of sentences affected is small, the ones that do benefit do so substantially. Further research will likely lead to even larger performance gains.

As mentioned in Chapter 1, another important issue for statistical machine translation systems is that of *reference sparsity*; one that is the primary problem that the work in this dissertation tries to solve. Chapters 4 and 5 will describe in detail how automatic sentential paraphrases can be used, very effectively, to alleviate the reference sparsity problem affecting statistical machine translation systems.

2.3 Paraphrase Recognition and Textual Entailment

A problem closely related to, and as important as, generating paraphrases is one of assigning a quantitative measurement to the semantic similarity between two phrases (Fujita and Sato, 2008b) or even two given pieces of text (Corley and

Mihalcea, 2005; Uzuner and Katz, 2005). A more complex formulation of the task would be to detect or recognize which sentences in the two texts are paraphrases of each other (Brockett and Dolan, 2005; Wu, 2005; Marsi and Krahmer, 2005b; João et al., 2007a,b; Das and Smith, 2009). Both of these task formulations fall under the category of paraphrase detection or recognition. The latter formulation of the task has become popular in recent years (Dolan and Dagan, 2005) and paraphrase generation techniques that require monolingual parallel or comparable corpora (discussed in Section 2.4) can benefit immensely from this task. In general, paraphrase recognition can be very helpful for several NLP applications. Two examples of such applications are text-to-text generation and information extraction.

Text-to-text generation applications rely heavily on paraphrase recognition. For a multi-document summarization system, detecting redundancy is a very important concern because two sentences from different documents may convey the same semantic content and it is important not to repeat the same information in the summary. On this note, Barzilay and McKeown (2005) exploit redundancy present in a given set of sentences by detecting paraphrastic parts and fusing them into a single coherent sentence. Recognizing similar semantic content is also critical for text simplification systems (Marsi and Krahmer, 2005a).

Information extraction enables the detection of regularities of information structure—events which are reported many times, about different individuals and in different forms—and making them explicit so that they can be processed and used in other ways. Sekine (2006) shows how to use paraphrase recognition to cluster together extraction patterns to improve the cohesion of the extracted information.

Another recently proposed natural language processing task is that of recognizing textual entailment: a piece of text T is said to entail a hypothesis H if humans reading T will infer that H is most likely true. The observant reader will notice that, in this sense, the task of paraphrase recognition can simply be formulated as bidirectional entailment recognition. The task of recognizing entailment is an application-independent task and has important ramifications for almost all other language processing tasks that can derive benefit from some form of applied semantic inference. For this reason, the task has received noticeable attention in the research community and annual community-wide evaluations of entailment systems have been held in the form of *Recognizing Textual Entailment* (RTE) Challenges (Dagan et al., 2006; Bar-Haim et al., 2007; Sekine et al., 2007; Giampiccolo et al., 2008).

Looking towards the future, Dagan (2008) proposes that the textual entailment task provides a comprehensive framework for semantic inference and argues for building a concrete inference engine that not only recognizes entailment but also searches for all entailing texts given an entailment hypothesis H and, conversely, generates all entailed statements given a text T. Given such an engine, Dagan claims that paraphrase generation is simply a matter of generating all entailed statements given any sentence. While this is a very attractive proposition that defines both paraphrase generation and recognition in terms of textual entailment, there are some important caveats. For example, textual entailment cannot guarantee that the entailed hypothesis H captures all of the same meaning as the given text T. Consider the following example:

T: *Yahoo's buyout of Overture was finalized.*

H_1: *Yahoo bought Overture.*

H_2: *Overture is now owned by Yahoo.*

While both H_1 and H_2 are entailed by T, they are not strictly paraphrases of T since some of the semantic content has not carried over. This must be an important consideration when building the proposed entailment engine. Of course, even these approximately semantically equivalent constructions may prove useful in an appropriate downstream application.

Of course, the relationship between paraphrasing and entailment is more tightly entwined than it might appear. Entailment recognition systems sometimes rely on the using paraphrastic templates or patterns as inputs (Iftene, 2009) and might even use paraphrase recognition to improve their performance (Bosma and Callison-Burch, 2007). In fact, examination of some RTE datasets in an attempt to quantitatively determine the presence of paraphrases has shown that a large percentage of the set consists of paraphrases rather than typical entailments (Bayer et al., 2005; Garoufi, 2007). It has also been observed that, in the entailment challenges, it is relatively easy for submitted systems to recognize constructions that partially overlap in meaning (approximately paraphrastic) than those that are actually bound by an entailment relation. On the flip side, work has also been done to extend entailment recognition techniques for the purpose of paraphrase recognition (Rus et al., 2008).

Detection of semantic similarity and, to some extent, that of bidirectional entailment are usually an implicit part of paraphrase generation. However, given the

interesting and diverse work that has been done in both these areas, any significant discussion beyond the treatment above merits a separate, detailed survey.

2.4 Paraphrasing with Corpora

This section explores in detail the data-driven paraphrase generation approaches that have emerged and become extremely popular in the last decade or so. These corpus-based methods have the potential of covering a much wider range of paraphrasing phenomena and the advantage of widespread availability of corpora.

This section is organized by the type of corpora used to generate the paraphrases: a single monolingual corpus, monolingual comparable corpora, monolingual parallel corpora and bilingual parallel corpora. This form of is the most instructive since most of the algorithmic decisions made for paraphrase generation will depend heavily on the type of corpus used. For instance, it is reasonable to assume that a different set of considerations will be paramount when using a large single monolingual corpus than when using bilingual parallel corpora.

However, before delving into the actual paraphrasing methods, it is useful to explore the motivation behind *distributional similarity*, an extremely popular technique that can be used for paraphrase generation with several different types of corpora. The following section is dedicated to such an explanation.

2.4.1 Distributional Similarity

The idea that a language possesses *distributional structure* was first discussed at length by Harris (1954). The term represents the notion that one can describe a language in terms of relationships between the occurrences of its elements (words, morphemes, phonemes) relative to the occurrence of other elements. The name for the phenomenon is derived from an element's *distribution*—sets of elements in particular positions that the element occurs with to produce an utterance or a sentence.

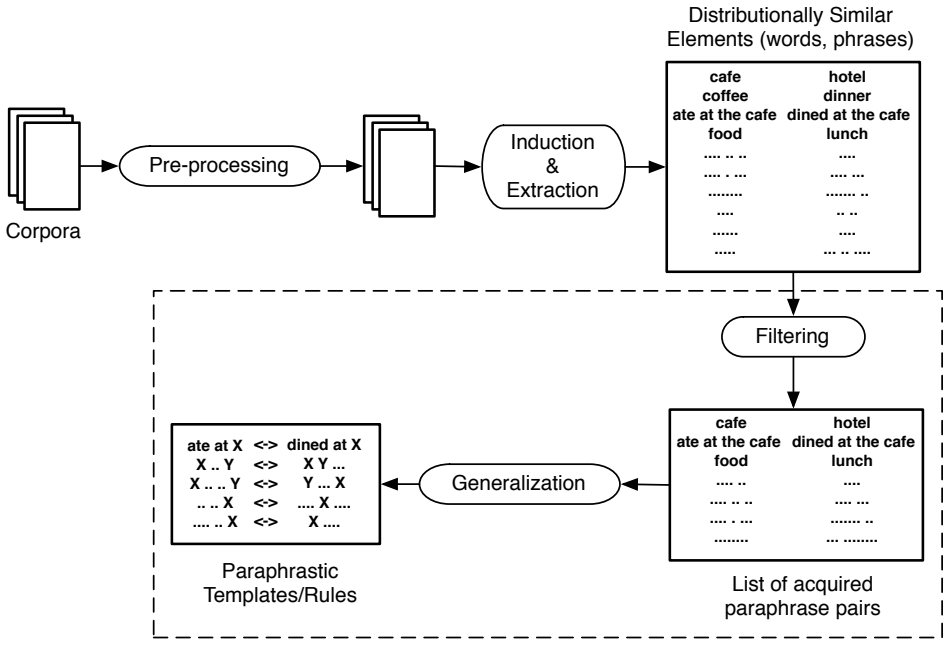


Figure 2.1: A general architecture for paraphrasing approaches leveraging the distributional similarity hypothesis.

More specifically, Harris presents several empirical observations to support the hypothesis that such a structure exists naturally for language. These observations are closely quoted here:

- Utterances and sentences are not produced by arbitrarily putting together the elements of the language. In fact, these elements usually occur only in certain positions relative to certain other elements.

- The empirical restrictions on the co-occurrents of a class are respected for each and every one of its members and are not disregarded for arbitrary reasons.

- The occurrence of a member of a class relative to another member of a different class can be computed as a probabilistic measure, defined in terms of the frequency of that occurrence in some sample or corpus.

- Not every member of every class can occur with every member of another class (think nouns and adjectives). This observation can be used as a measure of difference in meaning. For example, if the pair of words *teacher* and *instructor* is considered to be more semantically equivalent than say, the pair *teacher* and *musician*, then even the distributions of the first pair will be more alike than that of the latter.

Given the above observations, it's relatively easy to perceive the concept of *distributional similarity*—words or phrases that share the same *distribution*, i.e., the same set of words in the same context in a corpus, tend to have similar meanings.

Figure 2.1 shows the basic idea behind phrasal paraphrase generation techniques that leverage distributional similarity. The input corpus is usually a single or set of monolingual corpora (parallel or non-parallel). After preprocessing—which may include tagging the parts of speech, generating parse trees and other transformations—the next step is to extract pairs of words or phrases (or patterns)

that occur in the same context in the corpora and hence may be considered (approximately) semantically equivalent. This extraction may be accomplished by several means, e.g., by using a classifier employing contextual features or by finding similar paths in dependency trees. While it is possible to stop at this point and consider this list as the final output, the list usually contains a lot of noise and may require additional filtering based on other criteria, such as collocations counts from another corpus (or the Web). Finally, some techniques may go even further and attempt to generalize the filtered list of paraphrase pairs into templates or rules which may then be applied to other sentences to generate their paraphrases. Note that generalization as a post-processing step may not be necessary if the induction process can extract distributionally similar patterns directly.

One potential disadvantage of relying on distributional similarity is that items that are distributionally similar may not necessarily end up being paraphrastic: both elements of the pairs $\langle$*boys, girls*$\rangle$, $\langle$*cats, dogs*$\rangle$, $\langle$*high, low*$\rangle$ can occur in similar contexts but are not semantically equivalent.

2.4.2 Paraphrasing using a Single Monolingual Corpus

This section concentrates on paraphrase generation methods that operate on a single monolingual corpus. Most, if not all, such methods usually perform paraphrase induction by employing the idea distributional similarity as outlined in the previous section. Besides the obvious caveat discussed above regarding distributional similarity, the other most important factor affecting the performance of these

methods is the choice of distributional ingredients, i.e., the features used to formulate the distribution of the extracted units. Three commonly used techniques are considered that generate phrasal paraphrases (or paraphrastic patterns) from a single monolingual corpus but use very different distributional features in terms of complexity. The first uses only surface-level features while the other two use features derived from additional semantic knowledge. While the latter two methods are able to generate more sophisticated paraphrases by virtue of more specific and more informative ingredients, doing so usually has an adverse effect on their coverage.

Paşca and Dienes (2005) use as their input corpus a very large collection of Web documents taken from the repository of documents crawled by Google. While using Web documents as input data does require a non-trivial pre-processing phase since such documents tend to be noisier, there are certainly advantages to using Web documents as the input corpus: it does not need to have parallel (or even comparable) documents and can allow leveraging of even larger document collections. In addition, the extracted paraphrases are not tied to any specific domain and are suitable for general application.

Algorithm 1 shows the details of the induction process. Steps 3-6 extract all n-grams of a specific kind from each sentence: each n-gram has L_c words at the beginning, between M_1 to M_2 words in the middle and another L_c words at the end. Steps 7-13 can intuitively be interpreted as constructing a textual anchor A—by concatenating a fixed number of words from the left and the right—for each candidate paraphrase C and storing the (anchor, candidate) tuple in H. These anchors are taken to constitute the distribution of the words and phrases under

inspection. Finally, each occurrence of a pair of potential paraphrases, i.e. a pair sharing one or more anchors, is counted. The final set of phrasal paraphrastic pairs is returned.

Algorithm 1 (Paşca and Dienes 2005). Inducing a set of phrasal paraphrase pairs H with associated counts from a corpus of C pre-processed Web documents.
Summary. Extract all n-grams from C longer than a pre-stipulated length. Compute a *lexical anchor* for each extracted n-gram. Pairs of n-grams that share lexical anchors are then construed to be paraphrases.

Let N represent a set of n-grams extracted from the corpus
$N \leftarrow \{\phi\}$, $H \leftarrow \{\phi\}$
for each sentence **E** in the corpus **do**
    Extract the set of n-grams $N_E = \{\bar{e}_i \; s.t \; (2L_c + M_1) \leq |\bar{e}_i| \leq (2L_c + M_2)\}\}$, where M_1, M_2 and L_c are all preset constants and $M_1 \leq M_2$
    $N \leftarrow N \cup N_E$
end for
for each n-gram $\bar{e}$ in N **do**
    Extract the subsequence C, such that $L_c \leq |C| \leq (|\bar{e}| - L_c - 1)$
    Extract the subsequence A_L, such that $0 \leq |A_L| \leq (L_c - 1)$
    Extract the subsequence A_R, such that $(|\bar{e}| - L_c) \leq |A_R| \leq (|\bar{e}| - 1)$
    $A \leftarrow A_L + A_R$
    Add the pair (A, C) to H
end for
for each subset of H with the same anchor A **do**
    Exhaustively compare each pair of tuples (A, C_i) and (A, C_j) in this subset
    Update the count of the candidate paraphrase pair (C_i, C_j) by 1
    Update the count of the candidate paraphrase pair (C_j, C_i) by 1
end for
Output H containing paraphrastic pairs and their respective counts

---

This algorithm embodies the spirit of the hypothesis of distributional similarity: it considers all words and phrases that are distributionally similar, i.e., they occur with the same sets of anchors (or distributions), to be paraphrases of each other. Additionally, the larger the set of shared anchors for two candidate phrases, the stronger the likelihood that they are paraphrases of each other. After extracting the list of paraphrases, less likely phrasal paraphrases are filtered out by using an

appropriate count threshold.

Paşca and Dienes (2005) attempt to make their anchors even more informative by attempting variants where they extract the n-grams only from sentences that include specific additional information to be added to the anchor. For example, in one variant, they only use sentences where the candidate phrase is surrounded by named entities on both sides and they attach the nearest pair of entities to the anchor. As expected, the paraphrases do improve in quality as the anchors become more specific. However, they also report that as anchors are made more specific by attaching additional information, the likelihood of finding a candidate pair with the same anchor is reduced.

The ingredients for measuring distributional similarity in a single corpus can certainly be more complex than simple phrases used by Paşca and Dienes. Lin and Pantel (2001) discuss how to measure distributional similarity over dependency tree paths in order to induce generalized paraphrase templates such as:[3]

X found answer to Y $\Leftrightarrow$ *X solved Y*

X caused Y $\Leftrightarrow$ *Y is blamed on X*

While single links between nodes in a dependency tree represent direct semantic relationships, a sequence of links, or a *path*, can be understood to represent indirect relationships. Here, a path is named by concatenating the dependency relationships and lexical items along the way but excluding the lexical items at the end.

[3]Technically, these templates represent *inference rules*, i.e. the RHS can be inferred from the LHS but is not semantically equivalent to it. This form of inference is closely related to that exhibited in textual entailment. This work is primarily to induce such rules rather than strict paraphrases.

In this way, a path can actually be thought of as a pattern with variables at either end. Consider the first dependency tree in Figure 2.2. One dependency path that could be extracted would be between the node *John* and the node *problem*. Starting at *John*, the first item in the tree is the dependency relation *subject* that connects a noun to a verb and so that information is appended to the path.[4] The next item in the tree is the word *found* and its lemma (*find*) is appended to the path.

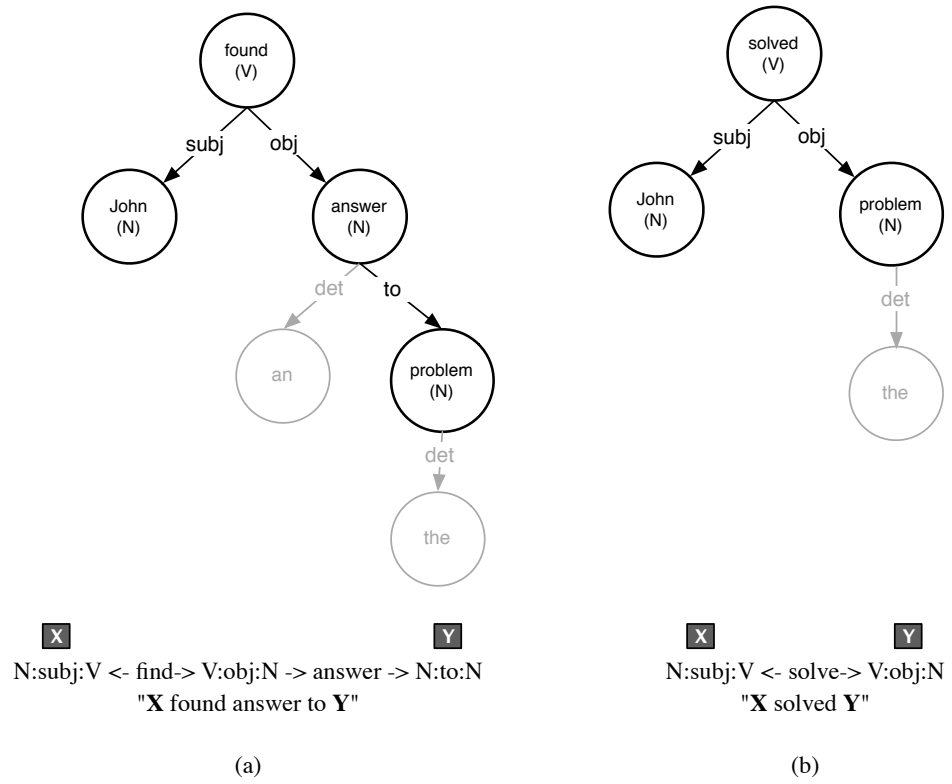


Figure 2.2: Two different dependency tree paths (a) and (b) that are considered paraphrastic since the same words ("*John*" and "*problem*") are used to fill the corresponding slots (shown co-indexed) in both the paths. The implied meaning of each dependency path is also shown.

Next is the semantic relation *object* connecting a verb to a noun, so that is

[4]Although the first item is the word *John*, the words at either end are, by definition, are considered *slots* and not included in the path.

Algorithm 2 (Lin and Pantel 2001). Produce inference rules from a parsed corpus C.

Summary. Adapt Harris' hypothesis of distributional similarity to paths in dependency trees: if two tree paths have similar distributions, i.e., they tend to link the same set of words, then they are likely mean the same thing and together generate an inference rule.

1: Extract paths of the form described above from the parsed corpus.
2: Initialize a hash H that stores, for each tuple of the form (p, s, w)—where p is a path, s is one of the two slots in p and w is a word that appears in that slot—the following two quantities:

   (a) A count $C(p, s, w)$ indicating how many times word w appeared in slot s in path p.

   (b) The mutual information $I(p, s, w)$ indicating the strength of association between slot s and word w in path p:

$$I(p, s, w) = \frac{C(p, s, w) \ \sum_{p', w'} C(p', s, w')}{\sum_{w'} C(p, s, w') \ \sum_{p'} C(p', s, w)}$$

3: **for** each extracted path p **do**
4: Find all instances (p, w_1, w_2) such that p connects the words w_1 and w_2.
5: **for** each such instance **do**
6: Update $C(p, SlotX, w_1)$ and $I(p, SlotX, w_1)$ in H.
7: Update $C(p, SlotY, w_2)$ and $I(p, SlotY, w_2)$ in H.
8: **end for**
9: **end for**
10: **for** each extracted path p **do**
11: Create a candidate set $\mathcal{C}$ of similar paths by extracting all paths from H that share at least one feature with p.
12: Prune candidates from $\mathcal{C}$ based on feature overlap with p.
13: Compute the similarity between p and the remaining candidates in $\mathcal{C}$. The similarity is defined in terms of the various values of mutual information I between the paths' two slots and all the words that appear in those slots.
14: Output all paths in $\mathcal{C}$ sorted by their similarity to p.
15: **end for**

appended. The process continues until the other slot (the word *problem*) is reached, at which point the process terminates.[5] The extracted path is shown below the tree. Similarly, a path can be extracted for the second dependency tree. Let's briefly mention the terminology associated with such paths:

- The relations on either end of a path are referred to as $SlotX$ and $SlotY$.
- The tuples $(SlotX, John)$ and $(SlotY, problem)$ are known as the two *features* of the path.
- The dependency relations inside the path that are not slots are termed *internal relations*.

Intuitively, one can imagine a path to be a complex representation of the pattern X *finds answer to* Y, where X and Y are variables. This representation for a path is a perfect fit for the extended distributional similarity hypothesis discussed above: if similar sets of words fill the same variables for two different patterns, then the patterns may be considered to have similar meaning, which is indeed the case for the paths in Figure 2.

Lin and Pantel (2001) use newspaper text as their input corpus and create dependency parses for all the sentences in the corpus in the pre-processing step. Algorithm 2 provides the details of the rest of the process: Steps 1 and 2 extract the paths and compute their distributional properties and steps 3–14 extract pairs of paths are *similar*, insofar as such properties are concerned.[6] At the end, there

[5] Any relations not connecting two content words, such as determiners and auxiliaries, are ignored.

[6] A demo of the algorithm is available online at `http://demo.patrickpantel.com/Content/LexSem/paraphrase.htm`.

are sets of paths (or inference rules) that are considered to have similar meaning by the algorithm.

The performance of their dependency path based algorithm depends heavily on the root of the extracted path. For example, while verbs frequently tend to have several modifiers, nouns tend to have no more than one. However, if a word has any fewer than two modifiers, no path can go through it as the root. Therefore, the algorithm tends to perform better for paths with verbal roots. Another issue is that this algorithm, despite the use of more informative distributional features, can generate several incorrect or implausible paraphrase patterns (inference rules). Recent work has shown how to filter out incorrect inferences when using them in a downstream application (Pantel et al., 2007).

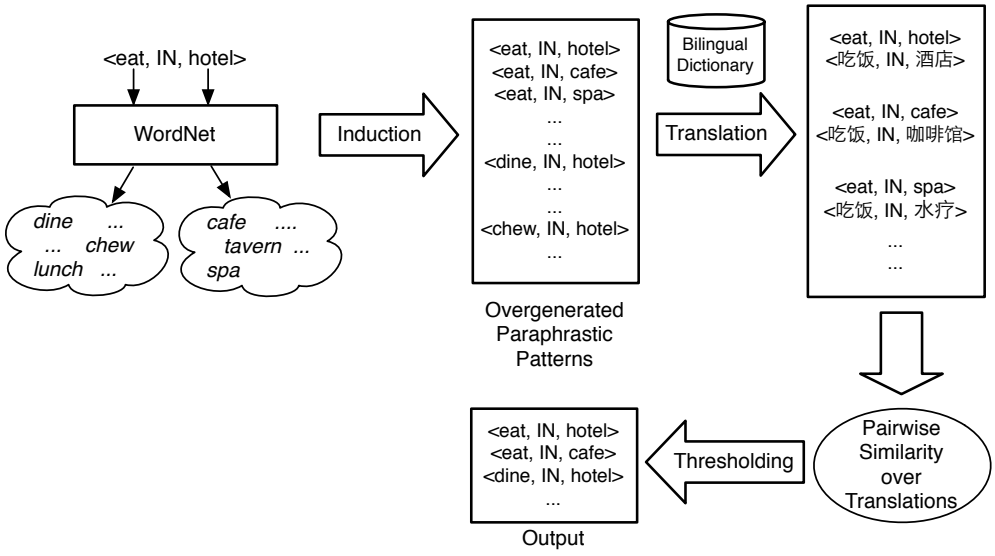


Figure 2.3: Using Chinese translations as the distributional elements to extract a set of English paraphrastic patterns from a large English corpus.

Finally, there is no reason for the distributional features to be in the same language as the one in which the paraphrases are desired. Wu and Zhou (2003)

describe a bilingual approach to extract English relation-based paraphrastic patterns of the form $\langle w_1, R, w_2 \rangle$, where w_1 and w_2 are English words connected by a dependency link with the semantic relation R. Figure 2.3 shows a simple example based on their approach. First, instances of one type of pattern are extracted from a parsed monolingual corpus. In the figure, for example, a single instance of the pattern $\langle$*verb*, IN, *pobj*$\rangle$ has been extracted. several new, potentially paraphrastic, English candidate patterns are then induced by replacing each of the English words with its synonyms in WordNet, one at a time. The figure shows the list of induced patterns for the given example. Next, each of the English words in each candidate pattern is translated to Chinese, via a bilingual dictionary.[7]

Given that the bilingual dictionary may contain multiple Chinese translations for a given English word, several Chinese patterns may be created for each English candidate pattern. Each Chinese pattern is assigned a probability value via a simple bag-of-words translation model (built from a small bilingual corpus) and a language model (trained on a Chinese collocation database); all translated patterns, along with their probability values, are then considered to be *features* of the particular English candidate pattern. Any English pattern can subsequently be compared to another by computing cosine similarity over their shared "features." English collocation pairs whose similarity value exceed some threshold are construed to be paraphrastic.

The theme of a trade-off between the precision of the generated paraphrase set— by virtue of the increased informativeness of the distributional features—and

[7]The semantic relation R is deemed to be invariant under translation.

its coverage is seen in this work as well. When using translations from the bilingual dictionary, a knowledge-rich resource, the authors report significantly higher precision than comparable methods that rely only on monolingual information to compute the distributional similarity. Predictably, they also find that recall values obtained with their dictionary-based method is lower than those obtained by other methods.

Paraphrase generation techniques using a single monolingual corpus have to rely on some form of distributional similarity since there are no explicit clues available that indicate semantic equivalence. The next section looks at paraphrasing methods operating over data that does contain such explicit clues.

2.4.3 Paraphrasing using Monolingual Parallel Corpora

It is also possible to generate paraphrastic phrase pairs from a parallel corpus where each component of the corpus is in the same language. Obviously, the biggest advantage of parallel corpora is that the sentence pairs are paraphrases almost by definition; they represent different renderings of the same meaning created by different translators making different lexical choices. In effect, they contain pairs (or sets) of sentences available that are either semantically equivalent (sentential paraphrases) or have significant semantic overlap. Extraction of phrasal paraphrases can then be effected by extracting phrasal correspondences from a set of sentences that represent the same (or similar) semantic content. Four techniques are presented in this section that generate paraphrases by finding such correspondences.

The first two techniques attempt to do so by relying, again, on the paradigm of distributional similarity: one by positing a bootstrapping distributional similarity algorithm and the other by simply adapting the previously described dependency path similarity algorithm to work with a *parallel* corpus The next two techniques rely on more direct, non-distributional methods to compute the required correspondences.

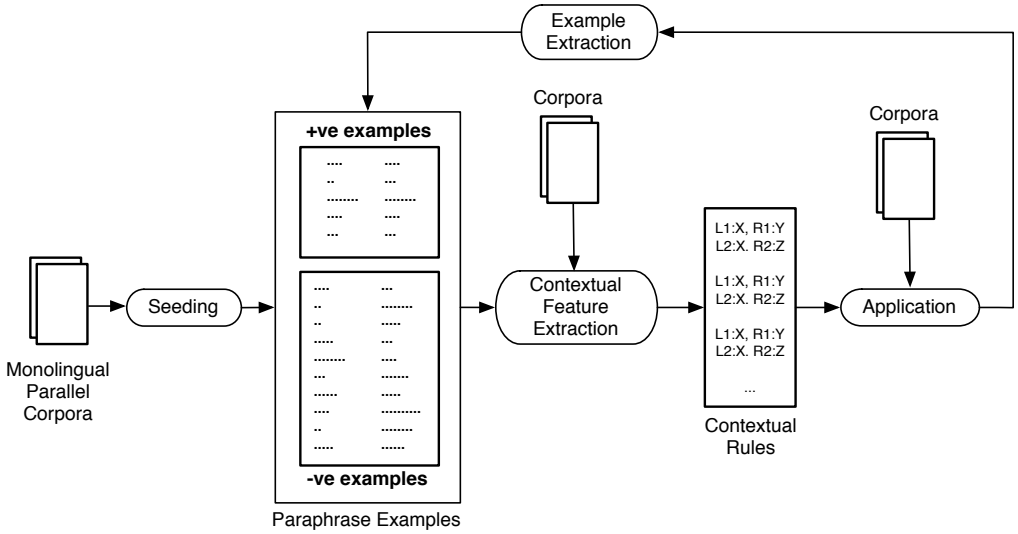


Figure 2.4: A bootstrapping algorithm to extract phrasal paraphrase pairs from monolingual parallel corpora.

Barzilay and McKeown (2001) align phrasal correspondences by attempting to move beyond a single-pass distributional similarity method. They propose a bootstrapping algorithm that allows for the gradual refinement of the features used to determine similarity and yields improved paraphrase pairs. As their input corpus, they use multiple human-written English translations of literary texts such as *Madame Bovary* and *Twenty Thousand Leagues Under the Sea* that are expected to be rich in paraphrastic expressions because different translators would use their own words while still preserving the meaning of the original text. The parallel com-

ponents are obtained by performing sentence alignment (Gale and Church, 1991) on the corpora to obtain sets of parallel sentences that are then lemmatized, part-of-speech tagged and chunked in order to identify all the verb and noun phrases. The bootstrapping algorithm is then employed to incrementally learn better and better contextual features that are then leveraged to generate semantically similar phrasal correspondences.

Figure 2.4 shows the basic steps of the algorithm. To seed the algorithm, some fake paraphrase examples are extracted by using identical words from either side of the aligned sentence pair. For example, given the following sentence pair:

S_1: *Emma burst into tears and he tried to comfort her.*

S_2: *Emma cried and he tried to console her.*

(*tried*, *tried*), (*her*, *her*) may be extracted as positive examples and (*tried*, *Emma*), (*tried*, *console*) may be extracted as negative examples. Once the seeding examples are extracted, the next step is to extract contextual features for both the positive and the negative examples. These features take the form of aligned part-of-speech sequences of a given length from the left and the right of the example. For instance, the contextual feature $[\langle \mathtt{L}_1 : \mathtt{PRP}_1, \mathtt{R}_1 : \mathtt{TO}_1 \rangle, \langle \mathtt{L}_2 : \mathtt{PRP}_1, \mathtt{R}_2 : \mathtt{TO}_1 \rangle]$ of length 1 can be extracted for the positive example (*tried*, *tried*) above. This particular contextual feature contains two tuples, one for each sentence. The first tuple $\langle \mathtt{L}_1 : \mathtt{PRP}_1, \mathtt{R}_1 : \mathtt{TO}_1 \rangle$ indicates that, in the first sentence, the POS tag sequence the left of the word *tried* is just the personal pronoun (*he*) and the POS tag sequence to the right of *tired* is just the preposition *to*. The second tuple is identical for this case. Note that

the tags of identical tokens are indicated as such by suffixes on the POS tags. All such features are extracted for both the positive and the negative examples for all lengths less than or equal to some specified length. In addition, a strength value is calculated for each positive (negative) contextual feature f using maximum likelihood estimation as follows:

$$\text{strength}(f) = \frac{\text{Number of positive (negative) examples surrounded by } f}{\text{Total occurrences of} f}$$

The extracted list of contextual features is thresholded on the basis of the above strength value. The remaining contextual rules are then applied to the corpora to obtain additional positive and negative paraphrase examples that, in turn, lead to more refined contextual rules and so on. The process is repeated for a fixed number of iterations or until no new paraphrase examples are produced. The list of extracted paraphrases at the end of the final iteration represents the final output of the algorithm. In total, about 9000 phrasal (including lexical) paraphrases are extracted from 11 translations of 5 works of classic literature. Furthermore, the extracted paraphrase pairs are also generalized into about 25 patterns by extracting part-of-speech tag sequences corresponding to the tokens of the paraphrase pairs.

Barzilay and McKeown also perform an interesting comparison with another technique that was originally developed for compiling translation lexicons from bilingual parallel corpora (Melamed, 2001). This technique first compiles an initial lexicon using simple co-occurrence statistics and then uses a competitive linking algorithm (Melamed, 1997) to improve the quality of the lexicon. The authors

apply this technique to their monolingual parallel data and observe that the extracted paraphrase pairs are of much lower quality than the pairs extracted by their own method. Similar observations are presented in Section 2.4.5 which highlight that while more recent translation techniques—specifically ones that use phrases as units of translation—are better suited to the task of generating paraphrases than the competitive linking approach, they continue to suffer from the same problem of low precision. On the other hand, such techniques can take good advantage of large bilingual corpora and capture a much larger variety of paraphrastic phenomena.

Ibrahim et al. (2003) propose an approach that applies a modified version of the dependency path distributional similarity algorithm proposed by Lin and Pantel (2001) to the exact monolingual parallel corpus (multiple translations of literary works) used by Barzilay and McKeown (2001). The authors claim that their technique is more tractable than Lin and Pantel's approach since the sentence-aligned nature of the input parallel corpus obviates the need to compute similarity over tree paths drawn from sentences that have zero semantic overlap. Furthermore, they also claim that their technique exploits the parallel nature of a corpus more effectively than Barzilay and McKeown's simply because their technique is the one that uses tree paths and not just lexical information. Specifically, they propose the following modifications to Lin and Pantel's algorithm:

1. **Extracting tree paths with aligned anchors**. Rather than using a single corpus and comparing paths extracted from possibly unrelated sentences, the authors leverage sentence-aligned monolingual parallel corpora; the same as

used in (Barzilay and McKeown, 2001). For each sentence in an aligned pair, anchors are identified. The anchors from both sentences are brought into alignment. Once anchor pairs on either side have been identified and aligned, a breadth-first-search algorithm is used to find the shortest path between the anchor nodes in the dependency trees. All paths found between anchor pairs for a sentence pair are taken to be distributionally—and, hence, semantically—similar.

2. **Using a sliding frequency measure**. The original dependency-based algorithm (Lin and Pantel, 2001) weights all subsequent occurrences of the same paraphrastic pair of tree paths as much as the first one. In this version, every successive induction of a paraphrastic pair using the same anchor pair is weighted less than the previous one. Specifically, inducing the same paraphrase pair using an anchor pair that has already been seen only counts for $\frac{1}{2^n}$, where n is the number of times the specific anchor pair has been seen so far. Therefore, induction of a path pair using *new* anchors is better evidence that the pair's paraphrastic, as opposed to the repeated induction of the path pair from the *same* anchor over and over again.

Despite the authors' claims, they offer no quantitative evaluation comparing their paraphrases with those from Lin and Pantel (2001) or from Barzilay and McKeown (2001).

It is also possible to find correspondences between the parallel sentences using a more direct approach instead of relying on distributional similarity. Pang et al.

(2003) propose an algorithm to align sets of parallel sentences driven entirely by the syntactic representations of the sentences. The alignment algorithm outputs a merged lattice from which lexical, phrasal as well as sentential paraphrases can simply be read off. More specifically, they use the Multiple-Translation Chinese corpus that was originally developed for machine translation evaluation and contains 11 human-written English translations for each sentence in a news document. Using several sentences explicitly equivalent in semantic content has the advantage of being a richer source for paraphrase induction.

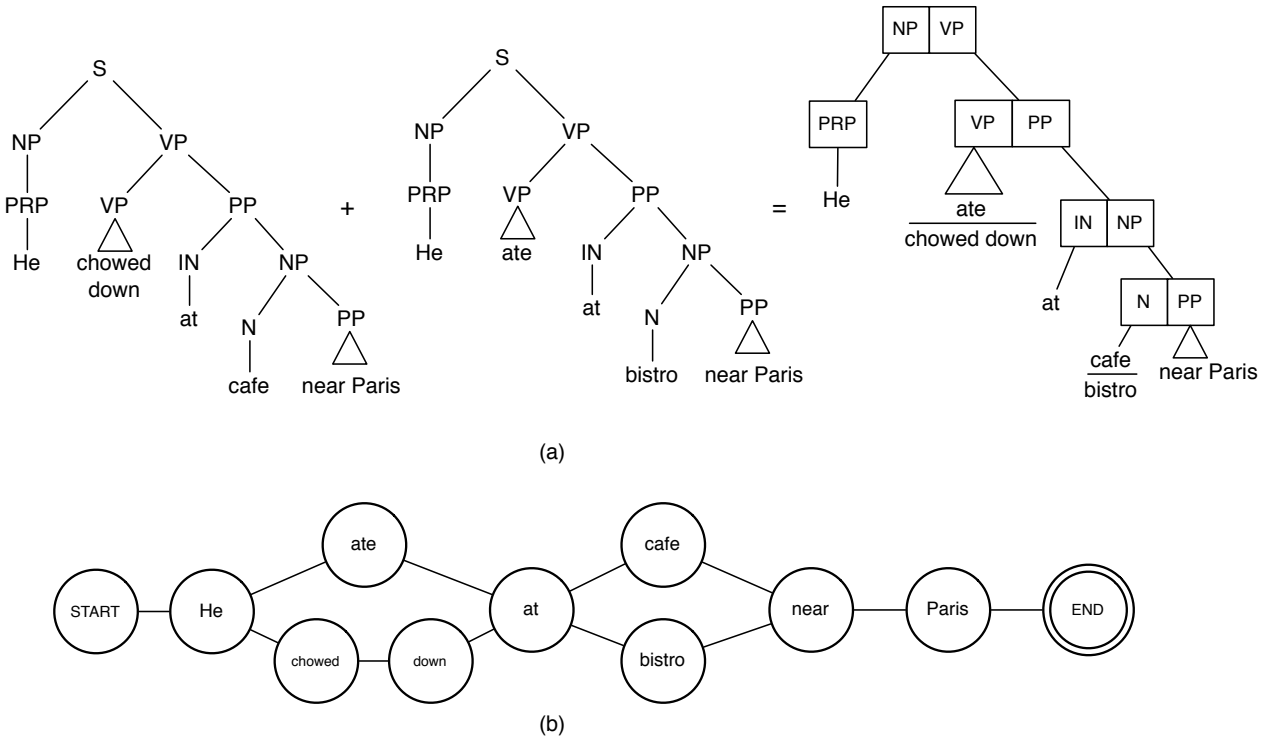


Figure 2.5: (a) shows how the merging algorithm works for two simple parse trees to produce a shared forest. Note that in order to preserve clarity, not all constituents are expanded fully. Leaf nodes with two entries represent paraphrases. (b) shows the word lattice generated by linearizing the forest in (a).

As a pre-processing step, any group (of 11 sentences) that contains sentences longer than 45 words are discarded. Next, each sentence in each of the groups

is parsed. All the parse trees are then iteratively merged into a shared forest. The merging algorithm proceeds top-down and continues to recursively merge constituent nodes that are expanded identically. It stops upon reaching the leaves or upon encountering the same constituent node expanded using different grammar rules. Figure 2.5(a) shows how the merging algorithm would work on two simple parse trees. In the figure, it is apparent that the leaves of the forest encode paraphrasing information. However, the merging only allows identical constituents to be considered as paraphrases. In addition, keyword-based heuristics need to be employed to prevent inaccurate merging of constituent nodes due to, say, alternations of active and passive voices among the sentences in the group. Once the forest is created, it is linearized to create the word lattice by traversing the nodes in the forest top-down and producing an alternative path in the lattice for each merged node. Figure 2.5(b) shows the word lattice generated for the simple two-tree forest. The lattices also require some post-processing to remove redundant edges and nodes that may have arisen due to parsing errors or limitations in the merging algorithm. The final output of the paraphrasing algorithm is a set of word lattices, one for each sentence group.

These lattices can be used as sources of lexical as well as phrasal paraphrases. All alternative paths between any pair of nodes can be considered to be paraphrases of each other. For example, besides the obvious lexical paraphrases, the paraphrase pair ⟨*ate at cafe, chowed down at bistro*⟩ can also be extracted from the lattice in Figure 2.5(b). In addition, each path between the `START` and the `END` nodes in the lattice represents a sentential paraphrase of the original 11 sentences used to create

the lattice.

The direct alignment approach is able to leverage the sheer width (number of parallel alternatives per sentence position; 11 in this case) of the input corpus to do away entirely with any need for measuring distributional similarity. In general, it has several advantages. It can capture a very large number of paraphrases: each lattice has on the order of hundreds or thousands of paths depending on the average length of the sentence group that it was generated from. In addition, the paraphrases produced are of better quality than other approaches employing parallel corpora for paraphrase induction discussed so far. However, the approach does have a couple of drawbacks:

- **No paraphrases for unseen data**. The lattices cannot be applied to new sentences for generating paraphrases since no form of generalization is performed to convert lattices into patterns.

- **Requirement of large number of human-written translations.** Each of the lattices described above is built using 11 manually written translations of the same sentence, each by a different translator. There are very few corpora that provide such a large number of human translations. In recent years, most MT corpora have had no more than 4 references, which would certainly lead to much sparser word lattices and number of paraphrases that can be extracted. In fact, given the cost and amount of effort required for humans to translate a relatively large corpus, it is common to encounter corpora with only a single human translation. With such a corpus, of course, this technique would be

unable to produce any paraphrases. One solution might be to augment the relatively few human translations with translations obtained from automatic machine translation systems. In fact, the corpus used (Huang et al., 2002) also contains, besides the 11 human translations, 6 translations of the same sentence by machine translation systems available on the Internet at the time. However, no experiments are performed with the automatic translations.

Finally, an even more direct method to align equivalences in parallel sentence pairs can be effected by building on word alignment techniques from the field of statistical machine translation (Brown et al., 1990). Current state-of-the-art SMT methods rely on unsupervised induction of word alignment between two bilingual parallel sentences to extract translation equivalences that can then be used to translate a given sentence in one language into another language. The same methods can be applied to monolingual parallel sentences without any loss of generality. Quirk et al. (2004) use one such method to extract phrasal paraphrase pairs. Furthermore, they use these extracted phrasal pairs to construct sentential paraphrases for new sentences.

Mathematically, their approach to sentential paraphrase generation may be expressed in terms of the typical channel model equation for statistical machine translation:

$$\mathbf{E}_{\mathbf{p}}^{*} = \arg\max_{\mathbf{E_p}} P(\mathbf{E_p}|\mathbf{E}) \tag{2.1}$$

The equation denotes the search for the optimal paraphrase $\mathbf{E_p}$ for a given sentence

E. This may be rewritten using Bayes' Theorem as follows:

$$\mathbf{E_p^*} = \arg\max_{\mathbf{E_p}} P(\mathbf{E_p})\ P(\mathbf{E}|\mathbf{E_p})$$

where $P(\mathbf{E_p})$ is an n-gram *language model* providing a probabilistic estimate of the fluency of a hypothesis $\mathbf{E_p}$ is and $P(\mathbf{E}|\mathbf{E_p})$ is the *translation*, or more appropriately for paraphrasing, the *replacement model* providing a probabilistic estimate of what is essentially the semantic adequacy of the hypothesis paraphrase. Therefore, the optimal sentential paraphrase may loosely be described as one that fluently captures most, if not all, of the meaning contained in the input sentence.

It is important to provide a brief description of the parallel corpus used here since unsupervised induction of word alignments typically requires relatively large number of parallel sentence pairs. The monolingual parallel corpus (or more accurately, quasi-parallel since not all sentence pairs are fully semantically equivalent) is constructed by scraping online news sites for clusters of articles on the same topic. Such clusters contain the full text of each article and the dates and times of publication. After removing the mark-up, they discard any pair of sentences in a cluster where the difference in the lengths or the edit distance is larger than some stipulated value. This method yields a corpus containing approximately $140,000$ quasi-parallel sentence pairs $\{(\mathbf{E_1}, \mathbf{E_2})\}$, where $\mathbf{E_1} = e_1^1 e_1^2 \dots e_1^m$, $\mathbf{E_2} = e_2^1 e_2^2 \dots e_2^n$.

The examples below show that the proposed method can work well:

S_1: *In only 14 days, US researchers have created an artificial bacteria-eating virus*

Algorithm 3 (Quirk, Dolan, and Brockett 2004). Generate a set M of phrasal paraphrases with associated likelihood values from a monolingual parallel corpus C.
Summary. Estimate a simple English to English phrase translation model from C using word alignments. Use this model to create sentential paraphrases as explained later.

$M \leftarrow \{\phi\}$

Compute lexical replacement probabilities $P(e_1|e_2)$ from all sentence pairs in C via IBM Model 1 estimation

Compute a set of word alignments $\{\mathbf{a}\}$ such that for each sentence pair $(\mathbf{E_1}, \mathbf{E_2})$

$$\mathbf{a} = a_1 a_2 \dots a_m$$

where $a_i \in \{0 \dots n\}, m = |\mathbf{E_1}|, n = |\mathbf{E_2}|$

for each word-aligned sentence pair $(\mathbf{E_1}, \mathbf{E_2})_\mathbf{a}$ in C **do**

Extract pairs of *contiguous* subsequences $(\bar{e_1}, \bar{e_2})$ such that:

(a) $|\bar{e_1}| \leq 5$, $|\bar{e_2}| \leq 5$

(b) $\forall i \in \{1, \dots, |\bar{e_1}|\}\ \exists j \in \{1, \dots, |\bar{e_2}|\},\ e_{1,i} \overset{\mathbf{a}}{\sim} e_{2,j}$

(c) $\forall i \in \{1, \dots, |\bar{e_2}|\}\ \exists j \in \{1, \dots, |\bar{e_1}|\},\ e_{2,i} \overset{\mathbf{a}}{\sim} e_{1,j}$

Add all extracted pairs to M.

end for

for each paraphrase pair $(\bar{e_1}, \bar{e_2})$ in M **do**

Compute $P(\bar{e_1}|\bar{e_2}) = \prod_{e_1^j \in \bar{e_1}} \sum_{e_2^k \in \bar{e_2}} P(e_1^j|e_2^k)$

end for

Output M containing paraphrastic pairs and associated probabilities

---

from synthetic genes.

S_2: *An artificial bacteria-eating virus has been made from synthetic genes in the record time of just two weeks.*

S_1: *The largest gains were seen in prices, new orders, inventories and exports.*

S_2: *Sub-indexes measuring prices, new orders, inventories and exports increased.*

For more details on the creation of this corpus, the reader is referred to (Dolan et al., 2004) and, more generally, to Section 2.5. Algorithm 3 shows how to generate a set of phrasal paraphrase pairs and compute a probability value for each such pair. In Step 2, a simple parameter estimation technique (Brown et al., 1993) is used to compute, for later use, the probability of replacing any given word with another. Step 3 computes a word alignment (indicated by $\mathbf{a}$) between each pair of sentences. This alignment indicates for each word e_i in one string that word e_j in the other string from which it was most likely produced (denoted here by $e_i \overset{\mathbf{a}}{\sim} e_j$). Steps 4-7 extract, from each pair of sentences, pairs of short contiguous phrases that are aligned with each other according to this alignment. Note that each such extracted pair is essentially a phrasal paraphrase. Finally, a probability value is computed for each such pair by assuming that each word of the first phrase can be replaced with each word of the second phrase. This computation uses the lexical replacement probabilities computed in Step 2 are used.

Now that a set of scored phrasal pairs has been extracted, these pairs can be used to generate paraphrases for any unseen sentence. Generation proceeds by

creating a lattice for the given sentence. Given a sentence **E**, the lattice is populated as follows:

1. Create $|\mathbf{E}| + 1$ vertices $q_0, q_1 \ldots q_{|\mathbf{E}|}$

2. Create N edges between each pair of vertices q_j and q_k $(j < k)$ such that N = number of phrasal paraphrases for the input phrase $e_{(j+1)}e_{(j+2)} \ldots e_k$. Label each edge with the phrasal paraphrase string itself and its probability value. Each such edge denotes a possible paraphrasing of the above input phrase by the replacement model.

3. Add the edges $\{(q_{j-1}, q_j)\}$ and label each edge with the token s_j and a constant u. This is necessary to handle words from the sentence that do not occur anywhere in the set of paraphrases.

Figure 2.6 shows an example lattice. Once the lattice has been constructed, it is straightforward to extract the 1-best paraphrase by using a dynamic programming algorithm such as Viterbi decoding and extracting the optimal path from the lattice as scored by the product of an n-gram language model and the replacement model. In addition, as with SMT decoding, it is also possible to extract a list of n-best paraphrases from the lattice by using the appropriate algorithms (Soong and Huang, 1990; Mohri and Riley, 2002).

Quirk et al. borrow from the statistical machine translation literature so as to align phrasal equivalences as well as to utilize the aligned phrasal equivalences to rewrite new sentences. The biggest advantage of this method is its SMT inheritance: it is possible to produce *multiple* sentential paraphrases for any new sentence. and

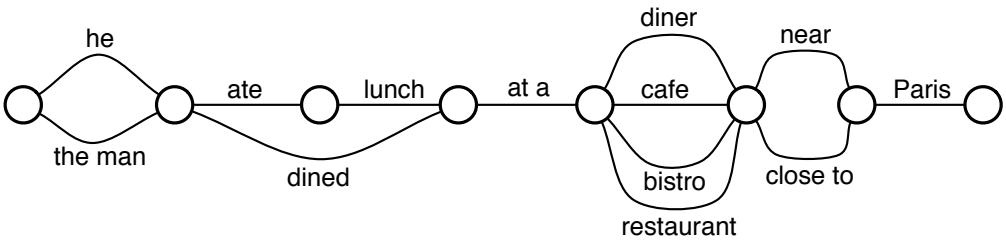


Figure 2.6: A paraphrase generation lattice for the sentence *He ate lunch at a cafe near Paris*. Alternate paths between various nodes represent phrasal replacements. The probability values associated with each edge are not shown for the sake of clarity.

there is no limit on the number of sentences that can be paraphrased.[8] However, there are certain limitations:

- **Monotonic Translation**. It is assumed that a phrasal replacement will occur in the exact same position in the output sentence as that of the original phrase in the input sentence, i.e., reorderings of phrasal units are disallowed.

- **Naive Parameter Estimation**. Using a bag-of-words approach to parameter estimation results in a relatively uninformative probability distribution over the phrasal paraphrases.

- **Reliance on edit-distance**. Relying on edit distance to build the training corpus of quasi-parallel sentences may exclude sentences that do exhibit a paraphrastic relationship but differ significantly in constituent orderings.

All of the above limitations combined lead to paraphrases that, while grammatically sound, contain very little variety. Most sentential paraphrases that are generated involve little more than simple substitutions of words and short phrases. In Sec-

[8]However, if no word in the input sentence has been observed in the parallel corpus, the paraphraser simply reproduces the original sentence as the paraphrase.

tion 2.4.5, other approaches will be discussed that also find inspiration from statistical machine translation and attempt to circumvent the above limitations by using a bilingual parallel corpus instead of a monolingual parallel corpus.

2.4.4 Paraphrasing using Monolingual Comparable Corpora

While it is clearly advantageous to have monolingual parallel corpora, such corpora are usually not very easily available. The corpora usually found in the real world are *comparable* instead of being truly parallel: parallelism between sentences is replaced by just partial semantic and topical overlap at the level of documents. Therefore, for monolingual comparable corpora, the task of finding phrasal correspondences becomes harder since the two corpora may only be related by way of describing events under the same topic. In such a scenario, possible paraphrasing methods either (a) forgo any attempts at directly finding such correspondences and fall back to the distributional similarity workhorse or, (b) attempt to directly induce a form of coarse-grained alignment between the two corpora and leverage this alignment.

This section describes three methods that generate paraphrases from such comparable corpora. The first method falls under category (a): here the elements whose distributional similarity is being measured are paraphrastic patterns and the distributions themselves are the named entities, with which the elements occur in various sentences. In contrast, the next two methods fall under category (b) and attempt to directly discover correspondences between two comparable corpora by leveraging a

novel alignment algorithm combined with some similarity heuristics. The difference between the two latter methods lies only in the efficacy of the alignment algorithm.

Shinyama et al. (2002) use two sets of 300 news articles from two different Japanese newspapers from the same day as their source of paraphrases. The comparable nature of the articles is ensured because both sets are from the same day. During pre-processing, all named entities in each article are tagged and dependency parses are created for each sentence in each article. The distributional similarity driven algorithm then proceeds as follows:

1. For each article in the first set, the most "similar" article is found from the other set, based on a similarity measure computed over the named entities appearing in the two articles.

2. From each Japanese sentence in each such pair of articles, extract all dependency tree paths that contain at least one named entity and generalize them into patterns wherein the named entities have been replaced with variables. Each class of named-entity (e.g., Organization, Person, Location etc.) gets its own variable. For example, the following sentence:[9]

   Vice President Kuroda of Nihon Yamamura Glass Corp. was promoted to President.

   may yield the following two patterns, among others:

   ⟨PERSON⟩ *of* ⟨ORGANIZATION⟩ *was promoted*

[9]While the authors provide motivating examples in Japanese (transliterated into romaji) in their paper, English is chosen here, for the sake of clarity.

⟨PERSON⟩ *was promoted to* ⟨POST⟩

3. Find all sentences in the two newswire corpora that match the above patterns. When a match is found, attach the pattern to the sentence and link all variables to the corresponding named entities in the sentences.

4. Find all sentences that are most similar to each other (above some preset threshold), again based on the named entities they share between them.

5. For each pair of similar sentences, compare their respective attached patterns. If the variables in the patterns link to the same or comparable named entities (based on the entity text and type), then consider the patterns to be paraphrases of each other.

At the end, the output is a list of generalized paraphrase patterns with named entity types as variables. For example, it may generate the following two patterns as paraphrases:

⟨PERSON⟩ *is promoted to* ⟨POST⟩

the promotion of ⟨PERSON⟩ *to* ⟨POST⟩ *is decided*

As a later refinement, Sekine (2005) makes a similar attempt at using distributional similarity over named entity pairs in order to produce a list of *fully lexicalized* phrasal paraphrases for specific concepts represented by keywords.

The idea of enlisting named entities as proxies for detecting semantic equivalence is interesting and has certainly been explored before (vide the discussion regarding (Paşca and Dienes, 2005) in Section 2.4.2). However, it has some obvious

disadvantages. The authors manually evaluate the technique by generating paraphrases for two specific domains: arrest events and personnel hirings and find that while the precision is reasonably good, the coverage is very low primarily due to restrictions on the patterns that may be extracted in Step 2 above. In addition, if the average number of entities in sentences is low, the likelihood of creating incorrect paraphrases is confirmed to be higher.

Consider the altogether separate idea of deriving coarse-grained correspondences by leveraging the comparable nature of the corpora. Barzilay and Lee (2003) attempt to do so by generating compact sentence clusters in template form (stored as word lattices with slots) separately from each corpora and then pairing up templates from one corpus with those from the other. Once the templates are paired up, a new incoming sentence that matches one member of a template pair gets rendered as the other member, thereby generating a paraphrase. They use as input a pair of corpora: the first (C_1) consisting of clusters of news articles published by Agence France Presse (AFP) and the second (C_2) consisting of those published by Reuters. The two corpora may be considered comparable since the articles in each are related to the same topic and were published during the same time frame.

Algorithm 4 shows how some details of their technique works. Steps 3–18 show how to cluster topically related sentences, construct a word lattice from such a cluster and convert that into a *slotted lattice*—basically a word lattice with certain nodes recast as variables or empty slots. The clustering is done so as to bring together sentences pertaining to the same topics and having similar structure. The word

Algorithm 4 (Barzilay and Lee 2003). Generate set M of matching lattice pairs given a pair of comparable corpora C_1 and C_2.
Summary. Gather topically related sentences from C_1 into clusters. Do the same for C_2. Convert each sentence cluster into a slotted lattice using a multiple-sequence alignment (MSA) algorithm. Compare all lattice pairs and output those likely to be paraphrastic.

Let W_{C_1} and W_{C_2} represent word lattices obtained from C_1 and C_2 respectively
$M \leftarrow \{\phi\}$, $W_{C_1} \leftarrow \{\phi\}$, $W_{C_2} \leftarrow \{\phi\}$
for each input corpus $C_i \in \{C_1, C_2\}$ **do**
  Create a set of clusters $G_{C_i} = \{G_{C_i,k}\}$ of sentences based on n-gram overlap such that all sentences in a cluster describe the same kinds of events and share similar structure.
  for each cluster $G_{C_i,k}$ **do**
    Compute an MSA for all sentences in $G_{C_i,k}$ by using a pre-stipulated scoring function and represent the output as a word lattice $W_{C_i,k}$
    Compute the set of backbone nodes B_k for $W_{C_i,k}$, i.e., the nodes that are shared by a majority ($>= 50\%$) of the sentences in $G_{C_i,k}$
    for each backbone node $b \in B_k$ **do**
      if no more than 30% of all the edges from b lead to the same node **then**
        Replace all nodes adjacent to b with a single slot
      else
        Delete any node with $< 30\%$ of the edges from b leading to it and preserve the rest
      end if
    end for
    Merge any consecutive slot nodes into a single slot
    $W_{C_i} \leftarrow W_{C_i} \cup \{W_{C_i,k}\}$
  end for
end for
for each lattice pair $(W_{C_1,j}, W_{C_2,k}) \in W_{C_1} \times W_{C_2}$ **do**
  Inspect clusters $G_{C_1,j}$ and $G_{C_2,k}$ and compare slot fillers in the cross-corpus sentence pairs written on the same day
  if comparison score $>$ a prestipulated threshold δ **then**
    $M \leftarrow M \cup \{(W_{C_1,j}, W_{C_2,k})\}$
  end if
end for
Output M containing paraphrastic lattice pairs with linked slots

lattice is the product of an algorithm that computes a *multiple-sequence alignment* (MSA) for a cluster of sentences (Step 6). A very brief outline of such an algorithm, originally developed to compute an alignment for a set of 3 or more protein or DNA sequences,[10] is as follows:

1. Find the most similar pair of sentences in the cluster according to a similarity scoring function. For this approach, a simplified version of the edit-distance measure (Barzilay and Lee, 2002) is used.

2. Align this sentence pair and replace the pair with this single alignment.

3. Repeat until all sentences have been aligned together.

The word lattice so generated now needs to be converted into a slotted lattice to allow its use as a paraphrase template. Slotting is performed based on the following intuition: areas of high variability between backbone nodes, i.e., several distinct parallel paths, may correspond to template arguments and can be collapsed into one path leading to a slot that can be filled by these arguments. However, multiple parallel paths may also appear in the lattice because of simple synonymy and those paths must be retained for paraphrase generation to be useful. To differentiate between the two cases, a *synonymy threshold* s is used, set to 30%. It is used in the algorithm as shown in Steps 8–14. The basic idea behind the threshold is that as the number of sentences increases, the number of different arguments will increase faster than the number of synonyms. Figure 2.7 shows how a very simple word lattice may be generalized into a slotted lattice.

[10]For more details on MSA algorithms, refer to (Gusfield, 1997; Durbin et al., 1998).

Once all the slotted lattices have been constructed for each corpus, Steps 19–24 try to match the slotted lattices extracted from one corpus to those extracted from the other by referring back to the sentence clusters from which the original lattices were generated, comparing the sentences that were written on the *same day* and computing a comparison score based on overlap between the sets of arguments that fill the slots. If this computed score is greater than some fixed threshold value δ, then the two lattices (or patterns) are considered to be paraphrases of each other.

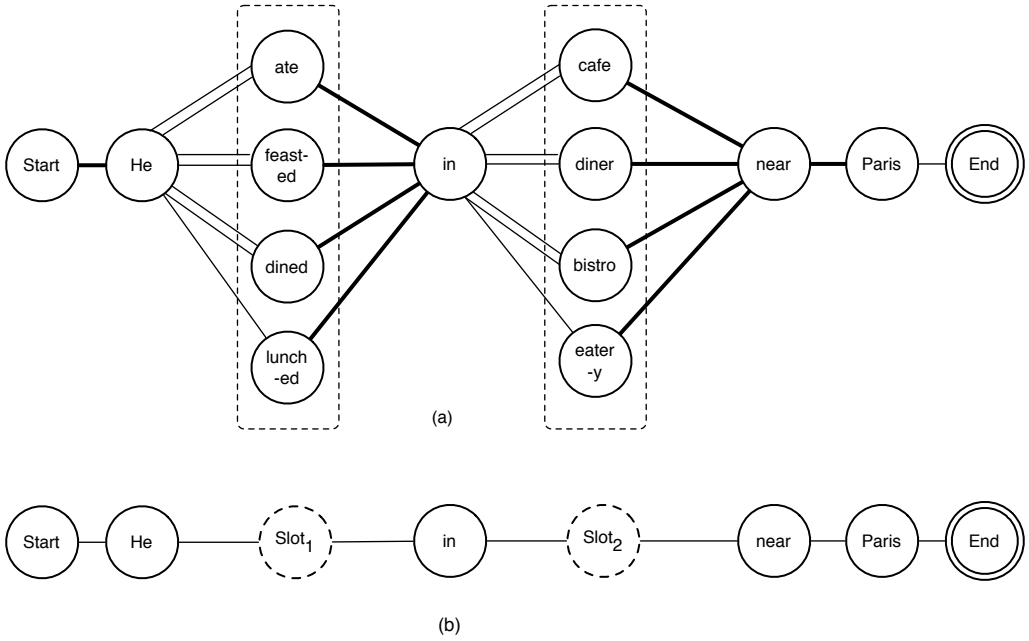


Figure 2.7: An example showing the generalization of the word lattice (a) into a slotted lattice (b). The word lattice is produced by aligning 7 sentences. Nodes having in-degrees > 1 occur in more than one sentence. Nodes with thick incoming edges occur in all sentences.

Besides generating pairs of paraphrastic patterns, the authors go one step further and actually use the patterns to generate paraphrases for new sentences. Given such a sentence S, the first step is to find an existing slotted lattice from either corpus that aligns best with S, in terms of the previously mentioned alignment

scoring function. If some lattice is found as a match, then all that remains is to take all corresponding lattices from the other corpus that are paired with this lattice and use them to create multiple rewritings (paraphrases) for S. Rewriting in this context is a simple matter of substitution: for each slot in the matching lattice, both the argument from the sentence that fills it and the slot in the corresponding rewrite lattice are known.

As far as quality of acquired paraphrases is concerned, this approach easily outperforms almost all other sentential paraphrasing approaches described in this chapter. However, a paraphrase is produced *only* if the incoming sentence matches some existing template which leads to a strong bias favoring quality over coverage. In addition, construction and generalization of lattices may become computationally expensive when dealing with much larger corpora.

Barzilay and Lee's work can be compared and contrasted with the work from Section 2.4.3 that seems most closely related: that of Pang et al. (2003). Both take sentences grouped together in a cluster and align them into a lattice using a particular algorithm. Pang et al. have a pre-defined size for all clusters since the input corpus is an 11-way parallel corpus. However, Barzilay and Lee have to construct the clusters from scratch since their input corpus has no predefined notion of parallelism at the sentence level. Both approaches use word lattices to represent and induce paraphrases since a lattice can efficiently and compactly encode n-gram similarities (sets of shared overlapping word sequences) between a large number of sentences. However, the two approaches are also different in that Pang et al. use the parse trees of all sentences in a cluster to compute the alignment (and build the

lattice) whereas Barzilay and Lee use only surface level information. Furthermore, Barzilay and Lee can use their slotted lattice pairs to generate paraphrases for novel and unseen sentences whereas Pang et al. cannot paraphrase new sentences at all.

Shen et al. (2006) attempt to improve Barzilay and Lee's technique by trying to include syntactic constraints in the cluster alignment algorithm. In that way, it is doing something similar to what Pang et al. do but with a comparable corpus instead of a parallel one. More precisely, whereas Barzilay and Lee use a relatively simple alignment scoring function based on purely lexical features, Shen et al. try to bring syntactic features into the mix. The motivation is to constrain the relatively free nature of the alignment generated by the MSA algorithm—which may lead to the generation of grammatically incorrect sentences—by using informative syntactic features. In their approach, even if two words are a *lexical match*—as defined by Barzilay and Lee (2003)—they are further inspected in terms of certain pre-defined syntactic features. Therefore, when computing the alignment similarity score, two lexically matched words across a sentence pair are not considered to fully match unless their score on syntactic features also exceeds a preset threshold.

The syntactic features constituting the additional constraints are defined in terms of the output of a *chunk parser*. Such a parser takes as input the syntactic trees of the sentences in a topic cluster and provides the following information for each word:

- **Part-of-speech tag**
- **IOB Tag**. This is a notation denoting the constituent covering a word and

its relative position in that constituent (Ramshaw and Marcus, 1995). For example, if a word has the tag I-NP, it can be inferred that the word is covered by an NP and located inside that NP. Similarly, B denotes that the word is at the beginning and O denotes that the word is not covered by any constituent.

- **IOB chain**. A concatenation of all IOB tags going from the root of the tree to the word under consideration.

With the above information and a heuristic to compute the similarity between two words in terms of their POS and IOB tags, the alignment similarity score can be calculated as the sum of the heuristic similarity value for the given two words and the heuristic similarity values for each corresponding node in the two IOB chains. If this score is higher than some threshold and the two words have similar positions in their respective sentences, then the words are considered to be a match and can be aligned. Given this alignment algorithm, the word lattice representing the global alignment is constructed in an iterative manner similar to the MSA approach.

Shen et al. present evidence from a manual evaluation that sentences sampled from lattices constructed via the syntactically informed alignment method receive higher grammaticality scores as compared to sentences from the lattices constructed via the purely lexical method. However, they present no analysis of the actual paraphrasing capacity of their, presumably better aligned, lattices. Indeed, they explicitly mention that their primary goal is to measure the correlation between the syntax-augmented scoring function and the *correctness* of the sentences being generated from such lattices, even if the sentences do not bear a paraphrastic relationship

to the input. Even if one were to assume that the syntax-based alignment method would result in better paraphrases, it still wouldn't address the primary weakness of Barzilay and Lee's method: paraphrases are only generated for new sentences that match an already existing lattice, leading to lower coverage.

2.4.5 Paraphrasing using Bilingual Parallel Corpora

In the last decade, there has been a resurgence in research on statistical machine translation (SMT). There has been also been an accompanying dramatic increase in the number of available bilingual parallel corpora due to the strong interest in SMT from both the public and private sectors. Recent research in paraphrase generation has attempted to leverage these very large bilingual corpora. This section looks at such approaches that rely on the preservation of meaning across languages and try to recover said meaning by using cues from the secondary language.

Using bilingual parallel corpora for paraphrasing has the inherent advantage that sentences in the other language are *exactly* semantically equivalent to sentences in the intended paraphrasing language. Therefore, the most common way to generate paraphrases with such a corpus exploits both its parallel and bilingual natures: align phrases across the two languages and consider all co-aligned phrases in the intended language to be paraphrases. The bilingual phrasal alignments can simply be generated by using the automatic techniques developed for the same task in the SMT literature. Therefore, arguably the most important factor affecting the performance of these techniques is usually the quality of the automatic bilingual phrasal

Algorithm 5 (Bannard and Callison-Burch 2005). Generate set M of monolingual paraphrase pairs given a bilingual corpus C.

Summary. Extract bilingual phrase pairs from C using word alignments and standard SMT heuristics. Pivot all pairs of English phrases on any shared foreign phrases and consider them paraphrases. The alignment notation from Algorithm 3 is employed.

Let B represent the bilingual phrases extracted from C
$B \leftarrow \{\phi\}$, $M \leftarrow \{\phi\}$
Compute a word alignment $\mathbf{a}$ for each sentence pair $(\mathbf{E}, \mathbf{F}) \in C$
for each aligned sentence pair $(\mathbf{E}, \mathbf{F})_{\mathbf{a}}$ **do**
    Extract the set of bilingual phrasal correspondences $\{(\bar{e}, \bar{f})\}$ such that:

$$\text{(a) } \forall e_i \in \bar{e} : e_i \overset{\mathbf{a}}{\sim} f_j \rightarrow f_j \in \bar{f}, \text{ and}$$

$$\text{(a) } \forall f_j \in \bar{f} : f_j \overset{\mathbf{a}}{\sim} e_i \rightarrow e_i \in \bar{e}$$

    $B \leftarrow B \ \cup \{(\bar{e}, \bar{f})\}$
end for
for each member of the set $\{\langle(\bar{e_j}, \bar{f_k}), (\bar{e_l}, \bar{f_m})\rangle \ s.t. \ (\bar{e_j}, \bar{f_k}) \in B \wedge (\bar{e_l}, \bar{f_m}) \in B \wedge \bar{f_k} = \bar{f_m}\}$ **do**
    $M \leftarrow M \ \cup \{(\bar{e_j}, \bar{e_l})\}$
    Compute $p(\bar{e_j}|\bar{e_l}) = \sum_{\bar{f}} p(\bar{e_j}|\bar{f_m}) p(\bar{f_m}|\bar{e_l})$
end for
Output M containing paraphrastic pairs and associated probabilities

---

(or word) alignment techniques.

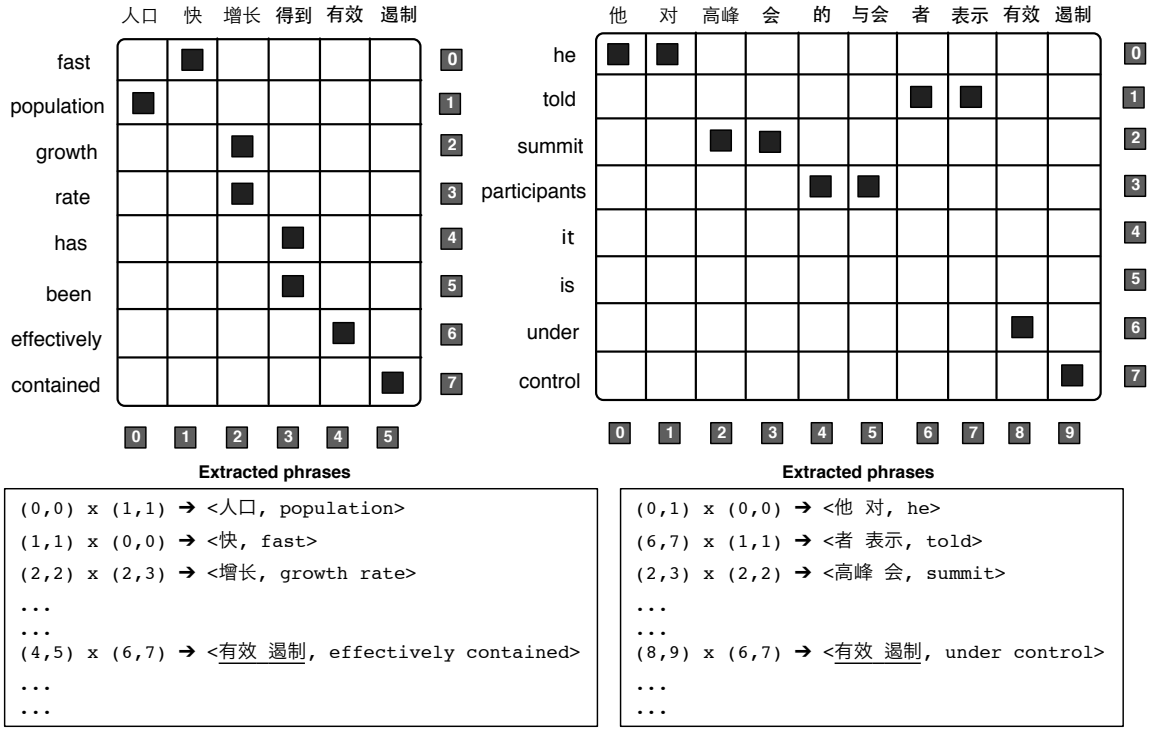


Figure 2.8: Extracting consistent bilingual phrasal correspondences from the shown sentence pairs. $(i_1, j_1) \times (i_2, j_2)$ denotes the correspondence $\langle f_{i_1} \dots f_{j_1}, e_{i_2} \dots e_{j_2} \rangle$. Not all extracted correspondences are shown.

One of the most popular methods leveraging bilingual parallel corpora is that proposed by Bannard and Callison-Burch (2005). This technique operates exactly as described above by attempting to infer semantic equivalence between phrases in the same language indirectly with the second language as a bridge. Their approach builds on one of the initial steps used to train a phrase-based statistical machine translation system (Koehn et al., 2003). Such systems rely on *phrase tables*—a tabulation of correspondences between phrases in the source language and phrases in the target language. These tables are usually extracted by inducing word alignments between sentence pairs in a training corpus and then incrementally building longer phrasal correspondences from individual words and shorter phrases. Once

such a tabulation of bilingual phrasal correspondences is available, correspondences between phrases in one language may be inferred simply by using the phrases in the other language as pivots.

Algorithm 5 shows how monolingual phrasal correspondences are extracted from a bilingual corpus C by using word alignments. Steps 3–7 extract bilingual phrasal correspondences from each sentence pair in the corpus by using heuristically induced bidirectional word alignments. Figure 2.8 illustrates this extraction process for two example sentence pairs. For each pair, a matrix shows the alignment between the Chinese and the English words. Element (i, j) of the matrix is filled if there is an alignment link between the i^{th} Chinese word and the j^{th} English word e_j. All phrase pairs *consistent* with the word alignment are then extracted. A consistent phrase pair can intuitively be thought of as a sub-matrix where all alignment points for its rows and columns are inside it and never outside. Next, steps 8–11 take all English phrases that all correspond to the same foreign phrase and infer them all to be paraphrases of each other.[11] For example, the English paraphrase pair ⟨*effectively contained, under control*⟩ is obtained from Figure 2.8 by pivoting on the Chinese phrase 有效 遏制, shown underlined for both matrices.

Using the components of a phrase-based SMT system also makes it easy to assign a probability value to any of the inferred paraphrase pairs as follows:

$$p(\bar{e_j}|\bar{e_k}) = \sum_{\bar{f}} p(\bar{e_j}, \bar{f}|\bar{e_k}) \approx \sum_{\bar{f}} p(\bar{e_j}|\bar{f}) p(\bar{f}|\bar{e_k})$$

[11] Note that it would have been equally easy to pivot on the English side and generate paraphrases in the foreign language instead.

where both $p(\bar{e_j}|\bar{f})$ and $p(\bar{f}|\bar{e_k})$ can both be computed using maximum likelihood estimation as part of the bilingual phrasal extraction process, e.g.,

$$p(\bar{e_j}|\bar{f}) = \frac{\text{number of times } \bar{f} \text{ is extracted with } \bar{e_j}}{\text{number of times } \bar{f} \text{ is extracted with any } \bar{e}}$$

Once the probability values are obtained, the most likely paraphrase can be chosen for any phrase.

Bannard and Callison-Burch are able to extract millions of phrasal paraphrases from a bilingual parallel corpus. Such an approach is able to capture a large variety of paraphrastic phenomena in the inferred paraphrase pairs but is seriously limited by the bilingual word alignment technique. Even state-of-the-art alignment methods from SMT are known to be notoriously unreliable when using them for aligning phrase pairs. The authors find via manual evaluation that the quality of the phrasal paraphrases obtained via manually constructed word alignments is *significantly* better than that of the paraphrases obtained from automatic alignments.

It has been widely reported that the existing bilingual word alignment techniques are not ideal for use in translation and, furthermore, improving these techniques does not always lead to an improvement in translation performance. (Callison-Burch et al., 2004; Ayan and Dorr, 2006; Lopez and Resnik, 2006; Fraser and Marcu, 2007). More details on the relationship between word alignment and SMT can be found in the comprehensive SMT survey recently published by Lopez (2008) (particularly Section 4.2). Paraphrasing done via bilingual corpora relies on the word alignments in the same way as a translation system would and, therefore, would

be equally be susceptible to the shortcomings of the word alignment techniques. To determine how noisy automatic word alignments affect paraphrasing done via bilingual corpora, a sample of paraphrase pairs were inspected for this thesis. These were extracted when using Arabic—a language significantly different from English—as the pivot language.[12] This study found that the paraphrase pairs in the sample set could be grouped into the following three broad categories:

(a) **Morphological variants**. These pairs only differ in the morphological form for one of the words in the phrases and cannot really be considered paraphrases. Examples: <`ten ton`, `ten tons`>, <`caused clouds`, `causing clouds`>.

(b) **Approximate Phrasal Paraphrases**. These are pairs that only shared partial semantic content. Most paraphrases extracted by the pivot method using automatic alignments fall into this category. Examples: <`were exiled`, `went abroad`>, <`accounting firms`, `auditing firms`>.

(c) **Phrasal Paraphrases**. Despite unreliable alignments, there were indeed a large number of truly paraphrastic pairs in the set that are semantically equivalent. Examples: <`army roadblock`, `military barrier`>, <`staff walked out`, `team withdrew`>.

Besides there being obvious linguistic differences between Arabic and English, the primary reason for the generation of phrase pairs that lie in categories (a) and

[12] The bilingual Arabic-English phrases were extracted from a million sentences of Arabic newswire data using the freely available and open source Moses SMT toolkit (`http://www.statmt.org/moses/`). The default Moses parameters were used. The English paraphrases were generated by simply applying the pivoting process described above to the bilingual phrase pairs.

(b) is due to incorrectly induced alignment between the English and Arabic words, and hence, phrases. Therefore, most subsequent work on paraphrasing using bilingual corpora, as discussed below, usually focuses on using additional machinery or evidence to cope with the noisy alignment process. Before continuing, it would be useful to draw a connection between Bannard and Callison-Burch's work and that of Wu and Zhou (2003) as discussed in Section 2.4.2. Note that both of these techniques rely on a secondary language to provide the cues for generating paraphrases in the primary language. However, Wu and Zhou rely on a pre-compiled bilingual dictionary to discover these cues whereas Bannard and Callison-Burch have an entirely data-driven discovery process.

In an attempt to address some of the noisy alignment issues, Callison-Burch (2008) recently proposed an improvement that places an additional syntactic constraint on the phrasal paraphrases extracted via the pivot-based method from bilingual corpora and showed that using such a constraint leads to a significant improvement in the quality of the extracted paraphrases.[13] The syntactic constraint requires that the extracted paraphrase be of the same syntactic type as the original phrase. With this constraint, estimating the paraphrase probability now requires the incorporation of syntactic type into the equation:

$$p(\bar{e_j}|\bar{e_k}, s(e_k)) \approx \sum_{\bar{f}} p(\bar{e_j}|\bar{f}, s(e_k))p(\bar{f}|\bar{e_k}, s(e_k))$$

where $s(e)$ denotes the syntactic type of the English phrase e. As before, maximum

[13]The software for generating these phrasal paraphrases along with a large collection of already extracted paraphrases is available at: `http://www.cs.jhu.edu/~ccb/howto-extract-paraphrases.html`

likelihood estimation is employed to compute the two component probabilities, e.g.,

$$p(\bar{e_j}|\bar{f}, s(e_k)) = \frac{\text{number of times } \bar{f} \text{ is extracted with } \bar{e_j} \text{ and type } s(e_k)}{\text{number of times } \bar{f} \text{ is extracted with any } \bar{e} \text{ and type } s(e_k)}$$

If the syntactic types are restricted to be simple constituents (NP, VP etc.), then using this constraint will actually exclude some of the paraphrase pairs that could have been extracted in the unconstrained approach. This leads to the familiar precision-recall tradeoff: it only extracts paraphrases that are of higher quality but the approach has a significantly lower coverage of paraphrastic phenomena that are not necessarily syntactically motivated. To increase the coverage, complex syntactic types such as those used in Combinatory Categorial Grammars (Steedman, 1996) are employed, which can help denote a syntactic constituent with children missing on the left and/or right hand sides. An example would be the complex type VP/(NP/NNS) which denotes a verb phrase missing a noun phrase to its right which, in turn, is missing a plural noun to its right. The primary benefit of using complex types is that less useful paraphrastic phrase pairs from different syntactic categories such as <*accurately*, *precise*>, that would have been allowed in the unconstrained pivot-based approach, are now disallowed.

The biggest advantage of this approach is the use of syntactic knowledge as one form of additional evidence in order to filter out phrase pairs from categories (a) and (b) as defined in the context of the manual inspection of pivot-based paraphrases above. Indeed, the authors conduct a manual evaluation to show that the syntactically constrained paraphrase pairs are indeed better than the those pro-

duced without such constraints. However, there are two additional benefits of this technique:

1. The constrained approach might allow induction of some *new* phrasal paraphrases in category (c) since now an English phrase only has to compete with other pivoted phrases of similar syntactic type and not all of them.

2. The effective partitioning of the probability space for a given paraphrase pair by syntactic types can be exploited: overly specific syntactic types that occur very rarely can be ignored and a less noisy paraphrase probability estimate can be computed, which may prove more useful in a downstream application than its counterpart computed via the unconstrained approach.

Note that requiring syntactic constraints for pivot-based paraphrase extraction restricts the approach to those languages where a reasonably good parser is available.

Kok and Brockett (2010) present a novel take on generating phrasal paraphrases with bilingual corpora. As with most approaches based on parallel corpora, they also start with phrase tables extracted from such corpora along with the corresponding phrasal translation probabilities. However, instead of performing the usual pivoting step with the bilingual phrases in the table, they take a graphical approach and represent each phrase in the table as a node leading to a bipartite graph. Two nodes in the graph are connected to each other if they are aligned to each other. Now, in order to extract paraphrases, they simply follow a path of even length from any English node to another. Note that the traditional pivot step is directly equivalent to following a path of length two: one English phrase to the

foreign pivot phrase and then to the potentially paraphrastic English phrase. By allowing paths of lengths longer than two, this graphical approach can find more paraphrases for any given English phrase.

Furthermore, instead of restricting themselves to a single bilingual phrase table, they take as input a number of phrase tables, each corresponding a different pair of 6 languages. Similar to the single table case, each phrase in each table is represented as a node in graph that is no longer bipartite in nature. By allowing edges to exist between nodes of *all* the languages if they are aligned, the pivot can now even be a set of nodes rather than a single node in another language. For example, one could easily find the following path in such a graph:

ate lunch → *aßen zu ittag* (German) → *aten een hapje* (Dutch) → *had a bite*

Each edge is associated with a weight corresponding to the bilingual phrase translation probability. Then random walks are sampled from the graph such that only paths of high probability end up contributing to the extracted paraphrases.

Obviously, the alignment errors discussed in the context of simple pivoting will also have an adverse effect on this approach. In order to prevent this, the authors add special *feature* nodes to the graph in addition to regular nodes. These feature nodes represent domain-specific knowledge of what would make good paraphrases. For example, nodes representing syntactic equivalence classes of the start and end words of the English phrases are added. This indicates that phrases that start and end with the same kind of words (interrogatives or articles) are likely to be paraphrases.

The authors extract paraphrases for a small set of input English paraphrases

and show that they are able to generate a larger percentage of correct paraphrases compared to the syntactically constrained approach proposed by Callison-Burch (2008). They conduct no formal evaluation of the coverage of their approach but show that in a limited setting, it is higher than the syntactically constrained approach. However, they do perform no comparisons of their coverage with the original pivot-based approach (Bannard and Callison-Burch, 2005). Astute readers will make the following two observations about the syntactic feature nodes used by the authors:

- Such nodes can be seen as an indirect way of incorporating a limited form of distributional similarity.
- By including such nodes essentially based on lexical equivalence classes, the authors are, in a way, imposing weaker forms of syntactic constraints that (Callison-Burch, 2008) uses but without requiring a parser.

2.5 Building Paraphrase Corpora

This section examines recent work on constructing paraphrase corpora, in preparation for a later discussion about techniques for evaluating paraphrase generation (Section 2.6 below). As part of this work, human subjects are generally asked to judge whether two given sentences are paraphrases of each other. A detailed examination of this manual evaluation task illuminates the nature of paraphrase in a practical, rather than a theoretical, context. In addition, it has obvious implica-

tions for any method, whether manual or automatic, that is used to evaluate the performance of a paraphrase generator.

Dolan and Brockett (2005) were the first to attempt to build a paraphrase corpus on a large scale. The Microsoft Research Paraphrase (MSRP) Corpus is a collection of 5801 sentence pairs, each manually labeled with a binary judgment as to whether it constitutes a paraphrase or not. As a first step, the corpus was created using a heuristic extraction method in conjunction with an SVM-based classifier that was trained to select likely sentential paraphrases from a large monolingual corpus containing news article clusters. However, the more interesting part of the task was the subsequent evaluation of these extracted sentence pairs by human annotators and the set of issues encountered when defining the evaluation guidelines for these annotators.

It was observed that if the human annotators were instructed to mark only the sentence pairs that were strictly semantically equivalent or that exhibited bidirectional entailment as paraphrases, then the results were limited to uninteresting sentence pairs such as the following:

S_1: *The euro rose above US$1.18, the highest price since its January 1999 launch.*

S_2: *The euro rose above $1.18 the highest level since its launch in January 1999.*

S_1: *However, without a carefully controlled study, there was little clear proof that the operation actually improves people's lives.*

S_2: *But without a carefully controlled study, there was little clear proof that the operation improves people's lives.*

Instead, they discovered that most of the complex paraphrases—ones with alternations more interesting than simple lexical synonymy and local syntactic changes—exhibited varying degrees of semantic divergence, e.g.,

S_1: *Charles O. Prince, 53, was named as Mr. Weill's successor.*

S_2: *Mr. Weill's longtime confidant, Charles O. Prince, 53, was named as his successor.*

S_1: *David Gest has sued his estranged wife Liza Minelli for beating him when she was drunk.*

S_2: *Liza Minellis estranged husband is taking her to court after saying she threw a lamp at him and beat him in drunken rages.*

Therefore, in order to be able to create a richer paraphrase corpus, one with many complex alternations, the instructions to the annotators had to be relaxed; the degree of mismatch accepted before a sentence pair was judged to be fully semantically divergent (or "non-equivalent") was left to the human subjects. It is also reported that, given the idiosyncratic nature of each sentence pair, only a few formal guidelines were generalizable enough to take precedence over the subjective judgments of the human annotators. Despite the somewhat loosely defined guidelines, the inter-annotator agreement for the task was 84%. However, a Kappa score of 62 indicated that the task was overall a difficult one (Cohen, 1960). At the end, 67%

of the sentence pairs were judged to be paraphrases of each other and the rest were judged to be non-equivalent.[14]

While the MSRP Corpus is a valuable resource and its creation provided valuable insight into what constitutes a paraphrase in the practical sense, it does have some shortcomings. For example, one of the heuristics used in the extraction process was that the two sentences in a pair must share at least three words. Using this constraint rules out any paraphrase pairs that are fully lexically divergent but still semantically equivalent. The small size of the corpus, when combined with this and other such constraints, precludes the use of the corpus as training data for a paraphrase generation or extraction system. However, it is fairly useful as a freely available test set to evaluate paraphrase recognition methods.

On a related note, Fujita and Inui (2005) take a more knowledge-intensive approach to building a Japanese corpus containing sentence pairs with binary paraphrase judgments and attempt to focus on variety and on minimizing the human annotation cost. The corpus contains 2031 sentence pairs each with a human judgment indicating whether the paraphrase is correct or not. To build the corpus, they first stipulate a typology of paraphrastic phenomena (e.g., rewriting light-verb constructions) and then manually create a set of morpho-syntactic paraphrasing rules and patterns describing each type of paraphrasing phenomenon. A paraphrase generation system (Fujita et al., 2004) is then applied to a corpus containing Japanese news articles and example paraphrases are generated for the sentences in the corpus.

[14]The MSR paraphrase corpus is available at: `http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/`.

These paraphrase pairs are then handed to two human annotators who create binary judgments for each pair indicating whether or not the paraphrase is correct. Using a class-oriented approach is claimed to have a two-fold advantage:

1. **Exhaustive Collection of Paraphrases**. Creating specific paraphrasing rules for each class manually is likely to increase the chance of the collected examples accurately reflecting the distribution of occurrences in the real world.

2. **Low Annotation Cost**. Partitioning the annotation task into classes is expected to make it easier (and faster) to arrive at a binary judgment given that an annotator is only concerned with a specific type of paraphrasing when creating said judgment.

The biggest disadvantage of this approach is that only two types of paraphrastic phenomena are used: light-verb constructions and transitivity alternations (using intransitive verbs in place of transitive verbs). The corpus indeed captures almost all examples of both types of paraphrastic phenomena and any that are absent can be easily covered by adding one or two more patterns to the class. The claim of reduced annotation cost is not necessarily borne out by the observations. Despite partitioning the annotation task by types, it was still difficult to provide accurate annotation guidelines. This led to a significant difference in annotation time—with some annotations taking almost twice as long as others. Given the small size of the corpus, it is unlikely that it may be used as training data for corpus-based paraphrase generation methods and, like the MSRP corpus, would be best suited to evaluate paraphrase recognition techniques.

Cohn et al. (2008) describe a different take on the creation of a monolingual parallel corpus containing 900 sentence pairs with paraphrase annotations that can be used for both development and evaluation of paraphrase systems. These paraphrase annotations take the form of alignments between the words and sequences of words in each sentence pair; these alignments are analogous to the word- and phrasal-alignments induced in Statistical Machine Translation (SMT) systems that were illustrated in Section 2.4.5. As is the case with SMT alignments, the paraphrase annotations can be of different forms: one-word-to-one-word, one-word-to-many-words as well as fully phrasal alignments.[15]

The authors start from a sentence-aligned paraphrase corpus compiled from three corpora that were already described in Sections 2.4.3 and 2.5: (a) The sentence pairs judged equivalent from the MSRP Corpus; (b) The Multiple Translation Chinese (MTC) corpus of multiple human-written translations of Chinese news stories used by Pang et al. (2003); and (c) Two English translations of the French novel *Twenty Thousand Leagues Under The Sea*, a subset of the monolingual parallel corpus used by Barzilay and McKeown (2001). The words in each sentence pair from this corpus are then aligned automatically to produce the initial paraphrase annotations that are then refined by two human annotators. The annotation guidelines required that the annotators judge which parts of a given sentence pair were in *correspondence* and to indicate this by creating an alignment between those parts (or correcting already existing alignments, if present). Two parts were said to

[15]The paraphrase-annotated corpus can be found at: `http://www.dcs.shef.ac.uk/~tcohn/paraphrase_corpus.html`

correspond if they could be substituted for each other within the specific context provided by the respective sentence pair. In addition, the annotators were instructed to classify the created alignments as either *sure* (the two parts are clearly substitutable) or *possible* (the two parts are slightly divergent either in terms of syntax or semantics). For example, given the following paraphrastic sentence pair:

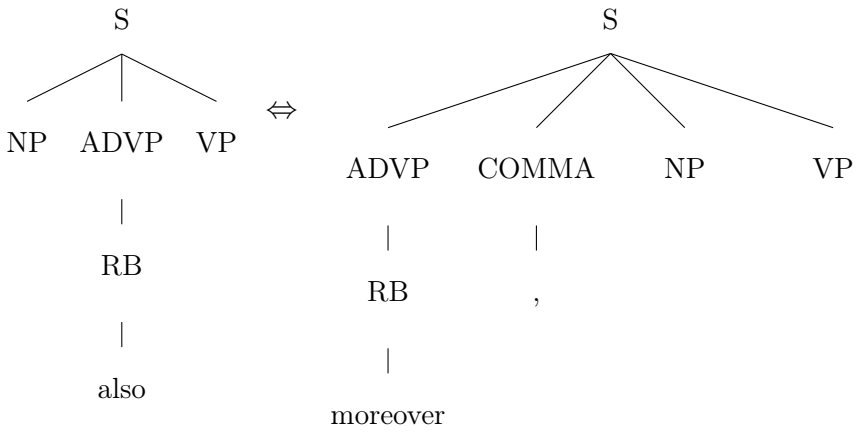
S_1: *He stated the convention was of profound significance.*

S_2: *He said that the meeting could have very long term effects.*

the phrase pair ⟨*the convention*, *the meeting*⟩ will be aligned as a sure correspondence whereas the phrase pair ⟨*was of profound significance*, *could have very long term effects*⟩ will be aligned as a possible correspondence. Other examples of possible correspondences could include the same stem expressed as different parts-of-speech (such as ⟨*significance*, *significantly*⟩) or two non-synonymous verbs (such as ⟨*this is also*, *this also marks*⟩). For more details on the alignment guidelines that were provided to the annotators, the reader is referred to (Callison-Burch et al., 2006a).

Extensive experiments are conducted to measure inter-annotator agreements and obtain good agreement values but that are still low enough to confirm that it is difficult for humans to recognize paraphrases even when the task is formulated differently. Overall, such a paraphrase corpus with detailed paraphrase annotations is much more informative than a corpus containing binary judgments at the sentence level such as the MSRP Corpus. As an example, since the corpus contains paraphrase annotations at the word as well as phrasal levels, it can be used to build systems that can learn from these annotations and generate not only fully lexi-

calized phrasal paraphrases but also syntactically motivated paraphrastic patterns. To demonstrate the viability of the corpus for this purpose, a grammar induction algorithm (Cohn and Lapata, 2007) is applied—originally developed for sentence compression—to the parsed version of their paraphrase corpus and show that they can learn paraphrastic patterns such as:



Most recently, research has been done on using Amazon Mechanical Turk to collect either human-authored phrasal paraphrases (Buzek et al., 2010) or human judgments for automatically generated paraphrases (Denkowski et al., 2010). Both approaches produce these resources not for the sake of building paraphrase corpora but rather only as a step towards improved statistical machine translation systems. However, there is no reason why the same resources could not be used to develop and evaluate automatic paraphrase systems.

In general, building paraphrase corpora, whether it be done at the sentence level or at the sub-sentential level, is extremely useful to foster further research and development in the area of paraphrase generation.

2.6 Evaluation of Paraphrase Generation

While other language processing tasks such as Machine Translation and Document Summarization usually have multiple annual community-wide evaluations using standard test sets and manual as well as automated metrics, the task of automated paraphrasing does not. An obvious reason behind this disparity could be that paraphrasing is not an application in and of itself. However, the existence of similar evaluations for other tasks that are not applications, such as dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007) and word sense disambiguation (Senseval), suggests otherwise. The view adopted in this thesis is that, over the years, paraphrasing has been employed in an extremely fragmented fashion. Paraphrase extraction and generation are used in different forms and with different names in the context of different applications (e.g., synonymous collocation extraction, query expansion). This usage pattern does not allow researchers in one community to share the lessons learned with those from other communities. In fact, it may even lead to research being duplicated across communities.

However, more recent work—some of it discussed in this chapter—on extracting phrasal paraphrases (or patterns) does include direct evaluation of the paraphrasing itself: the original phrase and its paraphrase are presented to multiple human judges, along with the contexts in which the phrase occurs in the original sentence, who are asked to determine whether the relationship between the two phrases is indeed paraphrastic (Barzilay and McKeown, 2001; Barzilay and Lee, 2003; Ibrahim et al., 2003; Pang et al., 2003). A more direct approach is to substi-

tute the paraphrase in place of the original phrase in its sentence and present both sentences to judges who are then asked to judge not only their semantic equivalence but also the grammaticality of the new sentence (Bannard and Callison-Burch, 2005; Callison-Burch, 2008). Motivation for such substitution-based evaluation is discussed in (Callison-Burch, 2007): the basic idea being that items deemed to be paraphrases may behave so in only some contexts and not others. Szpektor et al. (2007) posit a similar form of evaluation for textual entailment wherein the human judges are not only presented with the entailment rule but also with a sample of sentences that match its left hand side (called *instances*), and then asked to assess whether the rule holds under each specific instance.

Sentential paraphrases may be evaluated in a similar fashion without the need for any surrounding context (Quirk et al., 2004). An intrinsic evaluation of this form must employ the usual methods for avoiding any bias and for maximizing inter-judge agreement. In addition, given the difficulty of task even for human annotators, adherence to strict semantic equivalence may not always be a suitable guideline and intrinsic evaluations must be very carefully designed. In contrast, a number of these techniques also perform extrinsic evaluations, in addition to the intrinsic one, by utilizing the extracted or generated paraphrases to improve other applications such as machine translation (Callison-Burch et al., 2006b) and others as described in in the beginning of this chapter.

Another option when evaluating the quality of a paraphrase generation method is that of using automatic measures. The traditional automatic evaluation measures of precision and recall are not particularly suited to this task because in order to

use them, a list of reference paraphrases has to be constructed against which these measures may be computed. Given that it is extremely unlikely that any such list will be exhaustive, any precision and recall measurements will not be accurate. Therefore, other alternatives are needed. Since the evaluation of paraphrase is essentially the task of measuring semantic similarity or of paraphrase recognition, all of those metrics, including the ones discussed in Section 2.3, can be employed here.

Most recently, Callison-Burch et al. (2008a) discuss ParaMetric, another automatic measure that may be used to evaluate paraphrase extraction methods. This work follows directly from the work done by the authors to create the paraphrase-annotated corpus as described in the previous section. Recall that this corpus contains paraphrastic sentence pairs with annotations in the form of alignments between their respective words and phrases. It is posited that to evaluate any paraphrase generation method, one could simply have it produce its own set of alignments for the sentence pairs in the corpus and precision and recall could then be computed over alignments instead of phrase pairs. These alignment-oriented precision ($\mathrm{P}_{\mathrm{align}}$) and recall ($\mathrm{R}_{\mathrm{align}}$) measures are computed as follows:

$$\mathrm{P}_{\mathrm{align}} = \frac{\sum_{\langle s_1,s_2\rangle} |N_P(s_1,s_2) \cap N_M(s_1,s_2)|}{\sum_{\langle s_1,s_2\rangle} |N_P(s_1,s_2)|}$$

$$\mathrm{R}_{\mathrm{align}} = \frac{\sum_{\langle s_1,s_2\rangle} |N_P(s_1,s_2) \cap N_M(s_1,s_2)|}{\sum_{\langle s_1,s_2\rangle} |N_M(s_1,s_2)|}$$

where $\langle s_1, s_2\rangle$ denotes a sentence pair, $N_M(s_1, s_2)$ denotes the phrases extracted

via the manual alignments for the pair $\langle s_1, s_2 \rangle$ and $N_P(s_1, s_2)$ denotes the phrases extracted via the automatic alignments induced using the paraphrase method P that is to be evaluated. The phrase extraction heuristic used to compute N_P and N_M from the respective alignments is the same as that employed by (Bannard and Callison-Burch, 2005) and illustrated in Figure 2.8.

While using alignments as the basis for computing precision and recall is a clever trick, it does require that the paraphrase generation method be capable of producing alignments between sentence pairs. For example,the methods proposed by Pang et al. (2003) and Quirk et al. (2004) for generating sentential paraphrases from monolingual parallel corpora and described in Section 2.4.3, do produce alignments as part of their respective algorithms. Indeed, Callison-Burch et al. provide a comparison of their pivot-based approach—operating on bilingual parallel corpora—with the two monolingual approaches just mentioned in terms of ParaMetric since all three methods are capable of producing alignments.

However, for other approaches that do not necessarily operate at the level of sentences and cannot produce any alignments, falling back on estimates of traditional formulations of precision and recall is suggested, computed as explained above.

There has also been some preliminary progress toward using standardized test sets for the intrinsic evaluations. A test set containing 20 AFP articles (484 sentences) about violence in the Middle East that was used for evaluating the lattice-based paraphrase technique in (Barzilay and Lee, 2003) has been made freely avail-

able.[16] In addition to the original sentences for which the paraphrases were generated, the set also contains the paraphrases themselves and the judgments assigned by human judges to these paraphrases. The paraphrase-annotated corpus discussed in the previous section would also fall under this category of resources.

As with many other fields in NLP, paraphrase generation also lacks serious extrinsic evaluation (Belz, 2009). As described above, many paraphrase generation techniques are developed in the context of a host NLP application and this application usually serves as one form of extrinsic evaluation for the quality of the paraphrases generated by that technique. However, as yet there is no widely agreed upon method of extrinsically evaluating paraphrase generation. Addressing this deficiency should be a crucial consideration for any future community-wide evaluation effort.

An important dimension for any area of research is the availability of forums where members of the community may share their ideas with their colleagues and receive valuable feedback. In recent years, a number of such forums been made available to the automatic paraphrasing community (Inui and Hermjakob, 2003; Tanaka et al., 2004; Dras and Yamamoto, 2005; Sekine et al., 2007) which represent an extremely important step toward countering the fragmented usage pattern described above.

[16]The corpus is available at `http://www.cs.cornell.edu/Info/Projects/NLP/statpar.html`.

2.7 Future Trends

This section looks to the future of paraphrasing and examines general trends for the corresponding research methods. Several such trends are identified in the area of paraphrase generation that are gathering momentum.

The Influence of the Web. The web is rapidly becoming one of the most important sources of data for natural language processing applications, which should not be surprising given the phenomenal rate of growth. The (relatively) freely available web data, massive in scale, has already had a definite influence over data-intensive techniques such as those employed for paraphrase generation (Paşca and Dienes, 2005). However, availability of such massive amounts of web data comes with serious concerns for efficiency and has led to the development of efficient methods that can cope with such large amounts of data. Bhagat and Ravichandran (2008) extract phrasal paraphrases by measuring distributional similarities over a 150GB monolingual corpus (25 billion words) via *locality sensitive hashing*, a randomized algorithm that involves the creation of "fingerprints" for vectors in space (Broder, 1997). Since vectors that are more similar are more likely to have similar fingerprints, vectors (or distributions) can simply be compared by comparing their fingerprints leading to a more efficient distributional similarity algorithm (Charikar, 2002; Ravichandran et al., 2005). The view adopted in this thesis is that the influence of the web will extend to other avenues of paraphrase generation. For example, Fujita and Sato (2008a) propose evaluating phrasal paraphrase pairs, automatically generated from a monolingual corpus, by querying the web for snippets related to

the pairs and using them as features to compute the pair's "paraphrasability".

Combining Multiple Sources of Information. Another important trend in paraphrase generation is that of leveraging multiple sources of information to determine whether two units are paraphrastic. For example, Zhao et al. (2008) improve the sentential paraphrases that can be generated via the pivot method by leveraging five other sources in addition to the bilingual parallel corpus itself: (1) a corpus of web queries similar to the phrase, (2) definitions from the Encarta dictionary, (3) a monolingual parallel corpus, (4) a monolingual comparable corpus, and (5) an automatically constructed thesaurus. Phrasal paraphrase pairs are extracted separately from all six models and then combined in a log-linear paraphrasing-as-translation model and that is described in this thesis in subsequent chapters (Madnani et al., 2007). A manual inspection reveals that using multiple sources of information yields paraphrases with much higher accuracy. Such exploitation of multiple types of resources and their combination is an important development. Zhao et al. (2009) further increase the utility of this combination approach by incorporating application specific constraints on the pivoted paraphrases, e.g., if the output paraphrases need to be simplified versions of the input sentences, then only those phrasal paraphrase pairs are used where the output is shorter than the input.

Use of SMT Machinery. In theory, statistical machine translation is very closely related to paraphrase generation since it also relies on finding semantic equivalence, albeit in a second language. Hence, there have been numerous paraphrasing approaches that have relied on different components of an SMT pipeline (word alignment, phrase extraction, decoding/search) as seen above in Section 2.4.5. Despite

the obvious convenience of using SMT components for the purpose of "monolingual translation," doing so usually requires additional work to deal with the added noise due to the nature of such components. This thesis adopts the view that that SMT research will continue to influence research in paraphrasing; both by providing ready-to-use building blocks and by necessitating development of methods to effectively use such blocks for the unintended task of paraphrase generation.

Domain-specific Paraphrasing. Recently, work has been done to generate phrasal paraphrases in specialized domains. For example, in the field of health literacy, it is well known that documents targeted at health consumers are not very well-targeted to their purported audience. Recent research has shown how to generate a lexicon of semantically equivalent phrasal (and lexical) pairs of technical and lay medical terms from monolingual parallel corpora (Elhadad and Sutaria, 2007) as well as monolingual comparable corpora (Deléger and Zweigenbaum, 2009). Examples include pairs such as <`myocardial infarction`, `heart attack`>and <`leucospermia`, `increased white cells in the sperm`>. In another domain, Max (2008) proposes an adaptation of the pivot-based method to generate *rephrasings* of short text spans that could help a writer revise a text. Since the goal is to assist a writer in making revisions, the rephrasings do not always need to bear a perfect paraphrastic relationship to the original; a scenario suited for the pivot-based method. Several variants of such adaptations are developed that generate candidate rephrasings driven by fluency, semantic equivalence and authoring value respectively.

The view adopted in this thesis is that a large-scale annual community-wide evaluation is necessary to foster further research in, and use of, paraphrase extraction

and generation. While there have been recent workshops and tasks on paraphrasing and entailment as discussed in Section 2.6, this evaluation would be much more focused, providing sets of shared guidelines and resources, in the spirit of the recent NIST MT Evaluation Workshops (NIST, 2008).

2.8 Summary

Over the last two decades, there has been a lot of research on paraphrase extraction and generation within every research community in natural language processing in order to improve the specific application with which that community is concerned. However, a large portion of this research can be easily adapted for more widespread use outside the particular community and can provide significant benefits to the whole field. Only recently have there been serious efforts to conduct research on the topic of paraphrasing by treating it as an important natural language processing task independent of a host application.

This chapter has presented a comprehensive survey of paraphrase extraction and generation motivated by the fact that paraphrases can help in a multitude of applications such as machine translation, text summarization and information extraction. The aim was to provide an application-independent overview of paraphrase generation, while also conveying an appreciation for the importance and potential use of paraphrases in the field of NLP research. A large variety of paraphrase generation methods have been described, each with a very different set of characteristics, in terms of both its performance and its ease of deployment. While most of the

methods in this chapter can be used in multiple applications, the choice of the most appropriate method depends on how well the characteristics of the produced paraphrases match the requirements of the downstream application in which the paraphrases are being utilized.

The next chapter describes one of the primary components of the work done in this thesis: extending the work done by Bannard and Callison-Burch (2005)—presented in 2.4.5 above—to the sentence level and building a general sentential paraphrasing architecture that casts the problem of paraphrase generation as one of English-to-English translation that leverages a well-defined, extensible and entirely data-driven model.

3 The First 180 Degrees: Sentential Paraphrasing via SMT

———— * ————

Find another way
to say what you and I say.
Make the machine learn.
–Nitin Madnani

———— * ————

This chapter describes the design and implementation of a sentential paraphraser using SMT machinery. The statistical machine translation framework that is used to construct the paraphraser is first briefly presented. The induction of a monolingual translation model using this formalism is then shown. Two alternative methods are presented and compared that the induction process can use to compute the probabilities for the paraphrases. A few examples of the paraphrases that the system can generate by using parallel corpora with different foreign languages are also shown. Finally, a small-scale intrinsic evaluation of the paraphrases that are generated by this paraphraser is presented. It is observed from this evaluation that they are relatively noisy and not directly useful to humans in any way. In the next chapter, however, a large-scale extrinsic evaluation is conducted and it is found that these paraphrases prove to be extremely useful when employed appropriately in a bilingual SMT pipeline.

3.1 Background

The paraphrasing techniques described in this thesis makes use of a hierarchical SMT framework (Chiang et al., 2005; Chiang, 2007). Such a framework is formally based on a weighted synchronous context-free grammar (SCFG), containing synchronous rules of the form:

$$X \rightarrow \langle \bar{f}, \bar{e}, \phi_1^k) \rangle \tag{3.1}$$

where X is a symbol from the non-terminal alphabet, and $\bar{e}$ and $\bar{f}$ can contain both words (terminals) and variables (non-terminals) that serve as place-holders for other phrases. In the context of SMT, where phrase-based models are frequently used, these synchronous rules can be interpreted as pairs of *hierarchical phrases*. The underlying strength of a hierarchical phrase is that it allows for effective learning of not only the lexical re-orderings, but phrasal re-orderings as well. Each ϕ denotes a feature function defined on the pair of hierarchical phrases.[1] Several different kinds of feature functions can be used but some of the most common features are conditional and joint co-occurrence probabilities over the hierarchical paraphrase pair. Given a synchronous context-free grammar consisting of these translation rules, the actual translation process is simply equivalent to parsing with this grammar. Any given source sentence is parsed using the source side of the synchronous grammar. A target-language derivation is generated simultaneously via the target side of the same rules, and the yield of that hypothesized derivation represents the hypothesized

[1]Currently only one non-terminal symbol is used in a hierarchical phrase.

translation string in the target language.

The actual translation model defined over the set of synchronous derivations is a log-linear model of the form:

$$P(D) \propto \exp\left(\sum_i \lambda_i \phi_i(X \rightarrow \langle \bar{e}, \bar{f} \rangle)\right) \tag{3.2}$$

where each λ_i represents the weight for the respective feature ϕ_i.

The hierarchical translation framework allows learning grammars and feature values from parallel corpora, without requiring syntactic annotation of the data. Briefly, training such a model proceeds as follows:

- One-to-many word alignments for each sentence pair are induced on the parallel corpus in both directions (source $\rightarrow$ target and target $\rightarrow$ source) by using a tool like GIZA++ (Och and Ney, 2000). The two sets of word alignments are then combined into one set of many-to-many word alignments by using an alignment refinement heuristic.

- *Initial phrase pairs* are identified following the procedure typically employed in phrase-based systems (Koehn et al., 2003; Och and Ney, 2004). The extraction of such initial phrase pairs was discussed in Chapter 2 and an example shown in Figure 2.8.

- Grammar rules in the form of Equation 3.1 are induced by "subtracting" out hierarchical phrase pairs from these initial phrase pairs. This subtraction process is defined by a set of constraints that restrict the size of the extracted

grammar. The constraints are described in detail by Chiang (2007, Sec. 3.2).

- Fractional counts are assigned to each produced rule:

$$c(X \rightarrow \langle \bar{f}, \bar{e} \rangle) = \sum_{j=1}^{m} \frac{1}{n_{jr}} \tag{3.3}$$

  where m is the number of initial phrase pairs that give rise to this grammar rule and n_{jr} is the number of grammar rules produced by the j^{th} initial phrase pair.

- Feature functions ϕ_1^k are calculated for each rule using either the accumulated co-occurrence fraction counts from the previous step or other characteristics of the source and target sides of the rule.

Once a translation grammar has been extracted, an algorithm based on Powell's method of grid-based line minimization is used to find the optimal values for the parameters λ_i (Ostendorf et al., 1991; Och, 2003). The tuning algorithm is described in more detail in Chapter 4. Finally, with the feature weights (parameters) determined, the process of *decoding*—searching for the derivation with the highest model score—takes place using a CKY synchronous parser with beam search, augmented to permit efficient incorporation of language model scores (Chiang, 2007).

3.2 Induction of Monolingual Translation Model

The paraphraser described in this chapter approaches sentence-level paraphrasing as a problem of English-to-English translation, constructing the model

using English-F translation, for a second language F, as a pivot. Following Bannard and Callison-Burch (2005) as described in Section 2.4.5, first English-to-F correspondences are identified, and then English to English correspondences are obtained by following translation units from English to F and back. Then, generalizing this approach, those mappings are used to create a well defined English-to-English hierarchical translation model. The parameters of this model are tuned using a normal parameter optimization process, and then the model is used in an (unmodified) statistical MT system, yielding sentence-level English paraphrases by means of decoding input English sentences. The remainder of this section presents this process in detail.

3.2.1 Pivoting Bilingual Translation Rules

This thesis employs the same strategy as Bannard and Callison-Burch (2005) for the induction of the required monolingual translation grammar. First, the SMT system is trained in standard fashion on a bilingual English-F training corpus. Then, for each existing production in the resulting bilingual grammar, multiple new English-to-English productions are created by pivoting on the foreign hierarchical phrase in the rule. For example, assume that the following rules are present

in an English-Chinese bilingual grammar:

$$
\begin{aligned}
X &\rightarrow \langle X_1 \text{建} X_2, X_1 \text{ to build } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{建} X_2, X_1 \text{ to construct } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{建} X_2, X_1 \text{ to establish } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{建} X_2, X_1 \text{ to formulate } X_2 \rangle
\end{aligned}
$$

Note that the non-terminals representing the same sub-phrases on either side of the rule are co-indexed. If the Chinese hierarchical phrase X_1建X_2 is used as a pivot, then the following two distinct English-to-English rules can be extracted:

$$
\begin{aligned}
X &\rightarrow \langle X_1 \text{ to build } X_2, X_1 \text{ to construct } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{ to construct } X_2, X_1 \text{ to build } X_2 \rangle
\end{aligned}
$$

from the first pair of bilingual rules. In a similar fashion, pivoting on the same phrase for other bilingual rules will produce the following rules (the corresponding rules in the other direction are not shown):

$$
\begin{aligned}
X &\rightarrow \langle X_1 \text{ to build } X_2, X_1 \text{ to establish } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{ to build } X_2, X_1 \text{ to formulate } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{ to construct } X_2, X_1 \text{ to establish } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{ to construct } X_2, X_1 \text{ to formulate } X_2 \rangle \\
X &\rightarrow \langle X_1 \text{ to establish } X_2, X_1 \text{ to formulate } X_2 \rangle
\end{aligned}
$$

To limit noise during pivoting, only the top 25 source language pivots for any English hierarchical phrase are used, as determined by the bilingual fractional counts.

3.2.2 Feature Functions for Paraphrase Rules

Each synchronous rule production in the bilingual grammar is weighted by several feature values. The most commonly used features are usually probabilistic in nature and are computed from the fractional counts assigned to that rule during extraction. Examples of such probabilistic features include the maximum likelihood estimates of the conditional probabilities $p(\bar{f}|\bar{e})$ and $p(\bar{f}|\bar{e})$, where $\bar{f}$ and $\bar{e}$ are the source and target sides of the rule respectively. In order to perform accurate pivoting, these feature functions must be recomputed for the newly created English-to-English grammar. Two ways of computing these feature functions from the bilingual feature functions are used:

- **Derived Fractional Counts**. In this method, induced fractional counts are derived for the pivoted paraphrases from the fractional counts of the two participating bilingual phrases and then maximum likelihood estimates are computed for the probabilistic features from these induced counts. Calculating the proposed features is complicated by the fact that the counts for English-to-English rules don't exist; there is *no* English-to-English parallel corpus.

Instead, the fractional count for a monolingual rule is estimated as follows:

$$c(X \rightarrow \langle \bar{e_1}, \bar{e_2} \rangle) = \sum_{\bar{f}} c(X \rightarrow \langle \bar{f}, \bar{e_1} \rangle) * c(X \rightarrow \langle \bar{f}, \bar{e_2} \rangle) \tag{3.4}$$

An intuitive way to think about the formula above is by using an example at the corpus level. Assume that, in the given bilingual parallel corpus, there are m sentences in which the English phrase $\bar{e_1}$ co-occurs with the foreign phrase $\bar{f}$ and n sentences in which the same foreign phrase $\bar{f}$ co-occurs with the English phrase $\bar{e_2}$. The problem can then be thought of as defining a function $g(m, n)$ that computes the number of sentences in a hypothetical English-to-English parallel corpus wherein the phrases $\bar{e_1}$ and $\bar{e_1}$ co-occur. For this paper, $g(m, n)$ is defined to be the upper bound mn.

Given this definition, the probabilistic features for all pivoted rules in the English-to-English grammar can then be computed. First, the joint probability of the two English hierarchical paraphrases e_1 and e_2 can be calculated as:

$$p(\bar{e_1}, \bar{e_2}) = \frac{c(X \rightarrow \langle \bar{e_1}, \bar{e_2} \rangle)}{\sum_{\bar{e_1}', \bar{e_2}'} c(X \rightarrow \langle \bar{e_1}', \bar{e_2}' \rangle)} \tag{3.5}$$

$$= \frac{c(X \rightarrow \langle \bar{e_1}, \bar{e_2} \rangle)}{c(X)} \tag{3.6}$$

where the numerator is the fractional count of the rule under consideration and the denominator represents the marginal count over all the English hierarchical phrase pairs.

Next, the conditionals $p(\bar{e_1}|\bar{e_2})$ and $p(\bar{e_2}|\bar{e_1})$ can be defined as follows:

$$p(\bar{e_1}|\bar{e_2}) = \frac{c(X \rightarrow \langle \bar{e_1}, \bar{e_2} \rangle)}{\sum_{\bar{e_1}'} c(X \rightarrow \langle \bar{e_1}', \bar{e_2} \rangle)} \tag{3.7}$$

$$p(\bar{e_2}|\bar{e_1}) = \frac{c(X \rightarrow \langle \bar{e_1}, \bar{e_2} \rangle)}{\sum_{\bar{e_2}'} c(X \rightarrow \langle \bar{e_1}, \bar{e_2}' \rangle)} \tag{3.8}$$

- **Derived Features**. Here, instead of inducing fractional counts in the count space, probabilistic features for the monolingual rules are computed directly from the corresponding feature values for the bilingual rules. First, the conditionals $p(\bar{e_1}|\bar{e_2})$ and $p(\bar{e_2}|\bar{e_1})$:

$$\begin{aligned} p(\bar{e_1}|\bar{e_2}) &= \sum_{\bar{f}} p(\bar{e_1}|\bar{f}, \bar{e_2}) * p(\bar{f}|\bar{e_2}) && (3.9) \\ &\approx \sum_{\bar{f}} p(\bar{e_1}|\bar{f}) * p(\bar{f}|\bar{e_2}) && (3.10) \\ p(\bar{e_2}|\bar{e_1}) &= \sum_{\bar{f}} p(\bar{e_2}|\bar{f}, \bar{e_1}) * p(\bar{f}|\bar{e_1}) && (3.11) \\ &\approx \sum_{\bar{f}} p(\bar{e_2}|\bar{f}) * p(\bar{f}|\bar{e_1}) && (3.12) \end{aligned}$$

where the two constituent bilingual probabilities are the bilingual rule features. And now the joint probability of the two English hierarchical paraphrases e_1 and e_2 can be computed as:

$$p(\bar{e_1}, \bar{e_2}) = p(\bar{e_1}|\bar{e_2}) * p(\bar{e_2}) = p(\bar{e_2}|\bar{e_1}) * p(\bar{e_1}) \tag{3.13}$$

where the marginals $p(\bar{e_1})$ and $p(\bar{e_2})$ are computed from the bilingual fractional

counts.

Figure 3.1 shows a simple example which illustrates the difference between the two ways of computing the probabilistic feature functions for each paraphrase rule. Note that both way of computing the paraphrase probability features are approximations: the derived-counts method computes maximum likelihood estimates from counts induced based on a hypothetical English-to-English parallel corpus; the derived-features method is not actually estimating the actual paraphrase likelihood but rather the relative probability of *different* paraphrases of the same phrase, under the assumption that all of the alignments in the bilingual parallel data are correct. In Section 3.2.4, some experiments are described that were conducted to determine which of the feature computation methods produces better paraphrases.

For all induced rules, a feature given by $exp(-T(\bar{e_2}))$ is also calculated, where $T(\bar{e_2})$ just counts the number of terminal symbols in $\bar{e_2}$. This feature allows the monolingual translation model to learn whether it should produce shorter or longer paraphrases. In addition to the above features that are estimated from the training data, a trigram language model is also used. Since the production of English sentences is done via regular decoding, the same language model that is employed in a standard SMT setting can be and is used here.

3.2.3 Tuning Model Parameters

Since the sentential paraphraser is based on an English-to-English log-linear translation model, it also requires its own tuning of feature weights just as the bilin-

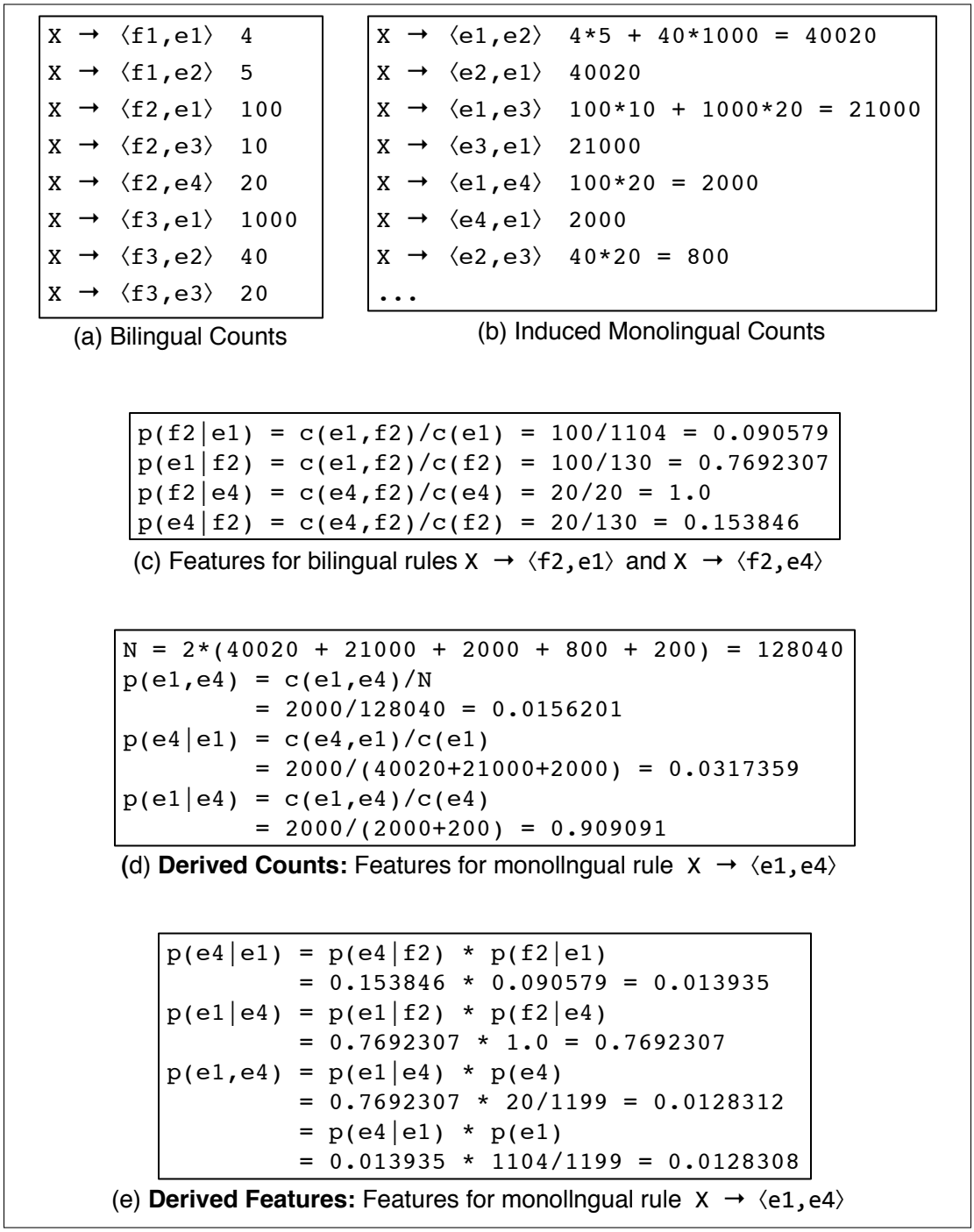


Figure 3.1: A toy example illustrating the two methods of computing features for pivoted monolingual translation or paraphrase rules. (a) the bilingual translation rules that are extracted from the parallel corpus along with associated fractional counts. For simplicity, $\bar{f}_1$ is denoted as `f1` and so on; (b) these fractional counts may be converted into counts for the various pivoted paraphrase rules; (c) the probabilistic features for two of the bilingual rules $X \rightarrow \langle \bar{f}_2, \bar{e_1} \rangle$ and $X \rightarrow \langle \bar{f}_2, \bar{e_4} \rangle$; (d) the **derived-counts** method of computing the feature values for the paraphrase rule $X \rightarrow \langle \bar{e_1}, \bar{e_4} \rangle$; (e) the **derived-features** method for computing the feature values for the same rule as in (d).

gual SMT system does. However, the tuning setup for the paraphraser is straightforward: regardless of how the paraphrasing model will be used, it is possible to use any existing set of English paraphrases as the tuning set for English-to-English translation. Given such a set of paraphrases, one sentence can be randomly chosen as the source sentence, and the remainder as the "reference paraphrases" for the purpose of finding the optimal feature weights. The optimization is then carried out exactly as it would be for tuning the weights for features in a bilingual translation system. The details of the actual tuning algorithm used are described in Chapter 4.

3.2.4 Evaluating Feature Computation Methods

Two methods of computing the feature values for the pivoted monolingual translation rules were proposed. In order to determine which of these methods produces better paraphrases, a simple manual evaluation was conducted using Amazon Mechanical Turk. To set up the experiment, a bilingual grammar was first extracted using approximately 1 million sentences of Chinese-English parallel newswire data. Two pivoted monolingual grammars were then constructed using the methodology described in the preceding sections: one where the features were computed using the derived-counts method and the other where the derived-features method was used. For tuning the parameters in both cases, the set of reference translations from the 2002 NIST MT evaluation exercise (hereafter referred to as NIST MT02) was used to find the optimal feature weights for the paraphraser. This set consists of 4 sets of translations written by humans for 878 Chinese source sentences and, therefore,

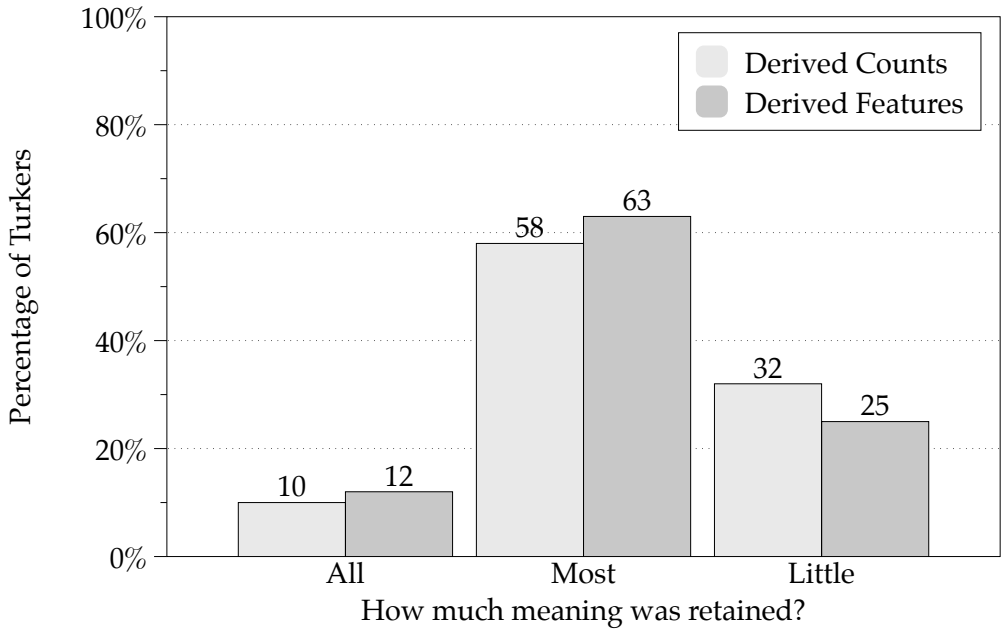


Figure 3.2: This figure shows the percentage of Turkers that rated a paraphrase as one that retains all of the meaning in the original sentence, most of the meaning or little to none of the meaning. Two different types of paraphrases were shown to the Turkers: the first type were generated with features computed using the derived-counts method and the second with features computed using the derived-features method.

may be considered to be a set of paraphrases.

Once any English-to-English translation model, represented by the paraphrase grammar, has been induced, it can be used within the SMT decoder, just as a bilingual translation model would be used, to paraphrase (translate) new sentences. As the input to the two paraphrasers, 100 sentences were randomly chosen from the NIST MT03 set and two sets of paraphrases were generated. 10 Human Intelligence Tasks (**HITs**) were created on Amazon Mechanical Turk such that each task contained 10 of the original English sentences along with the corresponding paraphrase from one of the paraphrasers. In each HIT, the Turkers were asked to read each of the 20 sentences and its corresponding paraphrase and choose one of the following

options:[2]

- All meaning expressed in the first sentence appears in the second sentence.

- Most of the meaning expressed in the first sentence appears in the second sentence.

- Little to none of the meaning expressed in the first sentence appears in the second sentence.

Note that each HIT was redundantly processed by three different Turkers and the final rating for each sentence was chosen by taking the majority of the three ratings. Figure 3.2 summarizes the results of these HITs. First, note that the two methods seem to produce relatively similar paraphrases even though it looks like the paraphrases produced with the derived-features method generally tend to retain more of the original meaning on average. Another important observation is that a large percentage of the sentential paraphrases, for either method, are not always fully semantically equivalent. This is not entirely unexpected due to the noisy nature of the word alignments for the bilingual parallel corpus. Based on these experiments, the derived-features method is chosen as the default method for computing the monolingual rule features heretofore. Figure 3.3 shows some examples of paraphrases produced with rules whose features were computed using this method.

The next chapter shows that despite being unsuitable for direct use by humans, the paraphrases produced by the sentential paraphraser as described in this chapter,

[2]In the first version of the HITs, five options were chosen instead of three but it was observed that with a larger list of choices, Turkers—who are entirely untrained—could not make the requisite fine distinctions. They always tended to pick one of the middle options and avoided committing to the extrema.

Orig	Alcatel added that the company's whole year earnings would be announced on February 4.
Para	Alcatel said that the company's total annual revenues would be released on February 4.
Orig	He was now preparing a speech concerning the US policy for the upcoming World Economic Forum.
Para	He was now ready to talk with regard to the US policies for the forthcoming International Economic Forum.
Orig	Tibet has entered an excellent phase of political stability, ethnic unity and people living in peace.
Para	Tibetans have come to cordial political stability, national unity and lived in harmony.
Orig	Its ocean and blue-sky scenery and the mediterranean climate make it world's famous scenic spot.
Para	Its harbour and blue-sky appearance and the border situation decided it world's renowned tourist attraction.

Figure 3.3: Example paraphrases with Chinese as the pivot language. **Orig** denotes the original sentence and **Para** its generated paraphrase. The sentences were chosen manually.

prove to be extremely useful for addressing the reference sparsity problem in a bilingual SMT system, thereby completing the circle of meaning.

4 The Next 180 Degrees: Improved SMT via Paraphrasing

———— * ————

Desire translation?
Show me how you would do it.
Not once but four times.
–Nitin Madnani

———— * ————

In this chapter, the following question is explored: assuming that only a *single* good reference translation is available for each item in the development set used for parameter tuning, how well can this reference sparsity be offset by augmenting that single reference with artificial references produced by the sentential paraphraser described in Chapter 3? This question is important for a number of reasons. First, with a few exceptions—notably NIST's annual MT evaluations—most new MT research data sets are provided with only a single reference translation. Second, obtaining multiple reference translations in rapid development, low-density source language scenarios (e.g. (Oard, 2003)) is likely to be severely limited (or made entirely impractical) by limitations of time, cost, and ready availability of qualified translators.

4.1 The Tuning Algorithm

Recall that the SMT system used in this dissertation uses a log-linear model employing several features to assign scores to any candidate translation. A log-linear model generally models the posterior probability of a candidate target language translation directly, i.e., for any given source sentence $\mathbf{f}$:[1]

$$P(\mathbf{e}|\mathbf{f}) = \frac{\exp\left(\sum_{m=1}^{N} \lambda_m \phi_m(\mathbf{e}, \mathbf{f})\right)}{\sum_{\mathbf{e}'} \exp\left(\sum_{m=1}^{N} \lambda_m \phi_m(\mathbf{e}', \mathbf{f})\right)} \tag{4.1}$$

where the ϕ's are the feature functions defined over the source and target language strings and the λ's are the weights for these features. The most likely candidate translation can then be obtained by the search—or decoding—process as given by:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e'}} P(\mathbf{e'}|\mathbf{f}) \tag{4.2}$$

As part of training this translation model, it is standard practice to *tune* the set of feature weights λ (or parameters) for such a model. While such tuning can certainly be carried out in the usual manner to maximize likelihood, it has been shown that a more effective method used for tuning these parameters is a multidimensional optimization technique wherein the objective is to directly minimize the translation error—measured over n-best lists of translation candidates against a set of reference translations—for held-out "development" source sentences paired with their

[1]Technically, the model scores derivations and not target language strings. However, without any loss of generality, it is assumed that such a score can be computed for the target language string—a linearization of a candidate derivation—as well.

corresponding reference translations (Och, 2003).

The learning process seeks to minimize the error over the entire development set F which is simply the sum of the errors for each individual source sentence in the set:

$$Err(F) = Err(\hat{\mathbf{e}}_1, E_1^R) + Err(\hat{\mathbf{e}}_2, E_2^R) + \cdots + Err(\hat{\mathbf{e}}_{\|F\|}, E_{\|F\|}^R) \tag{4.3}$$

where $\mathbf{E}_k^R$ represents the set of reference translations for the k^{th} source sentence. The set of tuned parameters minimize the above translation error on the development set, i.e.,:

$$\vec{\lambda}_{optimal} = \arg\min_{\vec{\lambda}} Err(F) \tag{4.4}$$

Notice that this construction of the optimization problem has an *argmax* embedded inside an *argmin* (via the search process for the best candidate in Equation 4.2 above). Therefore, no gradient-based optimization methods can be utilized. Given these circumstances, a globally optimal solution is not guaranteed. However, a heuristic is generally used to find a good locally optimal solution (Och, 2003). This heuristic is described below.

Ignoring the normalization constant in Equation 4.1, the unnormalized score of a candidate translation can be written as:

$$\text{scr}(\mathbf{e}|\mathbf{f}) = \lambda_1\phi_1(\mathbf{e}, \mathbf{f}) + \lambda_2\phi_2(\mathbf{e}, \mathbf{f}) + \cdots + \lambda_N\phi_N(\mathbf{e}, \mathbf{f}) \tag{4.5}$$

Notice that since the feature functions are being evaluated at a particular source sentence and a particular candidate translation, their values are essentially constant.

Assuming that the values of all parameters except one, say λ_1, are held fixed, then this score can be easily seen to be linear in λ_1:

$$\text{scr}(\mathbf{e}|\mathbf{f}) = \lambda_1 \underbrace{\phi_1(\mathbf{e},\mathbf{f})}_{constant} + \underbrace{\lambda_2\phi_2(\mathbf{e},\mathbf{f}) + \cdots + \lambda_N\phi_N(\mathbf{e},\mathbf{f})}_{constant} \tag{4.6}$$

$$= m\lambda_1 + c \tag{4.7}$$

Therefore, all candidate translations from the n-best list for a given source sentence can be represented as a line, in an N-dimensional space (N being the number of feature functions) spanned by the feature weight λ's, with properly defined slope and intercept. The next step, then, is to find the intersection points between all such lines. Once the intersection points are found, they yield a number of intervals along each of which the error remains constant, given solely by the line (candidate) that contributed to that interval. As mentioned earlier, the total error over the entire development set is deemed to be additive in nature and so by combining all such lines for all the source sentences, a piecewise-linear approximation to the actual error surface can be computed. All that remains to be done then is to traverse the intervals along this function to find the one with the lowest error value and finally, determine the optimum value of the one non-constant feature by reverse mapping. This line minimization procedure is then carried out for each of the remaining parameters. After each parameter value has been determined, one iteration of the tuning process is deemed to have been carried out. Successive iterations may be necessary depending on pre-stipulated convergence criteria which are usually defined either in terms

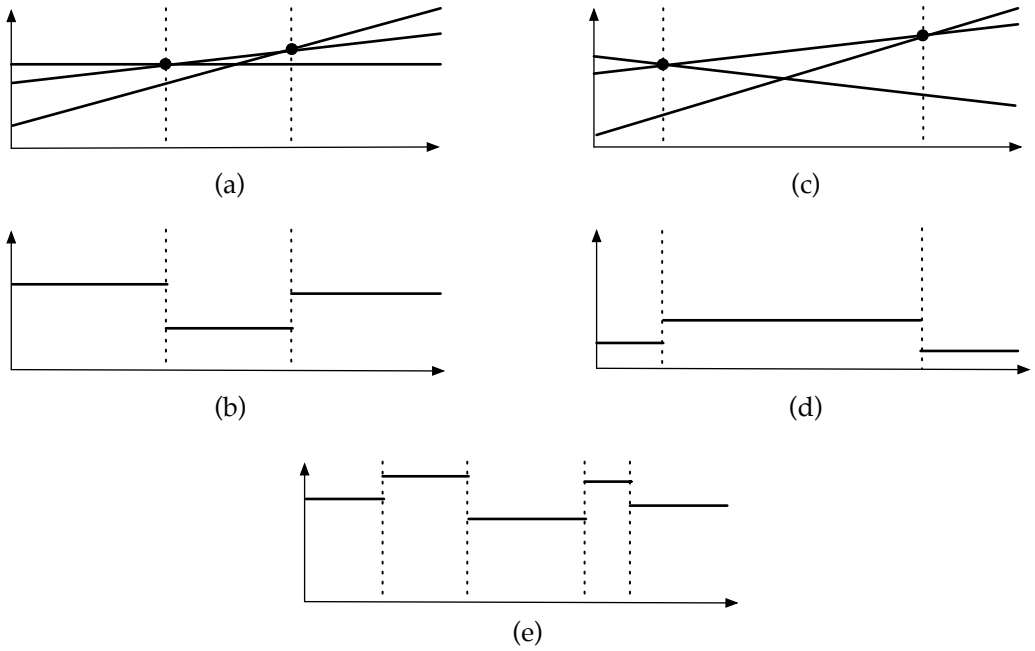


Figure 4.1: A visual depiction of the line minimization procedure that is at at the heart of the minimum error rate training procedure. In (a), the scores for the candidate translations for a given source sentence can be drawn as lines in λ_i with all other parameters held constant. Intersection points are then found in order to find intervals where the error remains constant. The intervals are shown in (b). The same procedure is then repeated for a different source sentence in (c) and (d). Assuming that the development set consists of only these two source sentence, an accumulated piecewise-linear error function can then be computed in (e). The optimal value of λ_i can then be found by simply finding the interval with the lowest error and mapping back to the X-axis to find the parameter value, usually taken to be that corresponding to the midpoint of the interval. This same procedure is repeated for each parameter value to optimize it in isolation. This figure is reproduced from (Lopez, 2008).

of how much the parameter values change and the diversity contained in the n-best lists. Figure 4.1, reproduced here from (Lopez, 2008), provides a visual depiction of the line minimization procedure.

For the sake of completeness, it should be mentioned that the method for finding an optimal point in a multidimensional parameter space using n-best hypothesis lists was originally proposed in the 1990s for automatic speech recognition systems (Ostendorf et al., 1991). The algorithm used there is Powell's method (Pow-

ell, 1965; Press et al., 1986), which iteratively optimizes the weights in successive conjugate directions. The primary algorithmic difference between the two methods is that the one using Powell's method can be used with arbitrary scoring criteria since it relies on standard grid-based line minimization for each conjugate direction. In contrast, the method described above and proposed by Och (2003) derive an *exact* line optimization technique specifically for log-linear models. However, both techniques are susceptible to local optima and heuristics such as multiple starting points and random restarts usually have to be employed.

Och (2003) also showed that the translation system achieves its best performance on unseen data when parameters are tuned using the *same* objective function that is used for evaluating the system. Since BLEU is the most commonly used evaluation metric, systems are generally tuned to maximize the translation quality of the system on the development set as measured by BLEU.[2] As with Equation 4.3, the BLEU scores are computed against a set containing *multiple* reference translations. Since BLEU is based on n-gram overlap between hypotheses and reference translations, it is most accurate when computed with as many distinct reference translations (usually four) as possible. Intuitively this makes sense: if there are alternative ways to phrase the meaning of the source sentence in the target language, then the translation quality criterion should take as many of those variations into account as possible. To do otherwise is to risk the possibility that the criterion might judge good translations to be poor when they fail to match the exact wording

[2]Since translation quality is inversely related to translation error, line *maximization* is employed to tune each parameter in isolation.

within the reference translations that have been provided.

However, this reliance on multiple reference translations creates a problem because reference translations are labor intensive and expensive to obtain. For example, producing reference translations at the Linguistic Data Consortium, a common source of translated data for MT research, requires undertaking an elaborate process that involves translation agencies, detailed translation guidelines, and quality control processes (Strassel et al., 2006). Therefore, most current MT development sets only come with a single reference translation, leading to *reference sparsity*. The view adopted in this thesis is that this sparsity can be effectively, and cheaply, addressed by using the sentential paraphraser from Chapter 3 to create additional references in an artificial manner and using them for the BLEU-driven parameter tuning process.

An alternate view that deserves mention is that of using a different MT metric for tuning; one that has an inherent notion of semantic equivalence, such as METEOR or TERp. Using one of these metrics does alleviate the effects of reference sparsity and, as such, they are being increasingly employed for MT evaluation. However, BLEU still remains the most commonly accepted evaluation metric and, therefore, the best translation performance is achieved by using BLEU for parameter tuning as well. In addition, tuning with METEOR or TERp is accompanied by additional issues that will need to be worked out before they can be used as replacements for BLEU.

4.2 Experimental Questions

At a finer level of granularity, the answer to this chapter's primary question—how well can we do armed with an automatic paraphraser and a single human reference?—can be deemed as a combination of answers to more specific questions that are listed below:

1. If only a single reference translation is available for tuning, does adding its best sentential paraphrase reference into the tuning process provide significant gains?

2. Can k-best paraphrasing, instead of just 1-best, lead to better optimization, and how does this compare with using additional human references translations?

3. Given the claim that the paraphraser provides additional n-gram diversity, can the paraphraser be useful in situations where the tuning criterion does not depend heavily on such overlap?

4. To what extent are the gains obtained from this technique contingent on the quality of the human references that are being paraphrased, if at all?

5. Are the paraphrased references are equally useful with larger tuning sets that can be created simply by borrowing from the parallel bitext?

Answering these questions will make it possible to characterize the utility of paraphrase-based optimization in real-world scenarios, and and how best to leverage it in those

scenarios where it does prove useful. The answers to the first two questions are described next in Sections 4.3.1 and 4.3.2 in terms of results of translation experiments and indicate the general utility of the paraphraser for multiple source languages. The answers to the remaining three, more specific, questions are presented in Sections 4.4, 4.5 and 4.6 respectively.

4.3 Translation Experiments and Results

Here, we address the experimental questions as to whether significant gains are obtained by including the 1-best (or even the k-best) sentential paraphrases into the tuning process when only one single reference translation is available. Machine translation experiments are described that use the sentential paraphraser to create additional references for parameter tuning and the results are compared to the results of baseline experiments that only use that one reference translation for tuning. Two kinds of results are presented for each set of experiments. The first kind compares translation systems to each other using automatic machine translation evaluation metrics BLEU and TER and the second compares them using human judgments.

4.3.1 Chinese-English

The first set of experiments is for Chinese-English translation with the following details:

1. **Training Data**. The Chinese-English parallel data used for this set of ex-

periments consist of approximately 2.5 million newswire segments. Both the translation and the paraphrase rules are generated using this data. Besides the parallel data, approximately 8 billion words of English text are used for language model (LM) training (3.7B words from the LDC Gigaword corpus, 3.3B words of web-downloaded text, and 1.1B words of data from CNN archives). These data are used to train two language models: a trigram LM used in decoding, and an unpruned 5-gram LM used in reranking both the SMT and the paraphraser n-best lists. Modified Kneser-Ney smoothing was applied to the n-grams in both cases (Chen and Goodman, 1998).

2. **Decoders**. Both the SMT and paraphrase decoders (Shen et al., 2010) use a state-of-the-art hierarchical phrase-based translation model where the translation (or paraphrasing) rules form a synchronous context free grammar (SCFG).

3. **Tuning Set**. As the tuning set, the NIST MT03 Chinese set containing 919 sentences is used. This dataset actually comes with 4 human authored reference translations per item and a tuning set is simulated in which only a single reference translation is available.[3] One way to create such a simulated set is simply to choose one of the 4 reference sets, i.e., all the translations with the same system identifier for all source documents in the set. However, for the NIST sets, each of the reference sets is typically created by a different human translator. In order to imitate a more realistic scenario where multiple human translators collaborate to produce a *single* set of reference translations, instead

[3]The reasons for choosing a set with 4 references will become clearer in the subsequent paragraphs.

of multiple sets, it is essential to normalize over any translator idiosyncrasies so as to avoid any bias. Therefore, the simulated single-reference set is created by choosing, at random, for *each* source document in the set, one of the 4 available reference translations.

4. **Validation Set**. The validation set is the NIST MT04+05 set which contains both NIST MT04 and NIST MT05 sets. The total number of sentences in this set is 2870. No changes are made to the number of references in the validation set. Only the tuning sets differ in the number of references across different experiments.

As the baseline, the simulated single-reference set (1H=1 Human) is used to tune the parameters of the SMT system and evaluate on the MT04+05 validation set The simulated set is then paraphrased, the 1-best paraphrase extracted as an additional reference, and the MT system tuned on this new 2 reference tuning set (1H+1P=1 Human, 1 Paraphrase). The results, shown in Figure 4.2, confirm that using a paraphrased reference when only a single human reference is available is extremely useful and leads to huge gains in both the BLEU and TER scores on the validation set.[4]

Since the paraphraser is an English-to-English SMT system, it can generate k-best hypothesis paraphrases from the chart for each input reference. An obvious extension to the above experiment then is to see whether using k-best paraphrase hypotheses as additional reference translations, instead of just the 1-best, can alle-

[4]BLEU and TER are calculated on lowercased translation output. For each experiment, BLEU scores shown in bold are significantly better (Koehn, 2004) than the appropriate baselines for that experiment ($p < 0.05$).

	BLEU	TER
1H	37.65	56.39
1H+1P	**39.32**	54.39

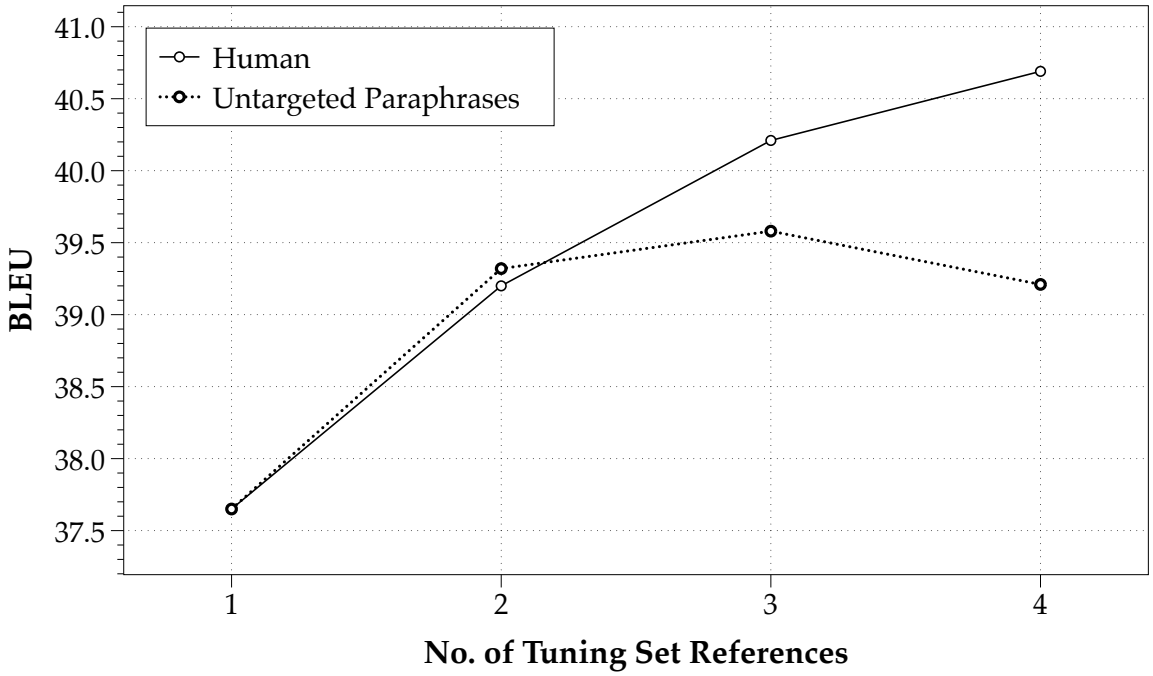
Figure 4.2: BLEU and TER scores are shown for MT04+05. 1H=Tuning with 1 human reference, 1H+1P=Tuning with the human reference *and* its paraphrase. Lower TER scores are better.

viate the reference sparsity to a larger extent during the optimization process. For this experiment, the top 1, 2 and 3 paraphrases for the MT03 simulated single reference set are used as additional references; three tuning sets 1H+1P, 1H+2P and 1H+3P respectively. As points of comparison, the tuning sets 2H, 3H and 4H are also constructed from MT03 in the same simulated fashion[5] as the single reference tuning set 1H. The results for this experiment are shown in Figure 4.3.

The graph shows that starting from the simulated single reference set, adding one more human reference translation leads to a significant gain in BLEU score, and adding more human references provides smaller but consistent gains at each step. With paraphrased references, gains continue up to 3 references, and then drop off; presumably beyond the top two paraphrases or so, n-best paraphrasing adds more noise than genuine diversity (one can observe this drop off in provided diversity[6] in the example shown in Figure 4.4). Crucially, however, it is important to note that *only* the performance difference with four references—between the human and the paraphrase condition— is statistically significant.

[5]By randomly choosing a sufficient number of random reference translations from the available 4 for each source document.

[6]This lack of diversity is found in most forms of n-best lists used in language processing systems and has been documented elsewhere in more detail (Langkilde, 2000; Mi et al., 2008).



# tuning refs	Human		Paraphrased	
	BLEU	TER	BLEU	TER
1 (1H+0)	37.65	56.39	37.65	56.39
2 (1H+1)	**39.20**	54.48	**39.32**	54.39
3 (1H+2)	**40.01**	53.50	**39.79**	53.71
4 (1H+3)	**40.56**	53.31	**39.21**	53.46

Figure 4.3: A graph showing the BLEU scores for the set NIST MT04+05 as human and paraphrased reference translations are added to a single human authored reference translation for the tuning set (NIST MT03). Note that the BLEU score for this validation set is measured against 4 human references in each case. Only the number of references for the tuning set is varied. The corresponding TER scores are shown in the accompanying table.

In addition to results with automatic metrics, it would also be worthwhile to get actual human judgments indicating whether the translations produced by the system augmented with paraphrases are significantly better than those produced by the baseline system that is tuned with the single human-authored reference. To obtain such judgments, two sets of experiments were conducted on Amazon

O	(Hong Kong, Macau and Taiwan) Macau passed legalization to avoid double tax.
$\mathbf{P_1}$	Macao adopted bills to avoidance of double taxation (Hong Kong, Macao and Taiwan).
$\mathbf{P_2}$	(Hong Kong, Macao and Taiwan) Macao adopted bills and avoidance of double taxation.
$\mathbf{P_3}$	(Hong Kong, Macao and Taiwan) Macao approved bills and avoidance of double taxation.

Figure 4.4: The 3-best paraphrase hypotheses for the original sentence O with Chinese as the pivot language. The amount of n-gram diversity decreases with each successive hypothesis.

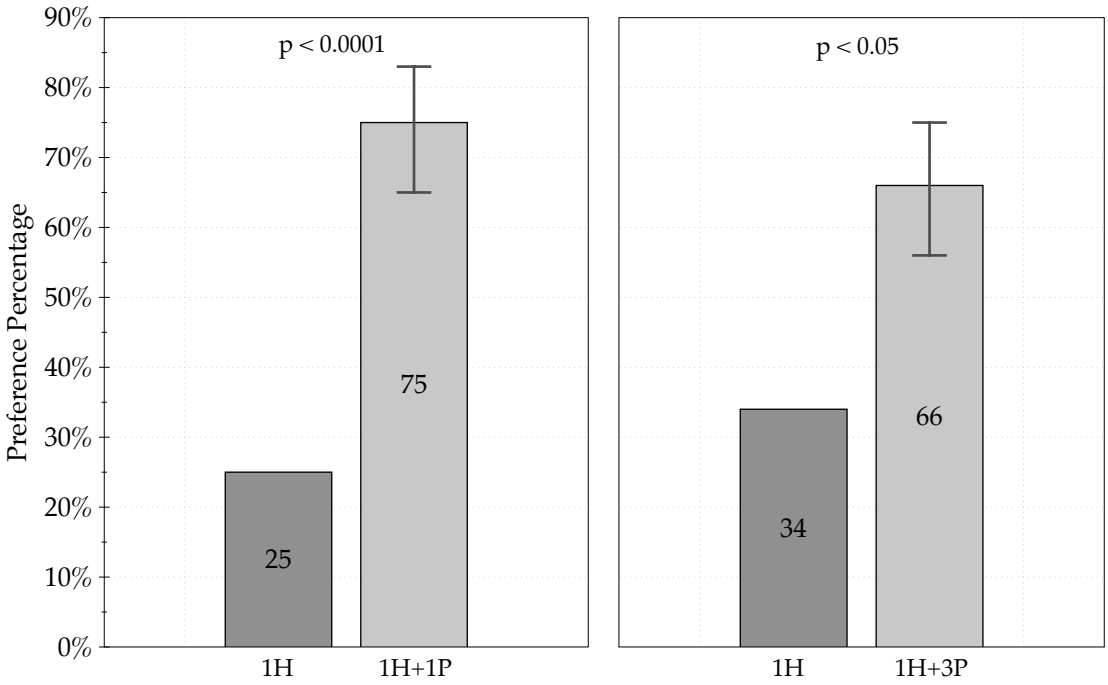


Figure 4.5: When translating Chinese sentences, human subjects on Amazon Mechanical Turk prefer–to a statistically significant extent—the translations produced by the MT system that was tuned with the paraphrase as additional, artificial references (1H+1P) compared to the system that used only the single human-authored reference (1H). The relatively lower performance with the noisy 3-best paraphrases (1H+3P), although still significantly better than the baseline, is also evident in these preference judgments.

Mechanical Turk:

- **1H vs 1H+1P**. 100 sentences were randomly chosen from the NIST MT04+05 Chinese validation set. 10 HITs were then created, each containing 10 of the

100 source sentences along with: (a) the corresponding reference translations (b) translation outputs from a system tuned with the single human-authored reference and, (c) translation outputs from a system tuned with the single human-authored reference *and* its 1-best paraphrase.

- **1H vs 1H+3P**. 100 sentences were randomly chosen from the NIST MT04+05 Chinese validation set. 10 HITs were then created, each containing 10 of the 100 source sentences along with: (a) the corresponding reference translations (b) translation outputs from a system tuned with the single human-authored reference and, (c) translation outputs from a system tuned with the single human-authored reference *and* its 3-best paraphrases instead of just the 1-best.

The instructions in each HIT for both sets of experiments above told the participating Turkers to pick the translation output that they thought was more correct. A third option indicating that there was no difference between the two was also provided. Answers from Turkers were validated by embedding a control question in each HIT for which the correct answer was known before hand. If the answer given by a Turker for this control question did not match the known answer, her answers for that HIT were discarded. Each sentence was annotated three times and the final answer for each sentence was picked by a simple majority vote. If the final answer for a sentence was the no-difference option, then that sentence was excluded from consideration.

The results for both of these sets of experiments are shown in Figure 4.5. It is clearly evident from the preference judgments obtained from the Turkers that using the paraphrases as artificial references yields significantly improved translation output when compared to the baseline system that was only tuned with a single reference. Assuming that choosing the system augmented with paraphrases is deemed as "success", the null hypothesis is that success and failure are equally likely with a probability of 0.5. Using a two-sided exact binomial test with these judgments shows that they are not in agreement with the null hypothesis and that the success is more likely than failure with p-values as shown in the figure.[7] 95% confidence intervals are also shown. As with the automatic metrics, using the 3-best paraphrases tends to lead to lower performance for the MT system due to the decreased n-gram diversity and increased noise.

4.3.2 French-English, German-English and Spanish-English

In addition to showing the applicability of the sentential paraphraser for tuning in Chinese-English, it would also be very useful to show the same for a language pair for which there is only a single reference available and no simulation is required. Of course, for such a scenario, it would be impossible to compare the performance of the artificial, paraphrased references to actual human references. In this section, machine translation experiments from three different European languages—French, German and Spanish—into English are presented with the following details:

1. **Training Data**. For these sets of experiments, mainly bitexts extracted

[7] The binomial test was carried out in R using the command `binom.test`.

from the proceedings of the European parliament (Koehn, 2005) are used; specifically, 1.7 million sentences for French-English, 1.6 million sentences for German-English and 1.7 million sentences for Spanish-English. In addition to these data, the smaller news commentary data for each language containing respectively 82K, 75K and 74K sentences for French-English, German-English and Spanish-English is also used. As the language model training data, the same data as the Chinese-English experiments were used.

2. **Decoders**. Both the SMT and paraphrase decoders are SCFG-based decoders using hierarchical phrase-based translation models.

3. **Tuning Set**. As the tuning set, the test set from the 2008 workshop on machine translation (Callison-Burch et al., 2008c) which contains $2,051$ sentences is used. This is in-domain data that was gathered from the same news sources as the validation set described below. Note that this set only contains a single human authored English reference translation.

4. **Validation Set**. The validation set is the newstest2009 set which contains $2,525$ sentences also with a *single* English reference translation.

Figure 4.6 shows the BLEU and TER results for the validation sets for each of the three language pairs. These results confirm that the benefits from the paraphrased references are able to generalize across multiple language pairs. However, as more paraphrases are added, the noise inherent in the paraphrases starts to overwhelm the benefits. One important thing to note about this set of results is that

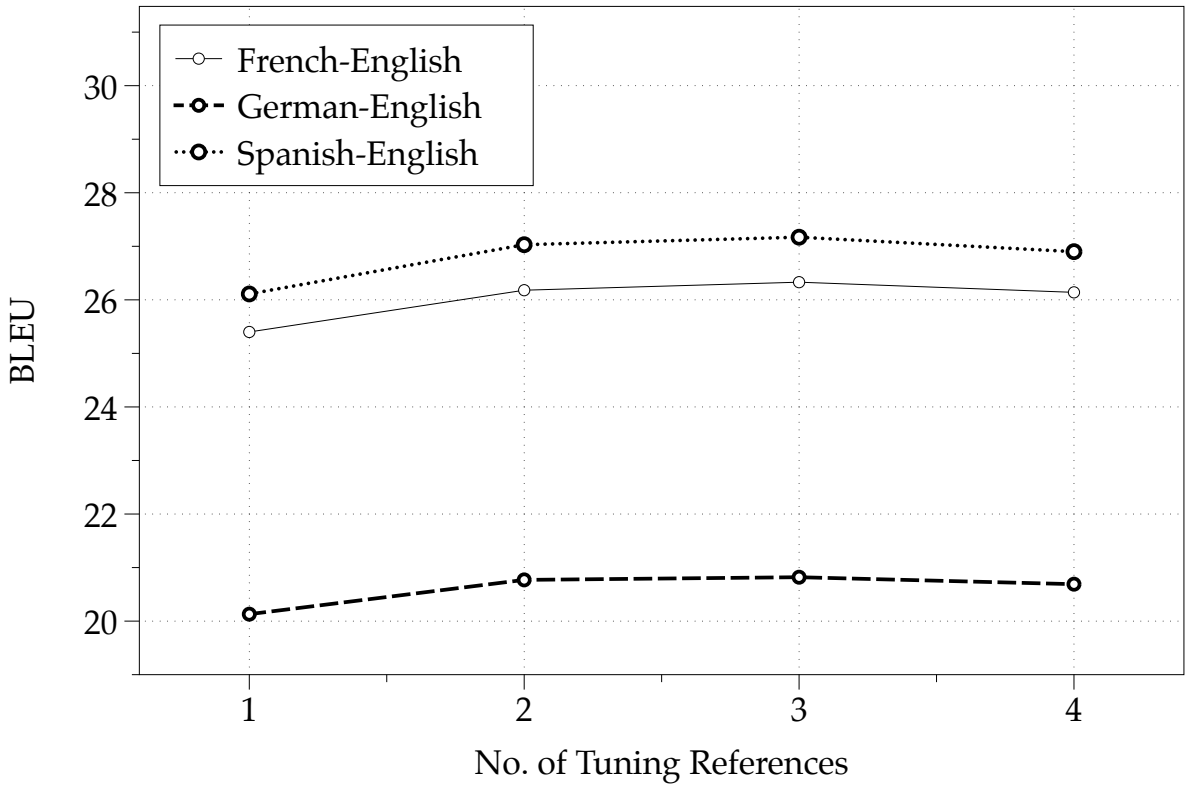
the improvements in the BLEU scores are much smaller than in the case of Chinese. The most likely reason for this is that the BLEU scores are being measured only against a single reference whereas in the case of Chinese-English translation, they were being measured against 4 reference translations that were available for the validation set.

To confirm whether human judgments of translation outputs would agree with the automatic metrics, experiments identical to the Chinese-English case were conducted on Amazon Mechanical Turk. Figure 4.7 shows the results of these experiments for the three European languages.

4.4 The Role of the Tuning Metric

This section addresses the question of whether the additional diversity provided by the paraphraser is useful in situations where the turning criterion does not depend heavily on such overlap.

Although BLEU is generally the most widely used metric for parameter tuning in SMT, other metrics may also be employed in different scenarios. For example, a genre that has recently gained in popularity is the weblog genre. Previous experience with this genre has shown that if BLEU is used as the tuning criterion for this genre, the TER scores on held-out validation sets tend to be disproportionately worse. It has been shown that a good criterion to use is a hybrid TER-BLEU



# refs	French-English		German-English		Spanish-English	
	BLEU	TER	BLEU	TER	BLEU	TER
1H+0	25.40	56.14	20.13	60.80	26.11	55.32
1H+1	**26.18**	55.47	**20.77**	60.17	**27.03**	54.44
1H+2	**26.33**	55.26	**20.87**	60.06	**27.17**	54.31
1H+3	**26.14**	55.40	**20.74**	60.18	**26.90**	54.55

Figure 4.6: Graph showing the BLEU scores for the set newstest2009 as paraphrased reference translations are added to a single human authored reference translation for the tuning set (newstest2008). Note that the BLEU score for this validation set is measured against one reference translation. The corresponding TER scores are shown in the table.

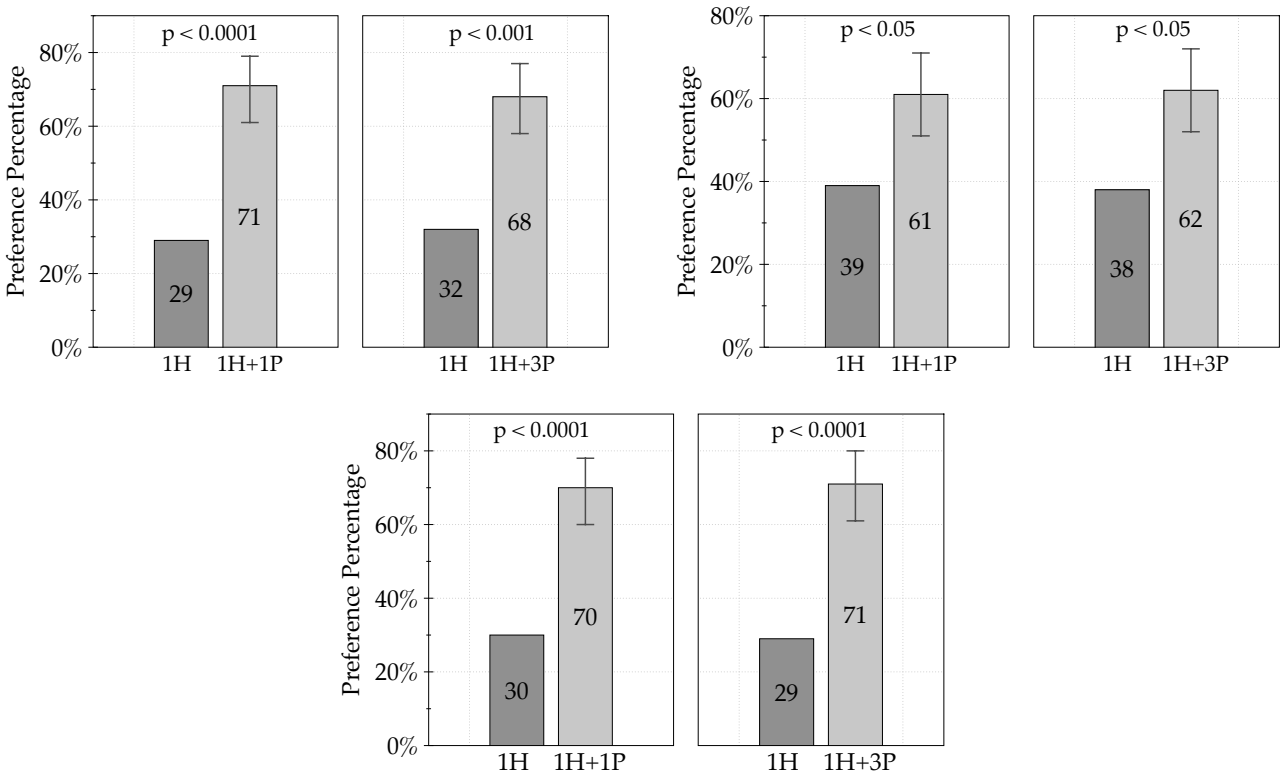


Figure 4.7: Just like Chinese-English translations, human subjects—when asked to judge translations for European languages—have a significant preference for outputs produced by an SMT system augmented with paraphrases (1H+1P and 1H+3P) as compared to baseline translation output (1H). Clockwise from top to bottom: French-English, German-English and Spanish-English.

measure (Matsoukas et al., 2009) given by:

$$\text{TERBLEU} = 0.5 * \text{TER} + 0.5 * (1 - \text{BLEU})$$

The same measure is used for tuning the MT system in this experiment in order to test how the use of a criterion that is not as heavily dependent on n-gram diversity as BLEU affects the utility of the paraphraser in a real-world scenario.

In order to test how the paraphrase approach works in that genre, both the MT

system and the paraphraser are trained on 600,000 sentences of weblog data. Note that this is substantially smaller than the amount of newswire text that was used to train the paraphraser for previous experiments. As the tuning set, an actual weblog data set is used with only a single reference translation and containing approximately 800 sentences. As the validation set, a second weblog data set (WEB) containing 767 sentences is used, also with a single reference translation. The results are shown in Figure 4.8.

	BLEU		TER
	Prec.	BP	
1H	16.85	0.90	68.35
1H+1P	17.25	0.88	68.00

Figure 4.8: BLEU and TER scores on the WEB validation set when using paraphrases for tuning an SMT system used to translate the weblog genre. The tuning metric used is TERBLEU instead of BLEU.

Since the validation set has a single reference translation, the 4-gram precision and brevity penalty components of BLEU scores can be separated out. It is important to focus on the precision which is directly affected by the increased n-gram diversity supplied by the paraphrase. For this experiment, it is seen that while there seem to be improvements in both the 4-gram precision and TER scores, they are statistically insignificant. In order to isolate whether the lack of improvement is due to the relatively small size of the training data or the metric mismatch, the same experiment is run with BLEU as the tuning criterion instead of TERBLEU.

The results, shown in Figure 4.9, indicate a significant gain in both the 4-gram precision and the overall BLEU score. Therefore, to answer the question, using

	BLEU		TER
	Prec.	BP	
1H	17.05	0.89	70.32
1H+1P	**18.30**	0.87	69.94

Figure 4.9: BLEU and TER scores on the WEB validation set when using paraphrases for tuning an SMT system used to translate the weblog genre. The tuning metric used here is BLEU. A significant gain in BLEU is achieved only when the tuning criterion for the MT system can take advantage of the diversity.

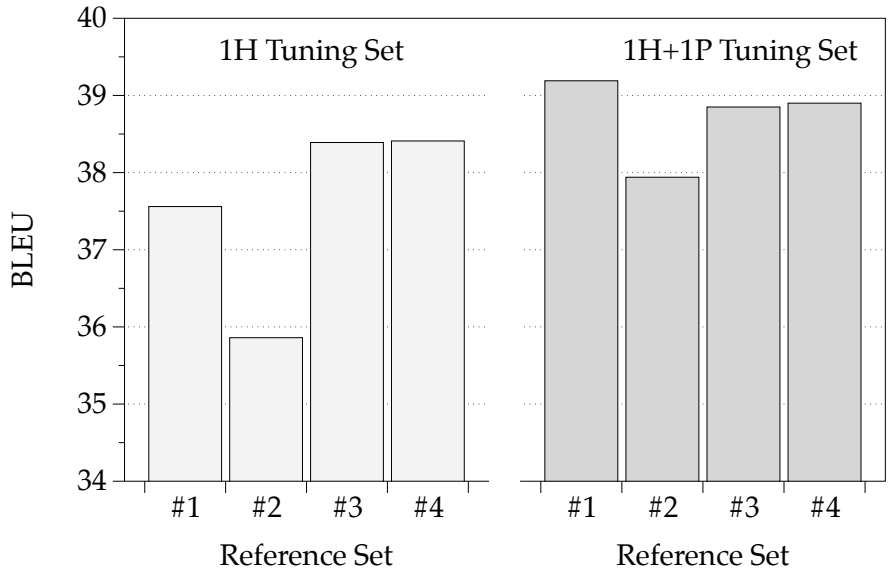
a tuning criterion that doesn't benefit from added n-gram diversity can certainly hamper the paraphraser's effectiveness in addressing reference sparsity.

4.5 Impact of Human Translation Quality

This section addresses the question as to the extent to which the gains obtained from the paraphrase-based technique are contingent on the quality of the human references that are being paraphrased.

Each of the 4 sets of references translations in the Chinese-English MT03 dataset was created by a different human translator. Since human translators are likely to vary significantly in the quality of translations that they produce, it is important to gauge the impact of the quality of a reference on the effectiveness of using its paraphrase, at least as produced by our proposed sentential paraphraser, as an additional reference. To do this, each of the 4 reference sets from MT03 is chosen in turn to create the simulated single-reference set (1H).[8] It is then paraphrased and the 1-best paraphrase used as an additional reference to create a 2-reference tuning set (1H+1P). Each of these 8 tuning sets is then used to tune the SMT system and

[8]Note that these per-translator simulated sets are different from the bias-free simulated set created in Sections 4.3.



	BLEU			
	#1	#2	#3	#4
1H	37.56	35.86	38.39	38.41
1H+1P	**39.19**	**37.94**	38.85	38.90

	TER			
	#1	#2	#3	#4
1H	57.23	60.55	54.50	54.12
1H+1P	54.21	56.42	53.40	53.51

Figure 4.10: BLEU and TER scores computed for MT04+05 for cases where tuning employs reference translations (created by different human translators) and their corresponding paraphrases. Tuning usefulness of human translations vary widely (e.g., Refset #2 vs Refset #4) and, in turn, impact the utility of the paraphraser.

the BLEU and TER scores are computed for the validation set MT04+05.

Figure 4.10 shows these results in graphical form. These results yield two very interesting observations:

- The human reference translations do vary significantly in quality. This is clearly seen from the significant differences in the BLEU and TER scores between the 1H conditions, e.g., the third and the fourth human reference

Tuning Set	# of Sentences
Base (MT03)	919
T1 (Base+600)	1519
T2 (T1+500)	2019
T3 (T2+500)	2519

Figure 4.11: Creating larger single reference tuning sets by adding sentences from the training corpus to the single reference base tuning set (MT03).

translations seem to be better suited for tuning than, say, the second reference. Note that the term "better" does not necessarily refer to a more fluent translation but to one that is closer to the output of the MT system.

- The quality of the human reference has a significant impact on the effectiveness of its paraphrase as an additional tuning reference. Using paraphrases for references that are not very informative, e.g. the second one, leads to significant gains in both BLEU and TER scores. On the other hand, references that are already well-suited to the tuning process, e.g., the fourth one, show much smaller improvements in both BLEU and TER on MT04+05.

4.6 Effect of Larger Tuning Sets

This section addresses the question of whether the paraphrased references are equally useful with larger tuning datasets. More precisely, it answers the question of whether using a larger set of sentences (with a single human reference translation) be as effective as using the sentential paraphraser to produce additional artificial reference translations.

Given that creating additional human reference translations is so expensive,

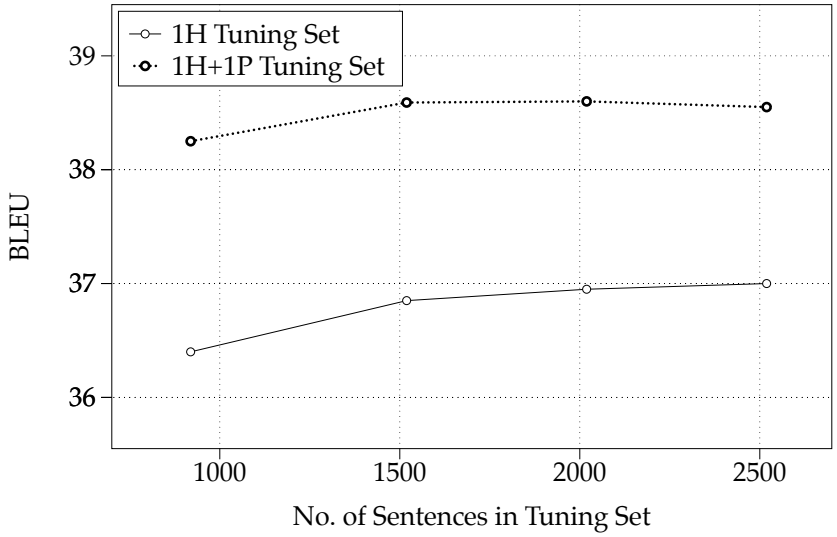
the most realistic and cost-effective option of scaling to larger tuning sets is to simply take the required number of sentences from the training data and add them to the tuning set (Zhang and Vogel, 2004). The parallel nature of the training corpus facilitates the use of the same corpus as a tuning set with a single human-authored reference translation.

In order to replicate this scenario, the single reference Chinese-English MT03 bias-free tuning set is chosen as the starting point. A block of sentences is then chosen from the MT training corpus[9] and is then added to the baseline MT03 tuning set in three steps to create three new tuning sets as shown in Figure 4.11.

Once the larger tuning sets are created, each them is used to tune the parameters of the SMT system (which is trained on a training corpus that excluded this block of sentences) and score the MT04+05 validation set. To see how this compares to the paraphrase-based approach, each of the tunings sets is paraphrased and the paraphrases are used as additional reference translations for tuning the MT system. Figure 4.12 shows these results in graphical form.

The most salient observation that can be made from the results is that doubling or even tripling the tuning set by adding more sentences from the training data *does not* lead to statistically significant gains. However, adding the paraphrases of the corresponding human reference translations as additional references for tuning *always* leads to significant gains, irrespective of the size of the tuning set.

[9]It has been verified that these sentences do *not* overlap with the paraphraser training data.



	BLEU			
	Base	T1	T2	T3
1H	36.40	36.85	36.95	37.00
1H+1P	**38.25**	**38.59**	**38.60**	**38.55**

	TER			
	Base	T1	T2	T3
1H	56.17	58.23	58.60	59.03
1H+1P	54.20	55.43	55.59	55.77

Figure 4.12: BLEU and TER scores for the validation set MT04+05 vary as the tuning set is enlarged—by adding sentences from the training data. The effectiveness of the paraphraser remains strong despite the larger size of the tuning set.

4.7 Summary

At this point, the circle of meaning can be deemed to be complete in the manner that it had been motivated in Chapter 1. An automatic sentential paraphraser was constructed entirely from SMT machinery in Chapter 3 (the first 180 degrees of the circle). In this chapter, both automatic and manual evaluation results showed that the same sentential paraphraser can provide an extremely effective solution for the reference sparsity issue that afflicts current state-of-the-art SMT systems.

However, the behavior of the sentential paraphraser is undesirable in one aspect: in contrast to human reference translations, adding any paraphrases beyond the 1-best actually degrades the system performance. In the next chapter, an improved instantiation of the sentential paraphraser is described that does not exhibit this behavior.

This chapter also answered some additional questions pertaining to the utility of the paraphraser in providing a solution to the reference sparsity problem that affects start-of-the-art SMT systems. It showed that if the tuning metric used to find the parameters of the SMT system cannot benefit directly from the n-gram diversity that is supplied by additional reference translations, then the utility of the paraphraser is limited. Results also showed that the impact that the paraphrases can have as references is contingent on the quality of the human-authored reference that is being paraphrased and that such references can vary widely in quality. Finally, experiments also showed that the utility of the paraphrases as reference translations is not diminished simply by increasing the size of the tuning set by borrowing from the training data.

5 Beyond the 1-best: A Targeted Sentential Paraphraser

———— * ————

Changes without bounds.
Limited utility.
Perhaps, some focus?
–Nitin Madnani

———— * ————

Chapter 4 described how the pivot-based sentential paraphraser could be used to create multiple, artificial reference translations by paraphrasing an existing good quality, human reference translation. It also showed that using the 1-best artificial reference so produced, combined with the already existing human reference, for tuning the parameters of a statistical machine translation system yields significant gains. Therefore, using the pivot-based paraphraser in such a fashion is a viable method to address the problem of reference sparsity that affects current statistical machine translation systems.

However, using more than just the 1-best output from the sentential paraphraser as additional references for tuning does not yield any further improvements. In fact, doing so causes the performance to degrade as shown in Figure 5.1 reproduced from Chapter 4. This degradation is a direct result of the increased noise in the paraphrase output as one goes down the n-best list. As detailed in Section 2.4.5,

the bilingual phrasal correspondences extracted via commonly used word alignment methods are already noisy. The process of pivoting, a process relying on an equivalence relationship between these noisy correspondences that is approximate at best, only serves to introduce additional noise into the generated monolingual paraphrastic correspondences and, therefore, into the generated sentential paraphrases as well. Hence, including the top k-best ($k <= 3$) instead of just the 1-best adds so many noisy n-grams into the mix that the noise overwhelms any benefits from the additional diversity of the small number of useful n-grams. This chapter describes a new SMT-specific instantiation of the sentential paraphraser that overcomes this behavior.[1]

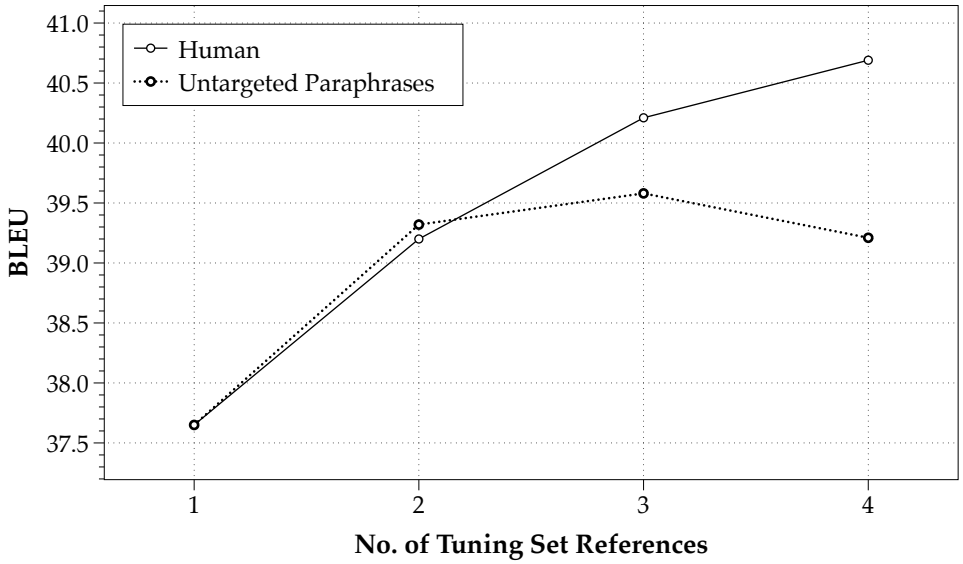


Figure 5.1: Adding k-best sentential paraphrases does not yield the same benefits as adding multiple human references.

Of course, to make the paraphraser more useful, one could try to control for the

[1]There are other ways in which the noise in the paraphrase pairs induced via pivoting can be reduced: (a) the syntactically constrained pivoting method proposed by Callison-Burch (2008)—and as described in Section 2.4.5 and, (b) using Amazon Mechanical Turk to filter out the noisy paraphrase pairs (Denkowski et al., 2010). However (a) leads to decreased coverage for the paraphraser and (b) requires humans intervention.

noise directly. This is certainly a reasonable option and one that will be attempted later in the chapter. However, there is a another, more serious problem with the sentential paraphraser as it exists at this point: there are absolutely no constraints imposed on the paraphrasing process. The paraphraser is free to paraphrase any and every n-gram in the input sentence with *any* n-gram that the pivoting process has declared to be equivalent to it. The result of such unconstrained paraphrasing is that there is no guarantee that the paraphrased reference *will* match the translation output for that particular source sentence. It is basically a crap shoot; by allowing the paraphraser free rein, the hope is that at least *some* of the new n-grams brought into the mix via the paraphrased references will prove useful to the tuning algorithm. However, due to the noisy alignment and pivoting process, it is observed that the likelihood of this event decreases as more paraphrased references are added. Therefore, a more useful approach might be one that ensures that the paraphrasing is performed in a constrained manner, wherein the constraints are designed to increase the likelihood of the paraphrased reference matching the translation output. One way of constraining the paraphrasing is by "targeting" it, in some fashion, towards the translation output. The rest of this chapter describes how such targeting can be incorporated into the sentential paraphraser and what such incorporation entails. For convenience, the unconstrained sentential paraphraser, as described in Chapters 3 and 4, is heretofore termed as the "untargeted" paraphraser.

5.1 Learning from HTER

Rather than coming up with an entirely new method for targeting the paraphrasing to the translation output, it is more prudent to base the heuristic on an already existing and successful example of such targeting in the SMT literature. The example used here is one that has been used in the world of machine translation evaluation where a recent trend has been to move from automatic metrics such as BLEU and TER to human-in-the-loop measures. The following paragraphs describe this trend in more detail and provide an intuitive explanation of how it fits with the targeting that is required for the paraphraser.

Translation Edit Rate (TER) is one of the most popular metrics used to measure the quality of the translation output produced by a machine translation system. TER is computed as the minimum number of edits needed to change a hypothesis so that it exactly matches one of the references, normalized by the average length of the references. Since the process is concerned with the minimum number of edits needed to modify the hypothesis, only the number of edits to the closest reference (as measured by the TER score) is measured. Possible edits include the insertion, deletion, and substitution of single words as well as shifts of word sequences.

However, the acceptability of a translation hypothesis cannot be entirely indicated by the TER score, which ignores notions of semantic equivalence. Therefore, a translation hypothesis containing the phrase *rise in unemployment* could be unfairly penalized even if the corresponding phrase in the reference is the semantically equivalent phrase *increase in joblessness*. This is where HTER (Human-targeted

Translation Edit Rate), a modified version of TER that employs human annotation enters the picture. HTER is the official evaluation measure used for the GALE research program (Olive, 2005) and has been shown to yield higher correlations with human judgments than BLEU (Snover et al., 2006). HTER involves a procedure for creating targeted references. In order to accurately measure the number of edits necessary to transform the translation hypothesis into a fluent target language sentence with the same meaning as the references, one must find the closest possible reference to the hypothesis from the space of all possible fluent references that have the same meaning as the original references.

To compute HTER, human annotators—who are fluent speakers in the target language—start from the original, untargeted reference and edit it to make it closer to the given translation hypothesis, *without* losing its original meaning. TER is then computed against this targeted reference and is referred to as the HTER score of the translation output.[2]

Given this description of the HTER computation process, the analogy between an HTER annotator and the sentential paraphraser is obvious. The annotator is essentially a "manual" sentential paraphraser; one that paraphrases the original reference by targeting to the translation output. The result is a new, *semantically equivalent* reference that has a higher likelihood of matching that output. By using this reference in place of the original reference, the translation output will not be unfairly penalized for using words that mean the same thing as the original refer-

[2]Usually the HTER instructions ask the annotator to do the reverse: edit the translation hypothesis to make it fluent and have the same meaning as the original reference; however, the process is essentially symmetric.

ence but are different on the surface. Therefore, it would be a reasonable strategy to fashion the sentential paraphraser to target the translation output in a similar manner. That is, the goal—of making the n-gram matching process of the parameter tuning algorithm fairer by providing additional references—would be better served by creating *targeted* paraphrases of the original reference, that are guaranteed to be closer to the translation output, rather than creating *untargeted* (and, in a way, random) paraphrases and hoping that they turn out to be useful.

5.2 Targeting Implementation

This section describes various details that must be taken into account when determining how to implement targeting for the sentential paraphraser. First, it presents modifications that need to be made to the way in which the sentential paraphraser is employed as part of the SMT tuning loop. Similar modifications need to be made to the way in which the paraphrases are generated for use in tuning. Next, the actual formulation of the targeting is outlined as it is implemented in the paraphraser and used in the rest of this thesis. Finally, other considerations are described that are important for the targeted paraphraser to function effectively.

In previous chapters, the paraphraser was *not* part of the tuning loop but instead only used as an offline component: the original human reference was paraphrased externally and the (untargeted) paraphrased references were added to relevant files of the tuning set. However, for a paraphraser that needs to target the translation output, the paraphrasing needs to happen online, i.e., *during* the pa-

rameter tuning. Here are several important changes that need to be made to the paraphraser and its role in the tuning process:

1. Given that the paraphraser is targeting to the translation output, the output itself needs to be passed as input to the paraphraser in addition to the original reference.

2. In almost all SMT systems (including the one used throughout this thesis), the output of the decoder when translating a given source sentence is an n-best list instead of a single hypothesis. In a way, each of the translation hypotheses in the n-best list can be considered to be a "different" translation output. The targeting implementation must be capable of handling all of these outputs. Therefore, each translation output must be separately targeted, i.e., for each sentence, the paraphraser must create a *different* paraphrased reference targeted to each of the hypotheses in the n-best list.

3. Usually, the algorithm that is used to tune the parameters of an SMT system is run for multiple iterations until some pre-stipulated convergence criterion is met. The tuning algorithm is described in detail in Chapter 3 but I reiterate here for the purpose of exposition. Each iteration of the tuning algorithm consists of (a) decoding the source sentences from the tuning set with the weights chosen at the end of the previous iteration, (b) scoring the n-best list against the references[3] (using BLEU) and, (c) performing a guided search over the space of weights to find the set of weights that maximizes the chosen objective

[3]While it is certainly possible to use other automatic metrics for the purpose of parameter tuning, BLEU remains the most commonly used metric and one that is used in this thesis.

function over the tuning set. It is obvious that the n-best list of translation outputs (the targets) will differ from iteration to iteration. Therefore, the paraphraser needs to be incorporated into the iterative loop of the tuning algorithm so that it can produce new targeted references during each tuning iteration.

The most optimal way to implement targeting for the sentential paraphraser is to leverage the log-linear nature of the paraphrasing model. Chapter 3 defined the paraphraser as essentially an English-English translation model and since the translation model is log-linear, targeting can simply be implemented as a feature in that model. A targeting feature is defined as follows:

$$h(\hat{\mathbf{e_h}}, \mathbf{T}; \mathbf{e_r}, \mathbf{f}) = \sum_{w \in \hat{\mathbf{e_h}} \wedge w \notin \mathbf{T}} 1 \tag{5.1}$$

where $\hat{\mathbf{e_h}}$ is a complete paraphrase hypothesis, $\mathbf{e_r}$ is the original reference that is being paraphrased and $\mathbf{T}$ is the translation output for the source sentence $\mathbf{f}$. The value of the targeting feature for a complete paraphrase hypothesis, as defined, is simply the number of words that are in the paraphrase hypothesis but that are *not* in the translation output (for that particular source sentence). Of course, this feature cannot be pre-computed for any translation rules (or phrase pairs) and is computed by the decoder as the paraphrase hypothesis is being constructed. Note also that given this definition of the feature, its weight must be negative in order to elicit targeting behavior.

Now that the formulation of the targeting feature has been described, the next

question that must be answered is how the weight for this feature will be computed. An obvious answer may be that the weight for this feature should be computed in the same way as the weights for the other paraphraser features were determined in Chapter 3. If the reader will recall, the weights were tuned by, once again, cleverly leveraging the fact that the sentential paraphraser is nothing more than an English-English SMT system. A tuning set was chosen for which there were four reference translations available, each created by a different human translator. One of the reference translations was randomly chosen to be the "source" (the English sentence to be paraphrased) and the other three as "reference paraphrases". Once this setup was complete, the exact same tuning algorithm and tuning criterion used in a bilingual SMT system could be employed for the monolingual SMT system, i.e., the paraphraser. However, as shown below, this strategy will *not* carry over to the targeted paraphraser even though targeting is realized as simply another feature in the model.

Recall that the objective of the tuning algorithm is to find the set of feature weights that maximize the BLEU score of the tuning set against the provided reference translations (paraphrases). In non-quantitative terms, the objective is to produce translations (paraphrases) as close to the reference translations (paraphrases) as possible. Therefore, the tuning algorithm has no reason to learn a non-zero weight for the targeting feature that is, in fact, *designed* to create paraphrases that are closer to an altogether different utterance: the translation output. One could, perhaps, modify the tuning criterion to be a weighted linear interpolation of two quantities: a measure of closeness to the reference paraphrases and a measure of closeness to

the translation output. However, this solution is less than ideal and, instead, I have designed and implemented a different method for choosing the weight for the targeting feature. This method, along with the rationale behind it, is discussed in Section 5.2.3. The experimental results obtained and presented later in this chapter validate the method.

The final issue that must be examined is one pertaining to the original human-authored reference translation that is the input to the paraphraser. For HTER computation, the original reference would be discarded and the human-targeted reference, created by the annotator, would be used in its place for computing the TER score. That scenario guarantees that the targeted reference translation will be *better* than the original translation because there is a trained annotator that creates it in accordance with strict guidelines. This new reference has, by virtue of real human annotation on it, all the "good" (already matching) n-grams from the original reference and none of the "bad" (semantically equivalent to the translation output but not matching on the surface) n-grams.

However, when using the targeted paraphraser, one cannot have the same level of confidence in the targeted reference because of the simple fact that the automatic paraphraser is less then perfect. Therefore, it is too risky to discard the original reference and, therefore, the targeted reference must be used in combination with the latter for tuning purposes.[4]

At this point, a reasonable method exists for incorporating targeting into the

[4]Looking at this from another angle, even if the original reference was made entirely redundant by the targeted reference, including it for tuning would have no significant impact on the computational effort involved.

sentential paraphraser and creating targeted paraphrased references for the tuning algorithm, instead of the untargeted paraphrased references employed in Chapter 4. However, the picture is not complete because one crucial aspect of the HTER computation process has not been addressed! The targeted reference created by an HTER annotator must be *semantically equivalent* to the original reference, i.e. it must have the same meaning as the original reference even if it uses the words from the translation output. Are there any guarantees that the paraphraser, in its targeted incarnation, also preserves the meaning of the original reference that it paraphrases? The next section answers this question in detail.

5.2.1 Preserving Semantic Equivalence More Strongly

This section presents a theoretical and empirical examination of whether the targeted sentential paraphraser is sufficiently equipped to retain the meaning of the original reference. To reason about this, Figure 5.2 clearly illustrates the differences between the untargeted and the targeted scenarios using "meaning diagrams".

Figure 5.2(a) shows the untargeted paraphrasing scenario. In this scenario, there is an explicit meaning preserving link between the original reference and the untargeted paraphrase, i.e., the paraphraser attempts to preserve the meaning of the original reference. The color of the node representing the untargeted paraphrase indicates the now understood fact that the paraphraser is only marginally successful at preserving the original meaning, given the noisy paraphrase induction process. There is also an implicit link between the original reference and the translation

output since they are both renderings of the same source language utterance. Note, however, that there is *no* link between the paraphrase and the translation output.

In the targeted paraphrasing scenario, as represented by Figure 5.2(b), the other links are the same as in the untargeted scenario except that there is now an additional, explicit link between the generated paraphrase and the translation output due to the targeting. As the color of the paraphrase node indicates, the targeted paraphrase is now semantically related to both the original reference *and* the translation output, albeit via different mechanisms. However, I investigate the following hypothesis:

Hypothesis. The existing semantic link between the original reference and the targeted reference—obtained via the application of pivoted paraphrase rules—is no longer adequate with the incorporation of the targeting feature.

Empirical results are presented in the next paragraph that validate the above hypothesis but a logical argument is presented first. In the untargeted scenario, the paraphrase generation process and the translation process are completely independent. Therefore, the additional artificial references, while extremely noisy, do not bias the tuning process in any way. However, in the targeted scenario, the existence of an explicit link between the paraphrase and the translation output provides an opportunity for the introduction of a systematic bias into the tuning process; the targeted paraphrases use words from the translation output but are also used as additional references against which the translation output is matched. Given the noisy nature of the existing semantic link between the original reference and paraphrase, it will not be adequate to counter this systematic bias.

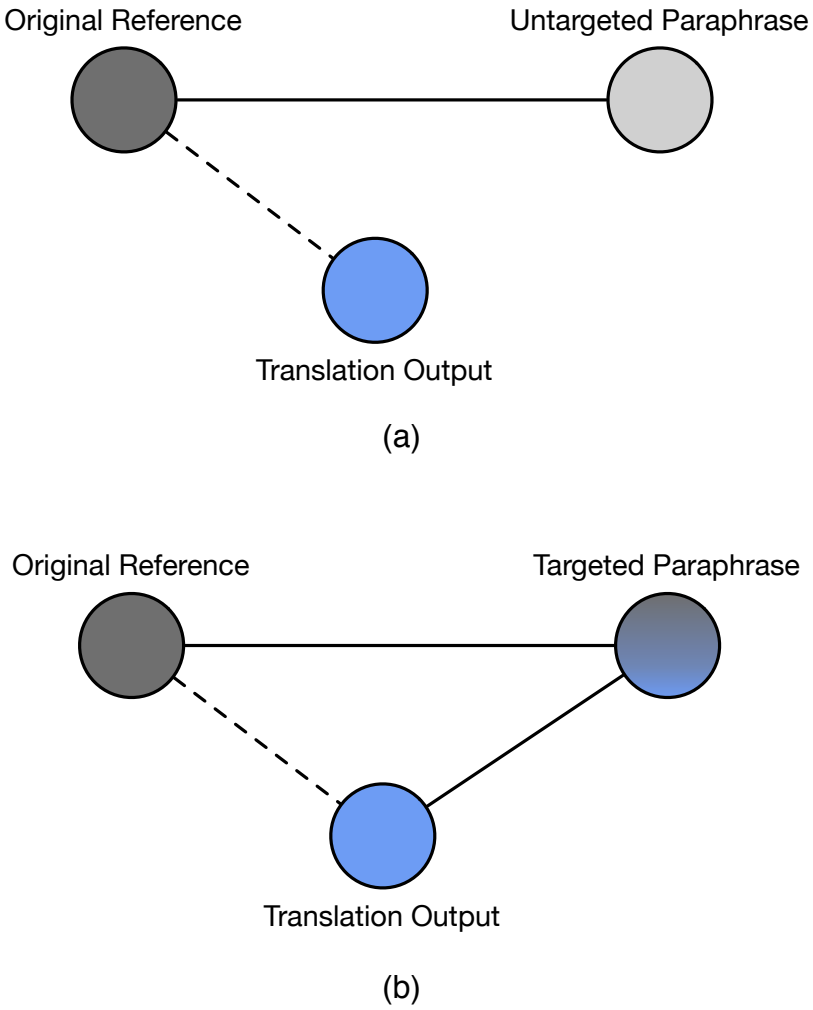


Figure 5.2: Meaning diagrams illustrating the differences between the (a) untargeted and (b) targeted paraphrasing scenarios.

At this point, empirical results are provided that confirm the above logical reasoning and validate the hypothesis. To do so, the targeted paraphraser is used—with only the previously motivated pivot-based semantic link between the paraphrase and the original reference—for tuning the weights of an SMT system and it is shown that the obtained results point to a biased tuning process.

Before the targeted paraphraser is used for tuning, a weight for the targeting feature must be chosen. Since a principled method by which choice must be made

has not yet been introduced, the weight is chosen manually for this experiment. As this is simply an experiment designed to show that using a "sufficiently well-targeted" paraphraser for parameter tuning will lead to biased results, choosing the weight manually is a reasonable option. To choose the targeting feature weight, consider the histogram shown in Figure 5.3. Each bar in this histogram corresponds to a different targeting feature weights and represents the TER value for the 1-best targeted paraphrases for 100 randomly chosen sentences from the NIST MT03 Chinese dataset. The TER values are computed against the translation output generated by an SMT system whose parameters were tuned using only the original reference.

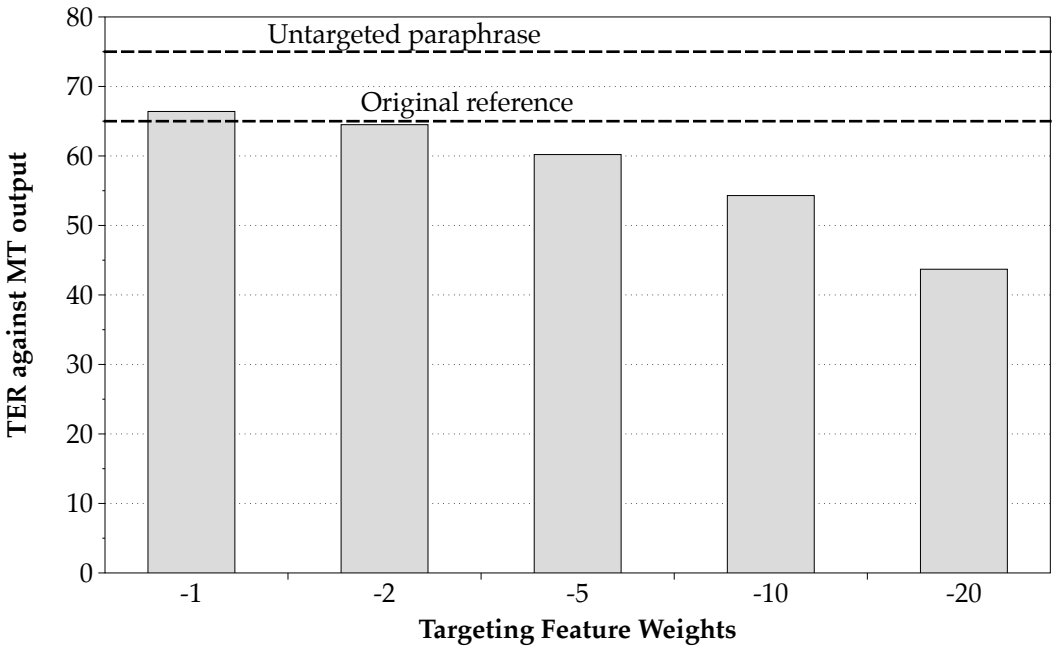


Figure 5.3: A histogram showing the TER values for the 1-best targeted paraphrases of 100 randomly chosen sentences from the Chinese NIST MT03 dataset. The TER values are computed against the translation output for the same sentences, that were also the corresponding targets. Lower TER values indicate more effective targeting.

The TER value corresponding to each weight can be thought to represent

how effective that weight is for targeting: stronger feature weights lead to more targeted paraphrases and, hence, lower TER values. For comparison, the graph also shows the TER values of the original reference and the 1-best untargeted paraphrase against the MT output. For the paraphrases to be "sufficiently well-targeted," the TER value must be significantly lower than the untargeted paraphraser as well as the original reference. Therefore, the weights of -10 and -20 are good choices for the proposed tuning experiment.

Figure 5.4 shows the results of the experiment using the targeted paraphraser with weights for parameter tuning chosen to be -10 and -20 for Chinese-English translation. The paraphraser was used in the fashion as described in Section 5.2 and the 1-best targeted paraphrase was used as an additional reference, in combination with the original reference.

The graph shows two sets of curves, one each for the two different targeting weights. Each set contains two curves: one showing the BLEU score obtained for the tuning set after each iteration of tuning and the other showing the BLEU score for the validation set. The tuning set used for these experiments was, as before, NIST MT03 (919 sentences) and the validation set was NIST MT04+05 (2870 sentences). For both feature weights, the curves verify the above hypothesis. The BLEU score on the tuning set obviously increases given that one of the references is designed to look like the translation output. However, since this increase is motivated more by a biased reference than by an improved set of feature weights for the translator, the BLEU score on the validation set degrades after each iteration of the tuning process. Note that the degradation is self-sustaining: as poorer sets of weights

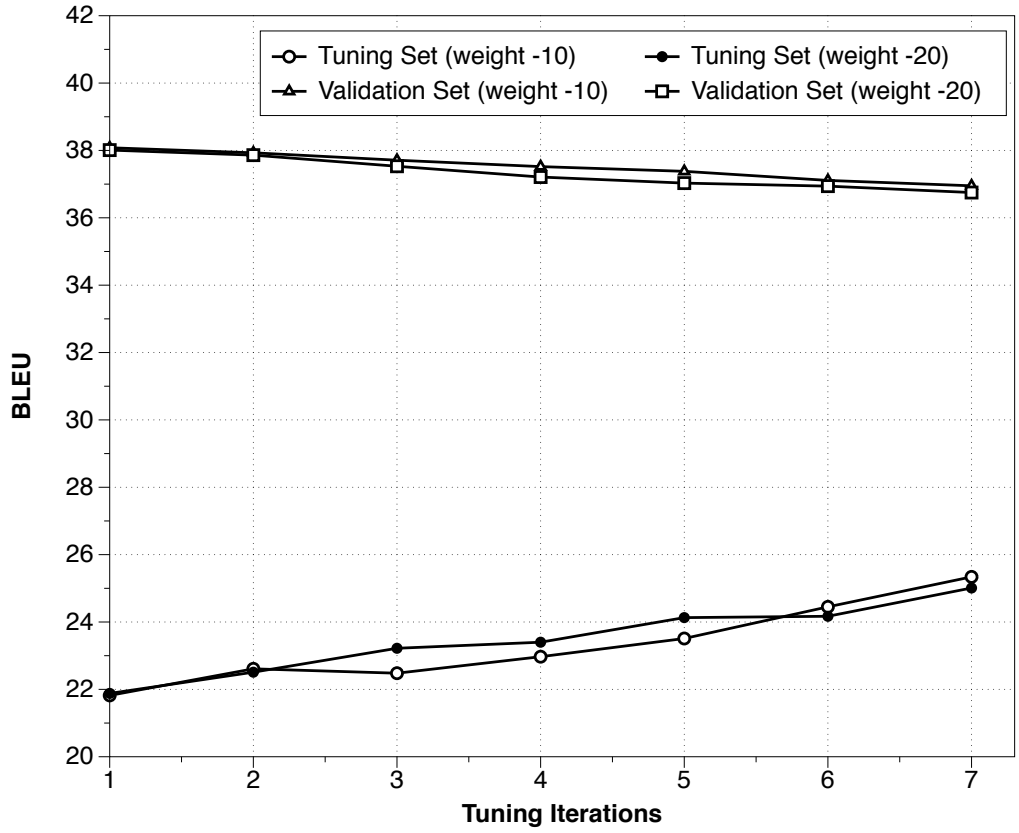


Figure 5.4: A graph showing the BLEU score for each iteration of the tuning algorithm employing targeted paraphrased references generated by using two different targeting feature weights. For each weight, the BLEU score curves for the tuning set (MT03) and the validation set (MT04+05) are shown.

are chosen based on a biased match, the translation output produced is worse in quality and, therefore, the targeted references retain even less of the meaning of the original reference. Had the set of references been completely independent of the translation output—as is generally the case—the tuning process would have automatically detected the worsening quality of the translation output and tried to move away from that part of the parameter space. However, given the strong link between the targeted references and the translation output, the tuning algorithm is unable to break the feedback loop.

Therefore, it is now clear that the existing semantic link between the original reference and the targeted paraphrases is not sufficient to counter the bias that may be introduced into the tuning process if the targeted paraphrases were used as additional references. A stronger explicit mechanism to retain the meaning of the original reference is necessary. The next section describes such a mechanism.

5.2.2 The Self-Paraphrase Bias

In this section, a mechanism is described wherein the automatic paraphrases of the human-authored reference are able to better retain its original meaning in the targeted scenario. The crux of the mechanism is that of redistribution of probability mass so that a pre-stipulated portion of the probability mass is devoted to "self-paraphrasing". In other words, a probabilistic bias is introduced into the paraphrasing process such that in the space of all possible paraphrases for a given human-authored reference, a fixed amount of probability mass is reserved for that paraphrase which is identical to the input, hence retaining *all* of its meaning. The following paragraphs describe the mathematical details of this bias.

As described in Chapter 3, the sentential paraphraser works by applying hierarchical paraphrase rules. The paraphrase rules are generated by pivoting bilingual hierarchical rules that are extracted from a parallel corpus. Each paraphrase rule, like each translation rule, has feature values associated with it, computed either as part of the pivoting process from the corresponding features of the bilingual rule or based on either the length or the content of the paraphrastic phrases themselves.

Out of such features, the three primary probabilistic features are:

1. The probability of the paraphrastic phrase given the original phrase, $p(e_2|e_1)$
2. The probability of the original phrase given the paraphrastic phrase, $p(e_1|e_2)$
3. The joint probability of the original and the paraphrastic phrase, $p(e_1, e_2)$

Chapter 3 describes how each of these values is computed along with examples. Since the pivoting process yields proper probability distributions, some probability mass is obviously already assigned to self-paraphrasing. This mass can be computed by the following formula:

$$p(e_1|e_1) = 1 - \sum_{e' \neq e_1} p(e'|e_1) \tag{5.2}$$

The point of the self-paraphrase bias is to rescale this probability such that for *every* original English phrase in the corpus, the *same* specified amount of probability mass is devoted to self-paraphrasing. The process by which the conditional and marginals are so rescaled is explained below using an example.

Let the English phrase that is to be paraphrased be e_1 and assume that as part of the pivoting process the self-paraphrase probability is $p(e_1|e_1) = 0.05$. Say that the self-paraphrase bias is designed to be 0.5, i.e., among all possible paraphrases of e_1, 50% of the probability mass should be devoted to the self-paraphrase e_1. Then, the conditional is simply rescaled as follows:

$$p_{\text{rescaled}}(e_1|e_1) = p'(e_1|e_1) = p(e_1|e_1) * \frac{0.5}{0.05} = 0.5 \tag{5.3}$$

Obviously, since the self-paraphrase probability is being increased, the probability values for all the other paraphrases, excluding the self-paraphrase, must be correspondingly scaled down. Considering the total probability mass that the pivoting process assigns to all phrases other than the self-paraphrase e_1:

$$p(e_2|e_1) + p(e_3|e_1) + \cdots + p(e_k|e_1) = 1 - p(e_1|e_1) = 1 - 0.05 = 0.95 \tag{5.4}$$

After the introduction of the self-paraphrase bias and the rescaling of the conditional as in Equation 5.3, the mass devoted to all phrases other than the self-paraphrase is:

$$p'(e_2|e_1) + p'(e_3|e_1) + \cdots + p'(e_k|e_1) = 1 - p'(e_1|e_1) = 1 - 0.5 = 0.5 \tag{5.5}$$

Therefore, all the other conditional paraphrase probabilities can be rescaled simply as:

$$p'(e|e_1) = \frac{0.5}{0.95} * p(e|e_1),\ \forall e \neq e_1 \tag{5.6}$$

To determine the rescaling factors for the joint probabilities, the marginal distribution $p(e_1)$ must first be considered. Bayes' rule states that:

$$p(e_2|e_1) = p(e_1|e_2) * \frac{p(e_2)}{p(e_1)}$$

Since the only quantity being changed is the self-paraphrase probability of the phrase e_1, it stands to reason that the quantities $p(e_1|e_2)$ and the marginal $p(e_2)$ are not

affected. Therefore, the only changed quantity in the Bayes law expression is the marginal $p(e_1)$. One can also think of this another way: since the conditional probability that e_1 is produced by paraphrasing e_1 is being increased, the marginal must also increase. Denote this new marginal by $p_{\text{rescaled}}(e_1)$ or $p'(e_1)$, as was done for the conditional. Given this, Equation 5.5 now becomes:

$$p(e_1|e_2) * \frac{p(e_2)}{p'(e_1)} + p(e_1|e_3) * \frac{p(e_3)}{p'(e_1)} + \cdots + p(e_1|e_k) * \frac{p(e_k)}{p'(e_1)} = 0.5 \tag{5.7}$$

By similarly expanding Equation 5.4 and some simple algebraic manipulation, the following ratio between the rescaled marginal and the original marginal can be obtained:

$$p'(e_1) = p(e_1) * \frac{0.95}{0.5} \tag{5.8}$$

Given these rescaling factors for the conditional and the marginal distributions, it is easy to compute the scaling factors for the joint distribution with the self-paraphrase e_1:

$$p'(e_1, e_1) = p'(e_1|e_1) * p'(e_1) = \frac{0.5}{0.05} * \frac{0.95}{0.5} * p(e_1, e_1) = \frac{0.95}{0.05} * p(e_1, e_1) \tag{5.9}$$

and for the joint distribution with all the other paraphrases $e \neq e_1$:

$$p'(e_1, e) = p'(e|e_1) * p'(e_1) = \frac{0.5}{0.95} * \frac{0.95}{0.5} * p(e_1, e_1) = p(e_1, e) \tag{5.10}$$

At this point, the self-paraphrase biasing procedure for the phrase e_1 is complete.

To be applicable at a global level, the rescaling process is carried out for *all* English phrases.

To determine whether adding this self-paraphrase bias has any effect on the generated paraphrases, a 25% self-paraphrase bias is added to the targeted paraphraser (with a sufficiently effective targeting feature weight, say, -12) and the resulting paraphrases are examined.[5] Figure 5.5 shows the 1-best targeted paraphrase generated with these attributes, for a particular human-authored reference and compares it to the paraphrase without any self-paraphrase bias. It is clearly evident that the inclusion of the self-paraphrase bias leads to paraphrases that retain more of the original meaning as expected.

Original Reference	Around the site of the explosion, 40 shops and 15 cars were damaged.
Translation Output	The blast destroyed 40 shops and 15 cars in the vicinity.
Untargeted Paraphrase	Close to the place of the bomb, 40 stores and 15 vehicles have been ruined.
Targeted Paraphrase	In the vicinity of the site of the explosion, 40 shops and 15 cars were damaged.

Figure 5.5: Showing the difference between the 1-best untargeted paraphrase and the 1-best targeted paraphrase for the original reference. It is clear that the targeted paraphrase with incorporated self-paraphrase bias not only retains more of the meaning of the original reference but is also closer to the MT output. The sentence was manually chosen from NIST MT03 for illustration purposes.

[5]While it may seem like these values are chosen randomly, they are actually chosen by the method described in the next section.

5.2.3 Finding the Balance: Self-paraphrase Bias vs Targeting

Now that a procedure has been outlined that allows explicit preservation of more of the meaning in the original reference, it must be pointed out that the two primary mechanisms involved in targeted paraphrases—the targeting feature and the self-paraphrase bias—are at odds with each other. The goal of the targeting feature is to *change* the original reference by using words from the translation output. On the other hand, the goal of the self-paraphrase bias is to *retain* as many of the words in the original reference as possible. At first this tension might seem like a problem. However, in this section, a technique is described—one that actually leverages this tension—to determines the targeting feature weight and the self-paraphrase bias value.

Before this technique is described, picture the original reference and the translation outputs as points in the vast space of possible "reference" translations that can be produced for a given source sentence. If the SMT system were perfect, then the translation output would be a perfectly valid reference translation. However, as it stands, it is not a very useful reference translation since it is obviously not correct. Obviously, the original reference is a good reference in that it is correct. However, it has no a priori likelihood of matching the translation output even if the output may be correct. Therefore, in this sense, the original reference is also not a very useful reference translation. The point of paraphrasing the original reference is to try and come up with more useful references that lie in this space of reference translations. If untargeted and targeted paraphrasing of the original references are compared in

the context of this picture, it can be inferred that:

- Untargeted paraphrasing results in reference points that are "random," i.e., they are created by changing words and phrases in the original with no regard to the translation output. The only reason a new reference point is useful is due to the sheer number of changes: a large percentage of the words in the original reference are changed and some of them happen to match the words in the translation output. Indeed, the noisy nature of the changes overwhelms any utility as more and more of these new reference points are used for tuning. Similar to the original reference, the untargeted paraphrases have no a priori reason to be closer to the MT output since it does not influence the paraphrasing in any way.

- Targeted paraphrasing, on the other hand, is designed to produce reference points that are closer to the translation output but, at the same time, do not sacrifice the meaning of the original reference. This form of paraphrasing comes closer to achieving the true goal of using paraphrasing to create additional reference translations.

With this picture in mind, the concept of "distance" in this space of reference translations can be introduced. All the new reference points that are created via paraphrasing (untargeted or targeted) can be assumed to be at some distance from both the original reference and the translation output, that are themselves points in this space. Note that this space is more conceptual than Euclidean. Therefore, this proposed distance is only required to satisfy a simple mathematical requirement: a

lower distance between two points should indicate that they are semantically closer and a larger value should indicate the opposite. Note that such a measure has been used earlier in this section when determining which targeting weights are the most effective at targeting: TER. Therefore, the distance measure between the points in this space will simply be the TER value.

Two such distances in this space of reference translations are most important:

1. The *distance* (in terms of TER) from the original reference (d_{ref}), and

2. The *distance* (in terms of TER) from the MT output (d_{mt}).[6]

These two distances can be used to verify the claim about untargeted paraphrasing not producing references that are closer to the MT output. 100 sentences are randomly sampled from the NIST MT03 Chinese data set and run through the untargeted paraphraser to generate the n-best lists of paraphrase hypotheses for these 100 sentences (here n=300). The two TER values mentioned above are computed for this n-best list. Figure 5.6 puts these distances in context by also showing the same distances for the starting point in the space—the original reference. As seen in this figure, the original reference is about 65 TER points away from the MT output. When run it through the paraphraser though, the new paraphrased reference is not only *very far away* from the original reference (indicating the sheer number of changes made) but also *further* away from the MT output than the starting point.

[6]Note that this "distance" has already been used in Figure 5.3 albeit for a different purpose.

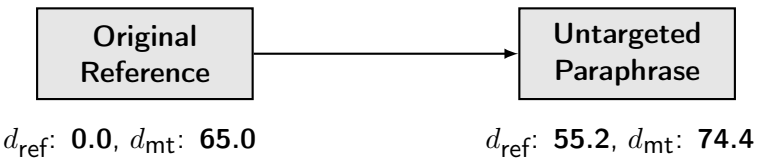


Figure 5.6: A picture illustrating that the untargeted paraphraser not only makes a lot of changes to the original reference (d_{ref} goes from 0 to 55) but also produces references that are *further away* from the MT output (d_{mt} goes from 65 to 74). The distance (TER) values are computed for 100 randomly chosen sentences from the NIST MT03 Chinese dataset.

These same two distances can be used to find the balance that is sought between the targeting and the self-paraphrasing. The nature of this balance must be such that while the latter disallows many changes to the original reference, any changes that are allowed should create new points in the reference space that are closer to the MT output. In terms of these two distances, this requirement can be stated as follows: the magnitude of the movement of the new reference point *away* from the original reference (i.e., the positive change in d_{ref}) should be approximately equal to the magnitude of the movement of the reference point *towards* the MT output (i.e., the negative change in d_{mt}). At this point, it would be worthwhile examining whether this behavior is actually even exhibited by the targeted paraphraser. Figure 5.7 represents such an examination. For this examination, it is again stipulated that all that is required is that the paraphraser be sufficiently well-targeted; a weight of -10 for the targeting feature weight satisfies this requirement, according to Figure 5.3. This figure shows four different points in the reference space each created with the same targeting feature weight (-10) but with four different self-paraphrasing bias values. For each point, values are also shown for d_{ref}, d_{mt} and the respective changes in those values (shown in parentheses) compared to the

original reference. It can be seen that while the changes in the two distances are certainly not equal, they do seem to approach each other as the bias value increases.

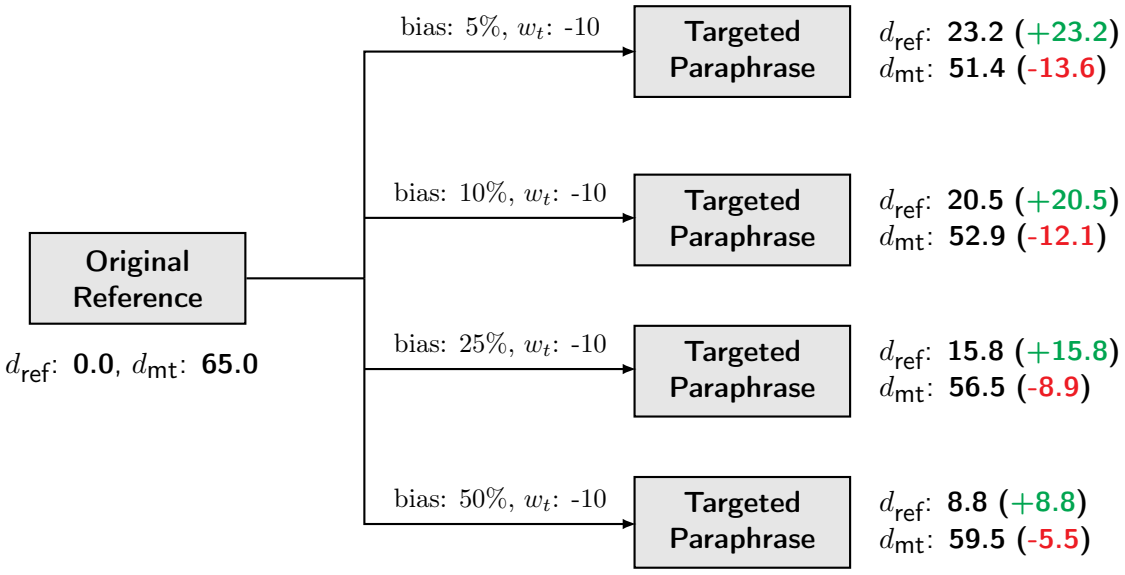


Figure 5.7: A picture illustrating that the targeted paraphraser can create useful new points in the reference space starting from the original reference. Points are useful if they are not too far away from the original reference (indicated by the change in d_{ref}) but also move closer to the MT output by approximately the *same* amount (indicated by the change in d_{mt}). Using a fixed but effective targeting weight of -10 in combination with 4 different bias values, it is seen that this behavior is indeed exhibited, to an extent.

Since the targeted paraphraser can likely be configured to behave the way as desired in the reference space, all that remains is a way to determine precisely what the optimal configuration should be. Note that this desired configuration is a *combination* of two quantities—the targeting feature weight and the self-paraphrase bias—and, therefore, the obvious search technique that can be applied to this problem is a grid search. For this search, the targeting feature weight is varied along one dimension and the self-paraphrasing bias along the other. For each pair of values, an n-best paraphrase list is generated and the difference $D = |\Delta d_{\text{mt}} - \Delta d_{\text{ref}}|$ is computed. In this expression, Δd_{ref} represents how much further away the new point

in the reference space (as represented by the output of the paraphraser instantiated with that specific feature weight and that specific self-paraphrase bias) is from the original reference. Similarly, Δd_{mt} represents how much the new point has moved closer to the translation output. As mentioned before, the ideal scenario would be that all movement away from the reference is converted to movement towards the MT output. Therefore, the optimal point on the grid would be one where this expression would be closer to zero.

Figure 5.8 shows the grid search process for Chinese. For each point $[p, w]$ on the grid—a specific targeted paraphraser configuration with -w as the targeting feature weight and p as the self-paraphrasing bias—D is computed over the paraphrase n-best list for 100 randomly chosen sentences from the NIST MT03 dataset. Given that the ideal point is one with D as close to 0 as possible, the point on the grid that best matches that criterion is (25, 14).

- **Why not choose the point (50,10)?** Technically speaking, the point on the grid with the lowest value of D is actually (50, 10). However, that point is not ideal since the targeted paraphrases produced with that configuration will hardly be any different from the original references themselves (as indicated by the value of Δd_{ref}). They will certainly not be targeted enough to the translation output (as indicated by the value of Δd_{mt}).

- **Why restrict the targeting feature weight to the range [10, 20]?** This decision is simply based on observing the targeting behavior of the paraphraser for the tuning set. It was observed that for any weight weaker than -10, the

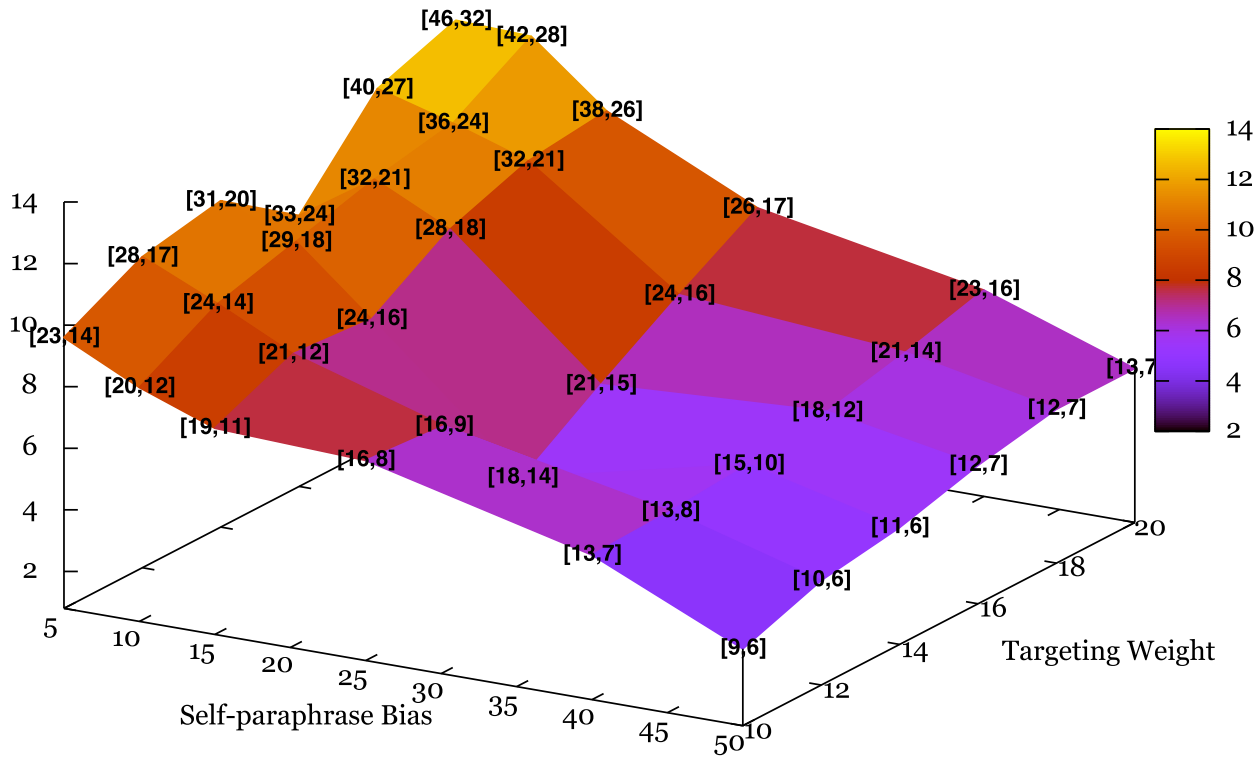


Figure 5.8: A color-mapped surface plot showing the grid search for 100 random sentences chosen from the NIST MT03 dataset. This plot is useful because it is able to convey multiple information elements for a single grid point. For each combination of targeting feature weight and self-paraphrasing bias, the quantity plotted is $D = |\Delta d_{\text{mt}} - \Delta d_{\text{ref}}|$. The color of the plot indicates the magnitude of D. In addition, each grid point is annotated with the tuple $[\Delta d_{\text{ref}}, \Delta d_{\text{mt}}]$. Given the definition of a useful point in the reference space, the ideal combination is the grid point (25,14).

targeting was not very effective. Similarly, at the other end of the interval, it was noted that for any weight stronger than -20, the targeting overwhelms any attempts at maintaining semantic equivalence.

- **Why only 100 sentences?** A subset of 100 randomly chosen sentences was used for the search as opposed to the entire set for two reasons: (a) using

a randomly chosen held out set seems less biased than using the entire set for analysis, and (b) paraphrasing (decoding) 100 sentences instead of 1000 sentences (the size of NIST MT03) is *much* faster. This is a useful amount of time reduction given that 36 different points on the grid are being explored.

- **Why not use an actual held out set?** A randomly chosen subset of the actual set that is to be paraphrased is used rather than another completely different held out set. The reason is simple: measurements need to be made for this specific set in order to paraphrase it usefully. If the search process were more sophisticated than a simple grid-based parameter sweep—a definite avenue for future research—perhaps the parameters chosen based on an actual held out set could be general enough to be used for the set to be paraphrased.

5.3 Putting It All Together

At this point, all the pieces are available for incorporating the notion of targeting into the sentential paraphraser: a reasonably good implementation of the targeting feature, the self-paraphrase biasing mechanism required to offset any systematic bias that may result from using targeted references and, finally, a technique that finds the right balance between these two opposing mechanisms. All that is required now is to put all of these pieces together to achieve improved parameter tuning.

Before the actual tuning process is discussed, it is important to mention the stages of the pipeline that need to be completed before the paraphraser can be

used in tuning. First, five sets of paraphrase rules are generated from the parallel corpora by using the pivoting process; one each for the 5 self-paraphrase bias values of 5%, 10%, 15%, 25%, 40% and 50%. 100 sentences are then randomly chosen from the set T that is to be used for SMT parameter tuning. The grid search process is then carried out, as described in the previous section, to determine the optimal combination of the targeting feature weight and the self-paraphrasing bias. The configuration for the paraphraser is now fully determined. The resulting targeted paraphraser can now be used for SMT parameter tuning.

The actual setup is shown in Figure 5.9. During each iteration of tuning, the source sentences from T are translated with an SMT decoder (whose weights have been tuned with just the original human reference) to produce an n-best list of translation hypotheses. Each hypothesis is then used as the target to produce an n-best list of targeted paraphrases for the corresponding original reference. At this point, rather then choosing the 1-best targeted paraphrases from this list, a reranking process is applied that reranks the various hypotheses in the n-best list, using some additional features. One of these features is the probability score assigned to each targeted paraphrase hypothesis by a higher order—and thus, more informative—n-gram language model (5-gram). Usually such higher order language models are used for reranking an n-best list rather than during the actual decoder search due to more demanding memory requirements.The other feature used by the reranker is the word error rate (WER) between the target (the specific translation hypothesis) and the paraphrase hypothesis.[7] The reranking process can help by moving to the top

[7]An even more useful reranking feature would be the TER between the targeted paraphrase

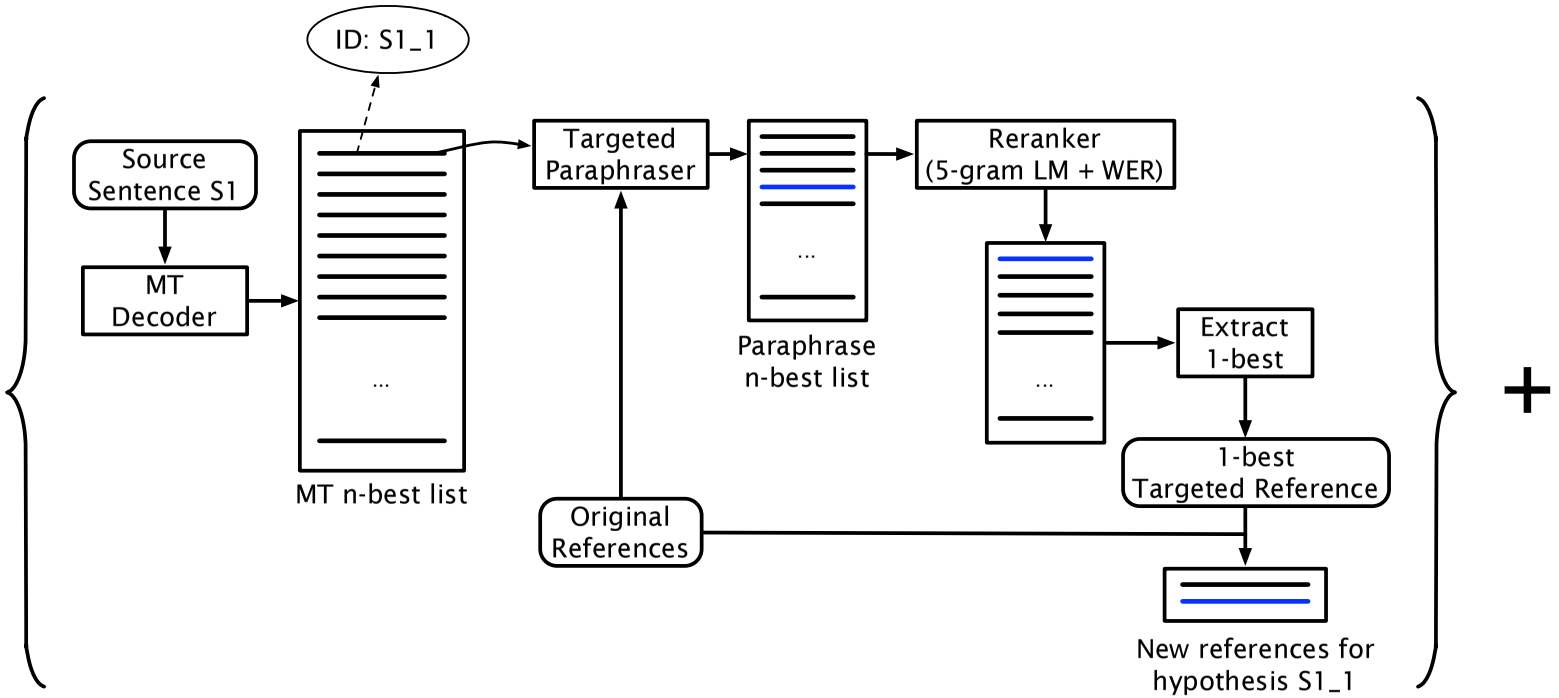


Figure 5.9: A system diagram illustrating the actual setup for using the targeted sentential paraphraser to generate new sets of references which are then used by the SMT parameter tuning algorithm. The new set of references include the original human authored reference and the 1-best paraphrase targeted specifically to that hypothesis. Note that in each tuning iteration, new references are generated for *each* individual translation hypothesis for *each* source sentence in the tuning set. Including the diagram in braces and following with a plus sign indicates that the entire process is repeated multiple times, once for each iteration of tuning.

of the n-best list the targeted paraphrases that are both more fluent (and, possibly, more meaningful) and better targeted. Once reranking is done, the 1-best targeted paraphrase is extracted from the reranked n-best list. This targeted paraphrase is then combined with the original reference and this pair now represents the new set of references for that specific translation hypothesis. The SMT parameter tuning algorithm will now use this new set of references in its search process. Although the figure only focuses on a single source sentence and a single translation hypothesis, new sets of references are actually generated for each translation hypothesis of each source sentence. Furthermore, this entire process is repeated for each iteration of the tuning process with the learned parameters in iteration k feeding forward into iteration (k+1). Generally, about 6-7 iterations of tuning are used. At the end of the last iteration, the learned parameters are taken and used to decode a validation set which is then scored against its own set of human-authored references.

Generally, the complete process would take an inordinately long time—3 weeks—to finish since the paraphraser needs to run for every single translation hypothesis. However, in its current state, it has been factored to run efficiently on a computer cluster containing several hundred nodes and only takes about 36 hours to complete 7 iterations of tuning and decode and score the validation set at the end.

The process of using the sentential paraphraser in a targeted fashion for parameter tuning has now been fully described. In the next section, actual translation

hypothesis and the target. However, note that this feature needs to be computed for each paraphrase hypothesis for each translation hypothesis for each source sentence. Using 300-best lists for both the SMT decoder and the paraphraser and the NIST MT03 Chinese set containing about 1000 source sentences, this would require computing $300 * 300 * 1000$ or 90 million TER values! Given that computing TER is much slower than computing WER, TER was not used.

experiments and results are presented. These confirm that incorporating targeting into the paraphraser indeed makes it more useful for parameter tuning.

5.4 Translation Experiments & Results

In this section, machine translation experiments are described that use the targeted paraphraser to create additional references for parameter tuning. Their results are then compared to results from previous experiments that used either just the original human reference or additional references created by the untargeted paraphraser. Just as in Chapter 3, two kinds of results are shown for each set of experiments. The first compares translation systems to each other using automatic machine translation evaluation metrics BLEU and TER and the second compares them using human judgments.

5.4.1 Chinese-English

The first experiment is for Chinese-English translation. Most of the details of the experiment are identical to the experiments conducted in Chapter 4 but are reproduced here:

1. **Training Data**. The Chinese-English parallel data used for this set of experiments consist of approximately 2.5 million newswire segments. Both the translation and the paraphrase rules are generated using these data. Besides the parallel data, approximately 8 billion words of English text are used for language model (LM) training (3.7B words from the LDC Gigaword corpus, 3.3B

words of web-downloaded text, and 1.1B words of data from CNN archives). These data are used to train two language models: a trigram LM used in decoding, and an unpruned 5-gram LM used in reranking both the SMT and the paraphraser n-best lists. Modified Kneser-Ney smoothing was applied to the n-grams in both cases (Chen and Goodman, 1998).

2. **Decoders**. As with previous experiments, both the SMT and paraphrase decoders (Shen et al., 2010) use a state-of-the-art hierarchical phrase-based translation model where the translation (or paraphrasing) rules form a synchronous context free grammar (SCFG).

3. **Tuning Set**. As the tuning set, the NIST MT03 Chinese set containing 919 sentences is used. The MT03 set actually comes with 4 human authored reference translation. In order to simulate a set with only a single reference, one of the 4 reference translations is randomly chosen for each document in the set.

4. **Validation Set**. The validation set is the NIST MT04+05 set which contains both NIST MT04 and NIST MT05 sets. The total number of sentences in this set is 2870.

The BLEU and TER results for the validation set are shown in Figure 5.10. It is clearly evident from this figure that translations produced by SMT systems that use the targeted sentential paraphraser are significantly better than those produced by systems that use the untargeted paraphraser. For obvious reasons, the performance

of systems that use only the human authored reference translations forms the upper bound for both the paraphrasers.

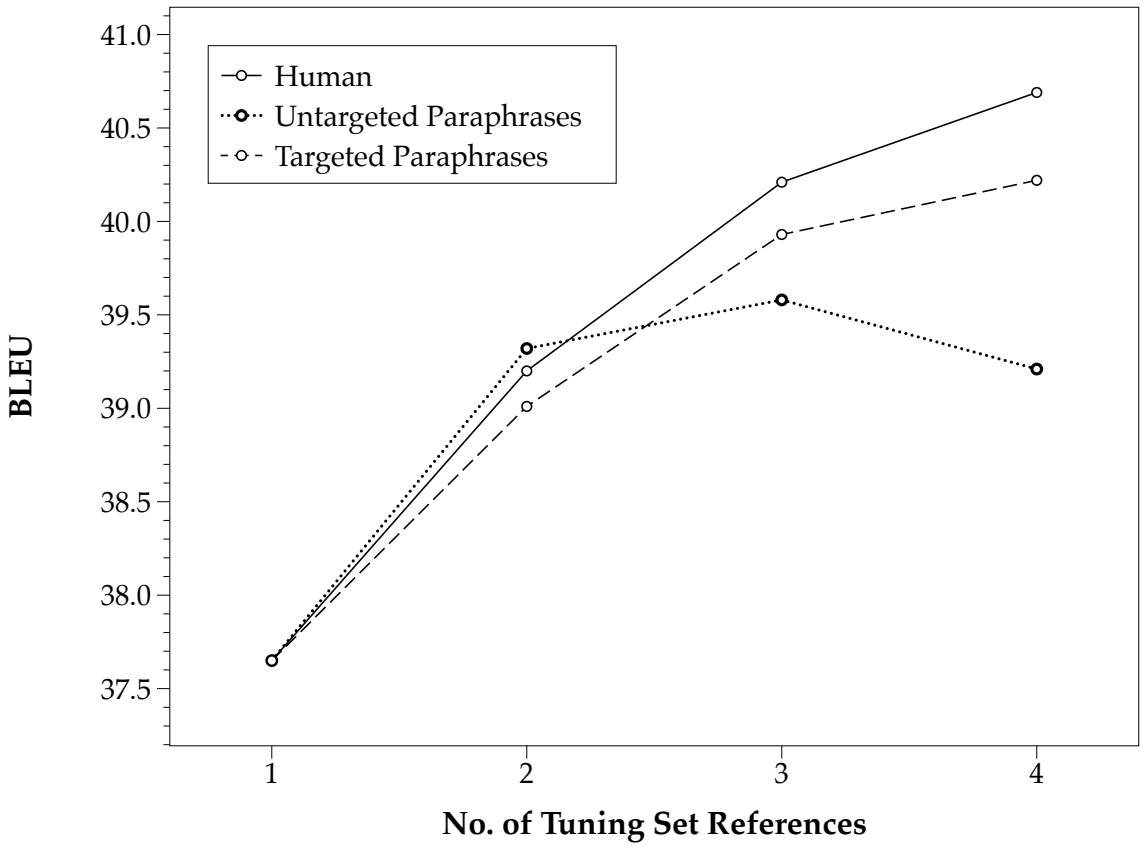
Similar to the human judgment experiments presented in Chapter 4, experiments were conducted to compare the translation outputs produced by an SMT system augmented with 3-best targeted paraphrases of the original human-authored reference to outputs produced by the baseline system using only the single human reference and also to those produced by a system augmented with its 1-best untargeted paraphrase.[8] The experiments were designed using Amazon Mechanical Turk in a similar fashion as those described in Chapter 4 and the results are shown in Figure 5.11.

5.4.2 French-English, German-English and Spanish-English

In this section, experiments and results for translating from three European languages—French, German and Spanish—into English are presented. The details, first described in Chapter 4, are reproduced below:

1. **Training Data**. For these sets of experiments, the training data from the shared translation task of 2009 workshop on machine translation (Callison-Burch et al., 2009) were used. The parallel training data for this task consist mainly of bitexts extracted from the proceedings of the European parliament (Koehn, 2005): 1.7 million sentences for French-English, 1.6 million sentences for German-English and 1.7 million sentences for Spanish-English.

[8] We compare to the system using the 1-best untargeted paraphrase instead of the 3-best targeted paraphrases because, as shown in Chapter 4, doing so yields better performance and, therefore, represents a stronger baseline for the targeted scenario.



# tuning refs	Human		Untargeted		Targeted	
	BLEU	TER	BLEU	TER	BLEU	TER
1 (1H+0)	37.65	56.39	37.65	56.39	37.65	56.39
2 (1H+1)	**39.20**	54.48	**39.32**	54.39	**39.01**	54.88
3 (1H+2)	**40.21**	53.50	**39.58**	53.87	**39.93**	53.71
4 (1H+3)	**40.69**	53.31	**39.21**	54.19	**40.22**	53.43

Figure 5.10: A graph showing the BLEU scores for the set NIST MT04+05 as different types of additional reference translations are added to a single human authored reference translation for the tuning set (NIST MT03). Note that the BLEU score for this validation set is measured against 4 human references in each case. Only the number of references for the tuning set is varied. The corresponding TER scores are shown in the accompanying table.

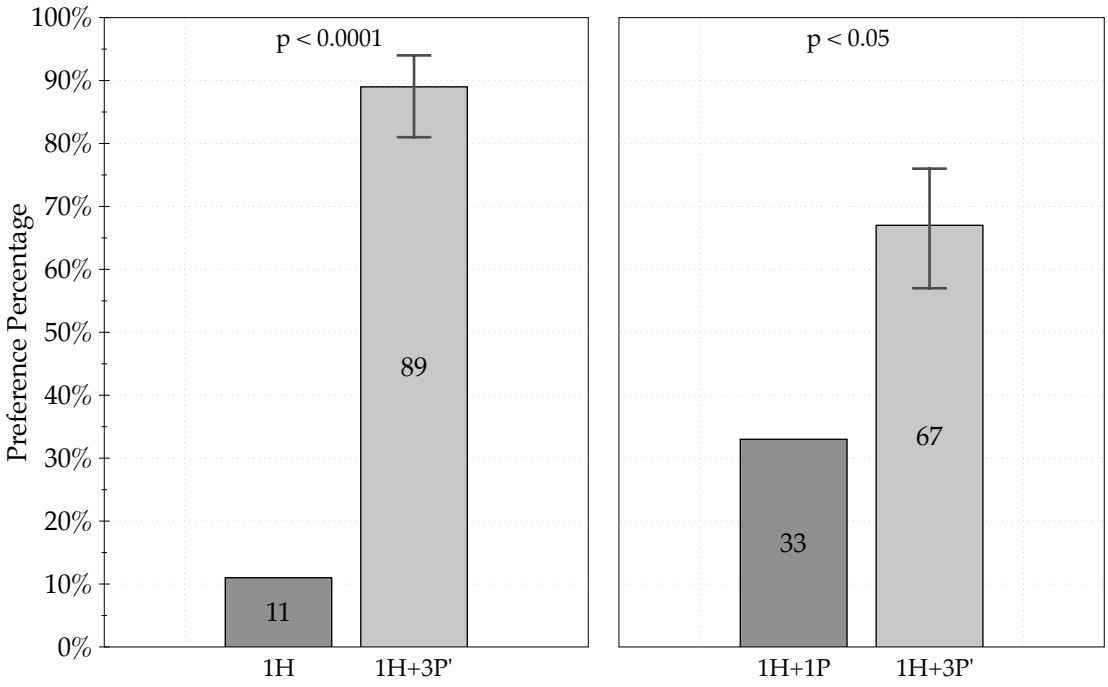


Figure 5.11: When translating Chinese sentences, human subjects on Amazon Mechanical Turk prefer–to a statistically significant extent—the translations produced by the MT system that was tuned with 3-best targeted paraphrases (1H+3P') as additional, artificial references when compared to a system that just uses the human-authored reference (1H). Similarly, they prefer the same system over one that uses untargeted paraphrases as the additional, artificial references (1H+1P).

In addition, the smaller news commentary data for each language containing respectively 82K, 75K and 74K sentences for French-English, German-English and Spanish-English was also used. As the language model training data, the same data as the Chinese-English experiments were used.

2. **Decoders**. Both the SMT and paraphrase decoders are SCFG-based decoders using hierarchical phrase-based translation models.

3. **Tuning Set**. As the tuning set, the test set from the 2008 workshop on machine translation (Callison-Burch et al., 2008c) containing $2,051$ sentences was used. This is in-domain data that was gathered from the same news

sources as the validation set described below. Note that this set only contains a *single* human authored English reference translation.

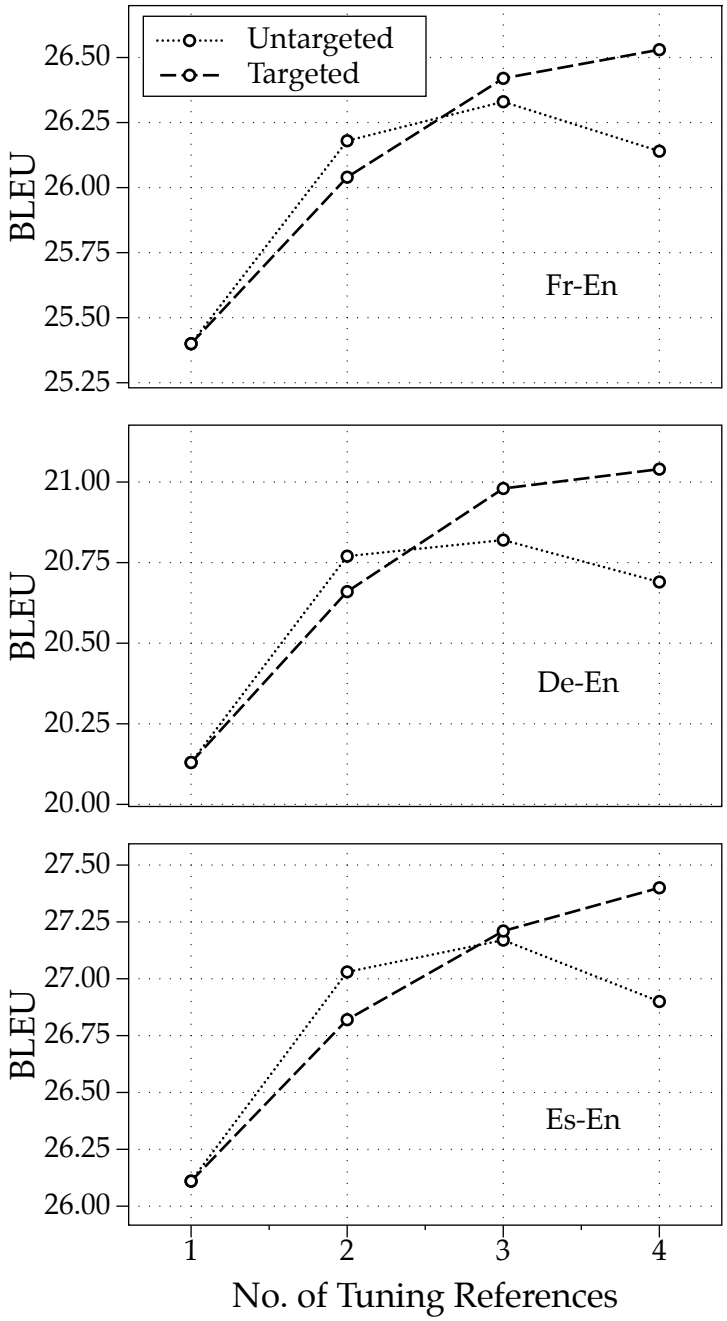
4. **Validation Set**. The validation set is the newstest2009 set which contains 2,525 sentences also with a *single* English reference translation.

Figure 5.12 shows the BLEU and TER results for the validation sets for each of the three language pairs. Recall from Chapter 4 that adding untargeted paraphrases beyond the 1-best contributes more noise than diversity. These results confirm that the benefits from the paraphrased references are able to generalize across multiple language pairs. The targeted paraphraser behaves in a much better fashion than the untargeted paraphraser; it provides performances gain as more paraphrases are added.

Figure 5.13 shows the results for the human judgment experiments for these three European languages as conducted on Amazon Mechanical Turk. These results confirm that the targeted paraphrase are more useful for SMT parameter tuning than using the untargeted paraphrases.

5.5 Afterword: Self-paraphrase Bias and Untargeted Paraphrasing

This section addresses an obvious question that follows from the discussion in the previous sections: given that the self-paraphrase bias is designed to strengthen the semantic link between the paraphrase and the original reference, couldn't it also be used to improve the untargeted paraphraser?



# refs	Untargeted		Targeted	
	BLEU	TER	BLEU	TER
1H+0	25.40	56.14	25.40	56.14
1H+1	**26.18**	55.47	**26.04**	55.59
1H+2	**26.33**	55.26	**26.42**	55.28
1H+3	**26.14**	55.40	**26.53**	55.12

# refs	Untargeted		Targeted	
	BLEU	TER	BLEU	TER
1H+0	20.13	60.80	20.13	60.80
1H+1	**20.77**	60.17	**20.66**	60.31
1H+2	**20.87**	60.06	**20.98**	59.97
1H+3	**20.74**	60.18	**21.04**	59.86

# refs	Untargeted		Targeted	
	BLEU	TER	BLEU	TER
1H+0	26.11	55.32	26.11	55.32
1H+1	**27.03**	54.44	**26.82**	54.64
1H+2	**27.17**	54.31	**27.21**	54.23
1H+3	**26.90**	54.55	**27.40**	54.01

Figure 5.12: Graphs showing the BLEU scores for the set newstest2009 as different types of paraphrased reference translations are added to a single human authored reference translation for the tuning set (newstest2008). Note that the BLEU score for this validation set is measured against one reference translation. From top to bottom: French-English, German-English and Spanish-English. The corresponding TER scores are shown in the accompanying tables.

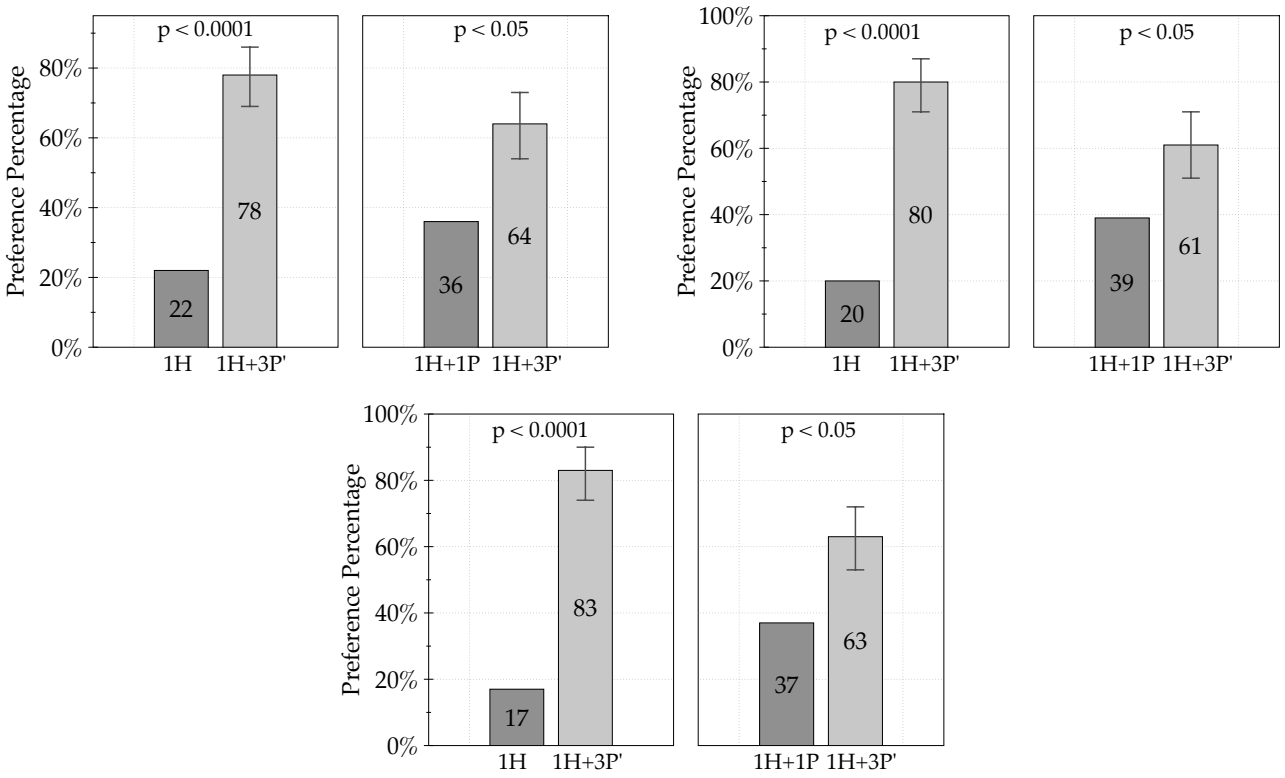


Figure 5.13: Just like Chinese-English translations, human subjects—when asked to judge translations for European languages—have a significant preference for outputs produced by an SMT system augmented with targeted paraphrases (1H+3P') as compared to two baselines: a system that only uses the human reference (1H) and one that is augmented with 1-best untargeted paraphrase (1H+1P). Clockwise from top to bottom: French-English, German-English and Spanish-English.

To answer this question, two self-paraphrasing bias values are chosen (25% and 50%) and applied to the untargeted paraphraser from Chapter 4 to create two new instances of the untargeted paraphraser. The Chinese-English SMT experiments described in Chapter 4 and Section 5.4 are then repeated with the regular unbiased targeted paraphraser and these two new biased untargeted paraphrasers. Figure 5.14 compares the results of these experiments for the MT04+05 validation set.

The curve for the untargeted paraphraser with 25% self-paraphrasing bias indicates that:

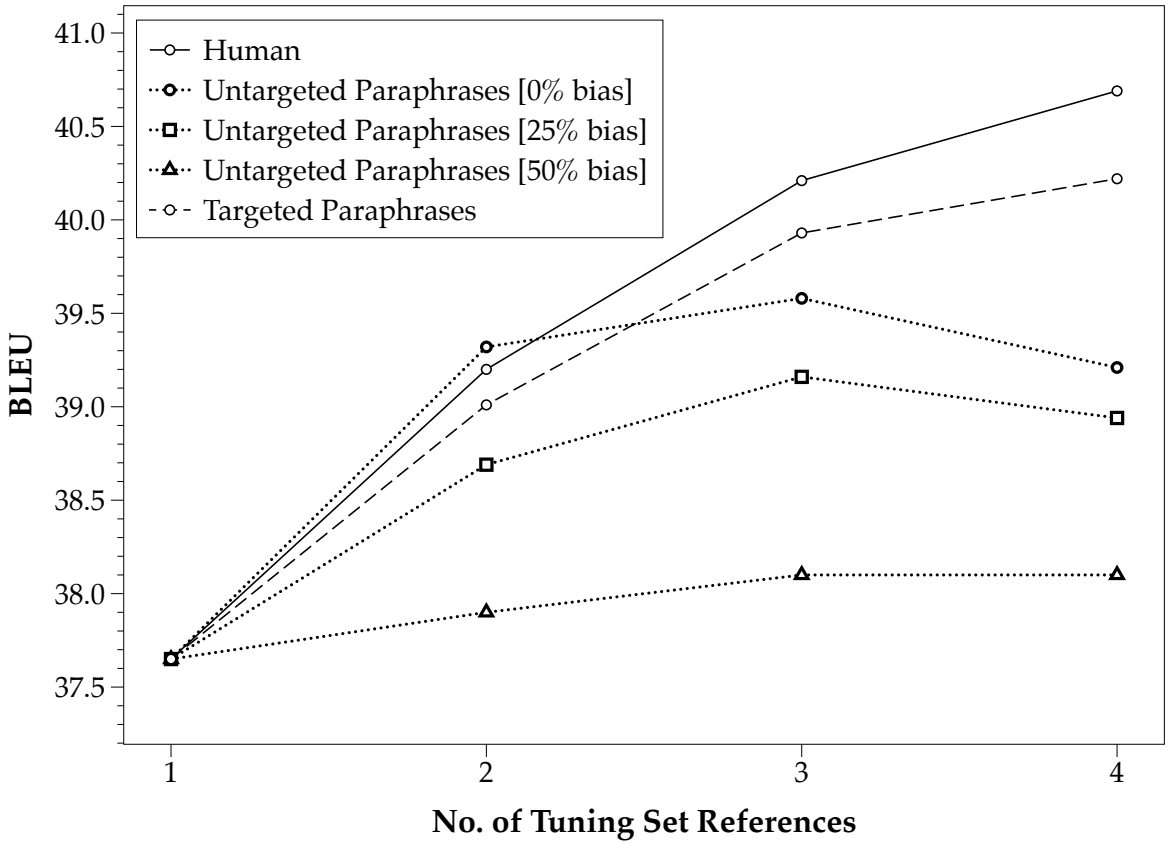


Figure 5.14: A graph showing the BLEU scores for the set NIST MT04+05 as different types of additional reference translations are added to a single human authored reference translation for the tuning set (NIST MT03).

- The gain from using only the 1-best paraphrase, while still significant, is not as large as the gain from using the unbiased untargeted paraphraser.

- The curve seems to follow the same trend as the curve for the unbiased paraphraser, albeit the losses are not as sharp.

On the other hand, the curve for the paraphraser with 50% bias indicates that there are no significant gains when its paraphrases are used for tuning. To see why this would be the case, paraphrases produced by this paraphraser are inspected. Figure 5.15 shows the top 10 paraphrases for an input sentence from the NIST MT03

tuning set. This shows that the paraphrases are almost identical to the original reference. In fact, there is little or no n-gram diversity in the top 3 paraphrases and, therefore, the additional artificial references are of almost no use in tuning.

Reference	Robert Redford calls on independent filmmakers to help protect freedom of speech.
Paraphrases	Robert Redford calls on independent filmmakers to help protect freedom of speech. Robert Redford calls on independent **artists** to help protect freedom of speech. Robert Redford calls on independent **directors** to help protect freedom of speech. Robert Redford calls on independent filmmakers to help **guarantee** freedom of speech. Robert Redford calls on independent filmmakers to **safeguard** freedom of speech. Robert Redford calls on independent filmmakers to protect freedom of speech. Robert Redford calls on independent **producers** to help protect freedom of speech. Robert Redford calls on independent **productions** to help protect freedom of speech. Robert Redford **called** independent filmmakers to help protect freedom of speech. Robert Redford calls on independent filmmakers to **ensure** freedom of speech.

Figure 5.15: The top 10 untargeted paraphrases for a given reference as output by an untargeted paraphraser incorporating 50% self-paraphrasing bias. Words in each paraphrase that differ from the reference are shown in bold.

The above analysis for the paraphraser with 50% bias case also sheds light on the 25% bias case. Recall the assertion from Chapter 4 the main reason the 1-best untargeted paraphrase helps tuning is because it makes a large number of changes and some of those changes just happen to match the translation output. As more and more untargeted paraphrases are added, such fortuitous matches become less and less likely. However, the main problem with the untargeted paraphraser

is not *just* that it makes too many changes. The problem is that it makes too many potentially *useless* changes, i.e., it replaces n-grams in the original reference with other n-grams that have a small a priori likelihood of matching the translation output.[9] Therefore, the solution cannot be to simply reduce the probability of making *any* changes to the original reference, which is what the self-paraphrase bias achieves. Instead, the right solution must be to increase the number of *useful* changes made to the original reference and that is precisely what the targeted paraphraser is designed to do. The best way to understand the purpose of the self-paraphrase bias is to imagine the analogy with the HTER computation process, which is what the targeted paraphraser is designed to emulate. The goal for HTER is to make the smallest number of (semantically equivalent) changes such that the reference is closer to the translation output. Note that if a human were creating the targeted reference (as is done for HTER computation), any explicit self-paraphrase bias would be unnecessary.

The next chapter expands upon some of the points above with a more detailed analysis and some paths for future work. Taking an overarching view of the reference sparsity problem, I argue that reference translations must possess three qualities in order to be useful for parameter tuning. An approach to measuring how the various types of reference translations used in this dissertation—human, untargeted paraphrases of a human reference and targeted paraphrases of a human reference—compare in terms of these qualities is also presented.

[9]Of course, given noisy word alignments, it is possible that the n-gram being replaced has no semantic overlap with its replacement. In that case, the change could be useless and detrimental.

6 Discussion & Future Work

This chapter takes an overarching view of the characteristics of reference translations insofar as they are related to the SMT parameter tuning process; it first uses a simple analogy as an expositional device such that the reader can gain an intuitive understanding of the various solutions discussed in this thesis for the reference sparsity problem. Secondly and more formally, it posits the exact qualities that a reference should possess in order to prove useful for the tuning process. The three types of references encountered in the previous chapters of this thesis—human, untargeted paraphrases of human references and targeted paraphrases of human references—are then compared in terms of these qualities, both theoretically as well as empirically. This analysis proves to be extremely valuable since it is able to explain, e.g., exactly why targeted paraphrases tend to prove more useful for tuning than untargeted paraphrases. Thirdly, a comparison to other MT-related paraphrasing approaches is provided. Finally, the conclusions reached in this thesis are presented along with several avenues of future research.

6.1 A Dartboard Analogy

This section presents an analogy of the n-gram matching process that lies at the heart of the parameter tuning algorithm. This analogy should serve to provide

the reader with an intuitive understanding of the effect of reference sparsity on this n-gram matching process and the fundamental contribution of the various solutions that attempt to address this sparsity.

For this analogy, imagine that matching an n-gram from the translation output against one in the reference is like trying to hit the bulls-eye on a dartboard.[1] The difficulty of hitting the bulls-eye should directly evoke the sparsity encountered by the tuning algorithm with a single reference. One possible solution for the reference sparsity is to use multiple (usually 4) human-authored reference translations and, in the dartboard analogy, this may be considered equivalent to scaling the dartboard four times which leads to a bulls-eye that is four times as large and, hence, four times easier to hit. In terms of the solutions proposed in this thesis, using untargeted paraphrases as artificial references can be considered equivalent to the situation wherein the bulls-eye is still scaled but it is broken into pieces that are scrambled all over the dartboard. This represents the fact that using the untargeted paraphrases still increase the chance of matching an n-gram but not a whole lot due to the inherent noise. Finally, for the case of using the targeted paraphrases, the bulls-eye is still scaled to a larger size and still broken and scrambled (albeit not as much due to the self-paraphrase bias) but now instead of throwing a dart, one can use a dart rifle with a scope (indicating the targeting).

[1]The parameter tuning algorithm almost always involves matching more than a single n-gram but the board only has a single bulls-eye. However, one could simply assume that a new dartboard is generated for every n-gram to be matched in the translation output and the analogy carries through.

6.2 What's in a (Tuning) Reference?

In this section, I argue that a reference (or set of references) must possess three qualities in order to be effective when used with the algorithm that tunes the parameters of an SMT system:

1. **Correctness**. The reference should be correct, i.e., it should possess exactly the same meaning as the corresponding source sentence in the tuning set. This is important since the goal of the tuning is to try to encourage the MT system to produce translations that look like the reference translation. It is reasonable to state that a more correct reference would be more useful for tuning.

2. **Reachability**. It is important that reference translation that the system is being tuned to has phrases that are possible to produce from the source sentence using the MT system. To the extent that part of a reference may not be reachable, that reference only provides a fixed error rate and is not helpful in tuning because the MT system cannot be coaxed to give that answer just by changing the parameters or feature weights. The point is that if the translation output can be *changed* to match the reference during the tuning process, then the reference is considered reachable. Note that it was previously mentioned that the translation output for the tuning set can itself be seen as points in a space of references, i.e., a set of reference translations. Obviously, these references are not correct but are 100% reachable.

3. **Focus**. It is easiest to understand focus with the help of an example. Assume

that a typical translation output in a tuning set has about 10 errors relative to a human reference. With the help of the tuning algorithm, say that 2 out of the 10 errors are fixed. However, it is quite possible that probably half (say 5) of the purported "errors" are actually not errors but paraphrases that didn't happen to match on the surface.[2] Therefore, out of the 2 errors that are fixed during tuning, it is likely that one of them is actually not an error at all. Having focus would mean knowing which errors actually need fixing, so that the tuning process can concentrate on the *real* errors.

6.3 Comparing References

This section compares the three references that have been used for parameter tuning in this thesis in terms of the three qualities that were described in Section 6.1. Given that the process of constructing the two types of artificial references (untargeted and targeted paraphrases) is well understood and described in detail in the earlier chapters of this thesis, a simple theoretical comparison of the references can prove to be sufficiently convincing. However, for the sake of completeness, this section also provides empirical quantitative measurements that provides additional evidence for the theoretical comparison. Note that the goal is not necessarily to measure the intended quality perfectly but in a reasonably correct manner so as to produce a useful comparison of the various types of reference translations used in this thesis.

[2]If HTER were to be computed for this sentence, it will be easily seen which of the 5 errors that are actually errors.

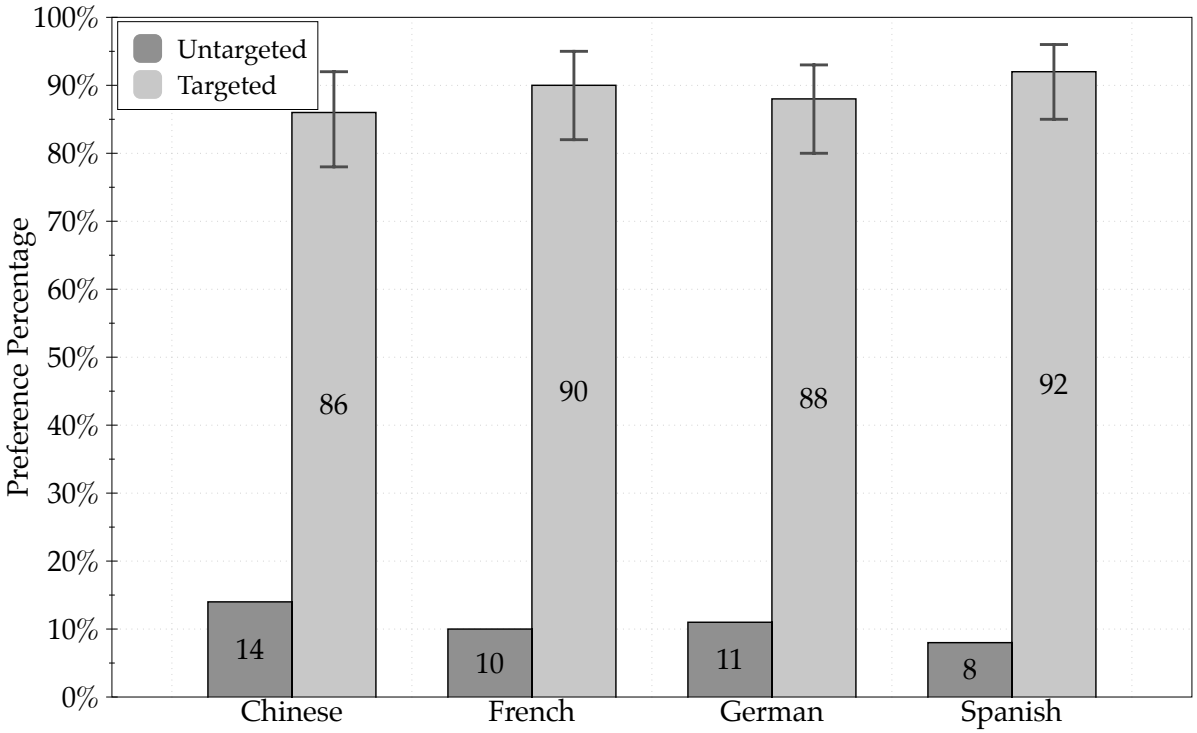


Figure 6.1: Human subjects on Amazon Mechanical Turk were shown the 1-best untargeted and the 1-best targeted paraphrases for 100 randomly chosen human references from the tuning sets of 4 languages. They were then instructed to choose the more correct paraphrase. The subjects preferred the targeted paraphrase compared to the untargeted one almost 9 out of 10 times. This is expected since the targeted paraphrases, by design, retain more of the words from the original reference and, hence, more of the meaning. All are significant at $p < 0.0001$.

1. **Correctness**. It is very likely that human-authored reference translations are fully correct although, when translating between very different languages (such as Chinese and English), this is not always guaranteed. It is definitely more likely that the human references are more correct than their untargeted paraphrases (that have been shown to be only approximately paraphrastic) and even the targeted paraphrases. However, between the two types of paraphrases, the targeted ones are created by a paraphraser that is explicitly designed to retain more of the original meaning (by virtue of the self-paraphrase bias), and therefore they are more likely to be *correct* than the untargeted

ones. Note that in all results shown in Chapters 5, using multiple human references yielded better results on the validation set (although not statistically significantly better in all cases) since *all* human references are usually correct and are able to provide more n-gram diversity than the targeted paraphrases of a single human reference.

In order to confirm whether these assertions are true, a simple experiment was conducted on Amazon Mechanical Turk. 100 human references were chosen at random from the respective Chinese-English, French-English, German-English and Spanish-English tuning sets that have been described in detail in Chapters 4 and 5. 10 HITs were then created, each containing 10 of the 100 human references along with their 1-best untargeted and 1-best targeted paraphrases. The instructions in each HIT told the participating Turkers to pick the paraphrase that they thought was more correct. A third option indicating that there was no difference between the two was also provided. Each sentence from each language was judged three times. The final answer for each case was picked by a simple majority vote.[3] If the final answer for a sentence was the no-difference option, then that sentence was excluded from consideration. Results for this experiment are shown in Figure 6.1. The fact that the Turkers deemed the targeted paraphrase to be more correct almost 9 out of 10 times for each of the 4 languages validates the assertion. Note that the instructions to the Turkers stated that a paraphrase may be more correct i.e., retained

[3]Like Mechanical Turks experiments described in earlier chapters, answers were validated by embedding a control question in each HIT for which the correct answer was known before hand.

more of the semantic concepts of the original reference, without being fluent.

2. **Reachability**. Although human-authored references maximize correctness, they are very often not reachable, i.e., it is not possible for the MT system to actually produce the reference n-grams in its translations. One possible way of increasing the reachability is to simply use several distinct human references which can be considered equivalent to manually paraphrasing a single reference.[4] However, there are no guarantees since it is still possible for the other references to contain n-grams that cannot be produced by the MT system.

   Automatically paraphrasing a single human reference is another way to increase reachability, assuming that the n-grams used to create the paraphrases are reachable.[5] With the sentential paraphraser described in this thesis, this is certainly the case; *all* the English phrases that compose the paraphrase rules are learned, via pivoting, from the same bitext that is used to train the MT system. Obviously, this means that the untargeted paraphrases—produced by a paraphraser designed to replace as many n-grams in the original reference with reachable, partially semantically equivalent counterparts—are more reachable than the targeted paraphrases that only make a small number of focused and targeted changes. Note that the reachability of both types of

[4]Note that, in practice, each reference is created by asking humans to translate the source sentence rather than by paraphrasing an already existing reference. However, the two processes can be considered theoretically equivalent since each one is presumably fully meaning preserving in nature.

[5] Using another sentential paraphraser from Chapter 2 that is not guaranteed to paraphrase with more reachable n-grams is the same as adding more human reference translations albeit the automatic paraphrases are less likely to be fully correct.

paraphrases increases as 2-best and the 3-best paraphrases are included in the reference set. However, in the untargeted case, the noise also increases significantly with k-best paraphrases and overwhelms any advantages of increased reachability.

Reachability can be measured by examining how much the tuning process is able to move the translation output towards a reference. So, for example, if the MT system is tuned against a human reference, the new (tuned) output will presumably be closer to the human reference used which can be measured by computing the BLEU score of the tuned translation output against the human reference. Similarly, if the system is tuned using 1 human reference and its 1-best untargeted paraphrase, then even though the original MT output does not match the untargeted paraphrased reference very well, perhaps after tuning, the MT output matches this untargeted paraphrase much better—possibly almost as well as it would match a second human reference. Again, this can be measured computing the BLEU score of the tuned output against the untargeted reference alone. In order to compute these numbers, three tuning experiments were performed; the first experiment used two human references (say A and B) for each source sentence, the second used A and its 1-best untargeted paraphrases, and the last experiment used A and the its 1-best targeted paraphrases (in the same manner as described in Chapter 5). For each of these experiments, the BLEU score before and after tuning was measured against the both references. Figure 6.2 shows these measurements. Column C

shows that the gain in the BLEU score for the untargeted paraphrase reference against the MT output indeed is the largest compared to both the second human reference and targeted paraphrases, indicating higher reachability as predicted above.

3. **Focus**. The targeted paraphraser creates references that are in the direction of HTER references since it is designed to crudely emulate an HTER annotator. It identifies words and phrases that were already (likely to be) correct, because they could be produced paraphrasing a correct reference. Hence, they increase the focus by allowing the tuning process to ignore these "errors" and focus on other ones. Human references are not focused since they are created without any information about the translation output. Neither are untargeted paraphrases of human references.

   Focus can be measured simply by looking at how many of the words or phrases in the original translation output appear to be different from the set of references. This is given by the BLEU score of the original MT output against the references. Column A shows that the targeted paraphrases have the highest value for this BLEU score, which means they are more focused on the remaining differences than either the human or the untargeted references.

6.4 Comparison to other MT-related Techniques

This section provides a comparison of the thesis research to two other research efforts. These could not be presented in the earlier Related Work section (Chapter

Type of second reference	For second reference			Gain on first reference	Total Gain
	Pre-tune (A)	Post-tune (B)	Gain (C)		
Human	20.4	22.3	2.1	2.3	4.4
Untargeted	19.6	22.5	2.9	2.0	4.9
Targeted	21.7	24.0	1.8	2.1	3.9

Figure 6.2: Various BLEU measurements for three different Chinese-English tuning experiments that all use the same set of human references as the first reference and one of three different types as the second reference: (a) a different set of human references (b) untargeted paraphrases of the first set of references and, (c) targeted paraphrases of the first set of references.

2) because the inner workings of the parameter tuning algorithm and the details of the targeting implementation needed to be described first, before the comparison would be accessible to the reader.

First, the idea of using the translation output to influence the reference has also been explored by Kauchak and Barzilay (2006). However, their work differs significantly from the ideas presented in this dissertation. One difference is that they create *one* paraphrase for the sole purpose of obtaining a more informative automatic evaluation score for the final translation output of an already tuned SMT system. This thesis creates *multiple* paraphrases, for the purpose of addressing reference sparsity in the SMT parameter tuning process. Another difference is that, although used for *evaluating* SMT output, their paraphrasing technique relies on machinery entirely unrelated to the translation. They paraphrase words in the reference by replacing them with synonyms from WordNet that might occur in the translation output. In contrast, this thesis creates a fully data-driven sentential paraphraser entirely from SMT machinery. Finally, the work of Kauchak and Barzilay (2006) shows that, by using this alternative reference *in place* of the original human ref-

erence for computing BLEU, better correlation with human judgments is obtained. The human judgments they used are manually assigned ratings on a 1 to 5 scale reflecting only the adequacy of the translation output but disregarding its fluency. This thesis shows that by using the targeted sentential paraphraser in addition to the human reference, parameter tuning can be improved significantly as measured both in terms of automatic metrics as well as human preference judgments. Past experience with annotations of human judgments of SMT output has shown that human raters have difficulty in consistently assigning absolute scores—such as those for adequacy—to MT system output, due to the large number of possibly correct translations. Callison-Burch et al. (2008b) showed that preference judgments are considerably easier to make and, therefore, more reliable.

More recently, researchers have demonstrated the feasibility of human-in-the-loop tuning of SMT parameters (Zaidan and Callison-Burch, 2009). This idea is related to the work presented in this thesis since it is concerned with the same stage of the SMT pipeline: parameter tuning. However, the motivation for their work is derived from a different, extrinsic problem associated with the SMT parameter tuning process rather than the intrinsic problem of reference sparsity. As described in this thesis, parameters of most SMT systems are tuned using BLEU. However, when evaluating the translations produced by these systems, a recent trend is to use a metric with a human component like HTER. Therefore, the authors propose a new metric, Ratio of Yes nodes in the Parse Tree (RYPT), which takes human judgments into account thereby lengthening the tuning process and requiring significantly more human effort. However, the metric only requires human input to build a database

that can be reused over and over again, hence eliminating the need for human input at tuning time. The authors show that their metric is a better predictor of human judgment of translation quality as compared to BLEU. Amazon Mechanical Turk is used to create the above database in a cost-effective manner. Although unexplored by the authors, this work could also have implications for the reference sparsity problem in that since the tuning algorithm is now driven by a human in the loop providing judgments of the translation quality, there may be no need for reference translations at all.[6]

6.5 Future Work

There are several possible avenues in which the work presented in this dissertation can be improved:

1. **Exploiting Monolinguality**. Despite the intimate relationship between the sentential paraphraser and an SMT system, one aspect of the sentential paraphraser radically differentiates it from an MT system: the fact that the source and the target languages are the same. The monolingual nature of the paraphrase generation task can be still further exploited than it has been in this thesis. This can be achieved by developing features and incorporating additional knowledge—much more easily than for a bilingual MT system—that can substantially improve the performance of the paraphraser and make it even

[6]The authors collect two types of human judgments when building the database: (a) those elicited by showing both the source sentence, the translation output and a single reference translation of the source and (b) those elicited by showing only the source sentence and the translation output. The results show that even for judgments where no reference was shown, their metric remains a better predictor of quality compared to BLEU.

more useful in scenarios where it may not yet perform up to its potential. For example, features could be developed that are more informative than the simple probabilistic features that are currently used to drive the paraphraser. One potentially useful feature could be derived from English synonyms that were automatically obtained from the English side of the bitext via distributional similarity techniques described in Chapter 2. Other English resources such as WordNet, CatVar (Habash and Dorr, 2003) and LCS lexicons (Jackendoff, 1987, 1990) can also prove useful.

2. **Improving Paraphraser Architecture**. There are several possible ways in which the general sentential paraphraser architecture proposed in Chapter 3 could be improved.

   (a) The noise inherent in pivoted monolingual translation rules could be reduced by using multiple pivot languages to generate several sets of paraphrasing rules and then averaging over them. This approach was indeed explored by Bannard and Callison-Burch (2005) and is a definite item of future work. Another possibility is to retain only those rules that lie at the intersection of these sets of paraphrases obtained from multiple languages. An obvious disadvantage of this approach is that the increased precision comes at the cost of decreased coverage. Finally, one could also keep the rules that are not in the intersection but weight them lower than the other rules that are more likely to be correct.

   (b) Another way to reduce the pivoting noise is to perform the pivoting in

fully lexicalized space, i.e., starting with the initial phrases as described previously in Section 2.4.5 and then converting the fully lexicalized monolingual rules into the corresponding hierarchical versions.

(c) Another possibility is to define a more sophisticated targeting feature that counts the number of undesirable n-grams instead of words so that the targeting behavior can be made even more effective than it already is.

3. **Doing Away With Separate Tuning Sets.** The results presented in this thesis point in a more ambitious direction: doing away *entirely* with any human translations beyond those already a part of the training bitext already expected by statistical systems. If the quality of the translations in the training set are good enough—or if a high quality subset can be identified—then the paraphrasing techniques presented here may suffice to obtain multiple reference translations with the qualities needed to tune statistical MT systems effectively.

4. **Tuning The Paraphraser**. Another area that merits consideration is the tuning metric used for the paraphrasers. Currently the feature weights for the paraphraser features are tuned as described in Chapter 3 similarly to how weights are tuned for an SMT system, i.e., by iteratively "translating" a set of source paraphrases, comparing the answers to a set of reference paraphrases according to the BLEU metric and updating the feature weights to maximize the BLEU value in the next iteration. While this is not unreasonable, it is

not optimal or even close to optimal: in addition to striving for semantic equivalence, an automatic paraphraser should also aim for lexical diversity especially if said diversity is required in a downstream application. However, the BLEU metric is designed to reward larger n-gram overlap with reference translations. Therefore, using BLEU as the metric for the tuning process might actually lead to paraphrases with lower lexical diversity. Metrics recently proposed for the task of detecting paraphrases and entailment (Dolan et al., 2004; João et al., 2007a,b) might be better suited to this task.

5. **Finding the Ideal Tuning Reference.** Finally, there is much more work that can be done to determine what would be the *ideal* tuning reference and whether it is something that can actually be obtained automatically using the techniques described in this thesis. How would this ideal reference be defined? Human-authored references are correct but not fully reachable and certainly not at all focused. References created by untargeted paraphrasing are fully reachable but not correct and not focused. Targeted paraphrasing creates references that are very focused but less reachable and not always correct due to the noisy interaction between pivoting and targeting. Even HTER references cannot be considered ideal since they can simply be thought of as a special case of the targeted paraphrases where the targeting is carried out by a human; the low reachability is still retained.

   So, what would constitute the ideal tuning reference? In theory, *all possible* paraphrases of the reference must be considered along with all possible (likely)

translations of the source sentence. The intersection of these two sets that has the highest translation model score and the highest paraphrase model score—indicating that it is possible for the decoder to get this answer with a high score and that it is not too far away from the reference, semantically speaking—yields our ideal answer. Note that the implementation of targeting in this thesis does try to approximate this: first the n-best MT outputs (sorted by translation model score) are found, and then for each one, a paraphrase of the reference that has a high paraphrase model score is created. One shortcoming of this approach is that using n-best lists is not a good enough source of diverse alternatives, an idea that has been explored in the literature (Langkilde, 2000; Mi et al., 2008). One could use packed representations, such as full lattices, to represent each of the sets, which would complicate the tuning process but make it easier to find the ideal tuning reference.

To the extent that a targeted paraphrases does well at matching the MT output, it represents a good tuning reference. However, it might be more useful to find and use those n-grams that are reachable by the system, rather than just those that have already been reached in existing translation output.

6.6 Conclusions

Chapter 1 posed the questions that are central to establishing the the symbiotic relationship that is at the heart of the thesis. These questions have now been fully answered:

1. Is it possible to extend the existing work on paraphrase generation to the sentential level by, in fact, casting this problem as one of English-to-English translation? Chapter 3 showed that it is indeed possible to build such a translation system by building on top of the phrasal paraphrasing work done by Bannard and Callison-Burch (2005) which works well and is able to generate n-best paraphrases for any given input sentence. It also showed that while the paraphrases are not perfectly semantically equivalent, they might be quite suitable for use in another downstream application which could take advantage of the partial semantic equivalence, such as statistical machine translation.

2. How should this English-to-English translation model be constructed and defined in order to maximize meaning preservation? Chapter 3 showed that the translation model used to build the paraphraser is a novel, extensible and well-defined log-linear model that can incorporate several different kinds of features. One of the most important advantages of using such a model is that since the same model is also used in bilingual translation, almost all of the machinery to train the model also carries over except for a few small parts that were adapted for use in a monolingual setting.

3. Given the expense of asking humans to create these translations, most new datasets only contain a single reference leading to reference sparsity and, ultimately lower quality translation. Is it possible to create additional, artificial references by paraphrasing the single reference using the paraphraser built in (1) above and improve the translation quality? Chapter 4 showed that using

the paraphrases of a single human-authored reference translation as additional references for parameter tuning provides significant gains in translation quality over the baseline case of using *just* that reference. These significant gains are demonstrated not only in terms of automatic MT evaluation metrics but also in terms of human preference judgments. However, one drawback of using the paraphraser is that, since it is basically a translation system, it is designed to "translate" or paraphrase all the words in the original reference. Therefore, the primary reason that it helps address reference sparsity is due to the sheer volume of changes in that some of them turn out to be useful, i.e., match the translation output. However, as more paraphrases from the n-best paraphrase list are added, the fraction of useful changes decreases and the behavior diverges from that obtained by using multiple human reference translations. Chapter 5 takes a step back and creates a new SMT-specific instantiation of the paraphraser that is a priori more likely to make only useful changes. Results show that using this version of the paraphraser to create additional, artificial references as described in Chapter 5 leads to significant gains in translation quality not only over the human-only baseline but also over the change-everything paraphraser approach presented in Chapter 4.

4. What characteristics should an artificial reference have in order for the translation system to learn as effectively as it might do with a human-authored reference translation? In addition to presenting empirical gains in Chapters 4 and 5 as indicators of the utility of the paraphrased references, this chapter

also discusses three qualities that a reference translation should have in order to be useful for tuning SMT parameters. A comparison of the various types of references used in this dissertation is presented in terms of these qualities, both intuitively and empirically. This comparison illuminates how the two different instantiations of the paraphraser implemented in this thesis operate at a fundamental level.

Through the research conducted in this thesis, I have made several significant research contributions: (a) a new general sentential paraphraser architecture that is built entirely using bilingual SMT machinery and by extending previous research on phrasal paraphrase generation (b) the first automatic solution that directly addresses the reference sparsity problem in SMT by using the above sentential paraphraser to create artificial references. The solution is successful in that these additional references are shown to lead to statistically significant gains in translation quality as measured both by automatic MT evaluation metrics as well as human preference judgments (c) the first detailed characterization of a reference translation in terms of three qualities of reference translations: correctness, reachability, and focus, along with a theoretical and empirical analysis and, (d) a comprehensive overview of paraphrasing approaches that brings together fragmented research on data-driven paraphrase generation and draws broad philosophical connections among related efforts.

A Translation Examples

SRC	N'empêche qu'il existe suffisamment de raisons de se procurer un lecteur indépendant.
REF	In spite of this, there are many reasons to get a separate MP3 player.
BSN	Despite that it sufficiently exists of reason for providing an independent player.
UNT*	But there are plenty of reasons to get an independent player.
SRC	Alors que les dettes au premier chef ont à peine bougé, les cours des prêts de second plan fléchissent.
REF	Whereas senior debts are hardly moving, junior bonds are dropping more significantly.
BSN	While the debts of the first leader have not moved, the ready courses the of minor plan bend.
UNT*	While the debts have barely budged, loans from second warehouses been dropping.
SRC	New York dispose de centaines d'avenues, de boulevards, de rues et d'autres routes parallèles, certaines célèbres, d'autres uniquement fonctionnelles.
REF	New York City has thousands of avenues, boulevards, streets and other byways, some famous, others merely utilitarian.
BSN	N.Y arranges of hundreds of avenue, of boulevards, of streets and others parallel roads, famous, of the other simply functional.
UNT*	New York has hundreds of avenues, boulevards, streets and other parallel roads, some well-known, others merely functional.
SRC	Il doit obligatoirement informer les autorités s'il désire aller à l'étranger pour plus de trois jours.
REF	He must give the authorities notice if he wishes to travel abroad for more than three days.
BSN*	He must compulsorily inform the authorities if he wants to go abroad more than three days.
UNT	It must inform the authorities if it wants to go abroad to more from three days.
SRC	De ces racines européennes naîtra un des visages glorieux du rêve américain.
REF	From these European roots were born the glorious faces of the American dream.
BSN*	Of these European roots were born a glorious faces of the American Dream.
UNT	Of these European roots into a glorious visages the American dream.

Figure A.1: Examples of translations produced for randomly chosen **French** (SRC) sentences. UNT is the translation produced by the SMT system tuned using **untargeted** paraphrases and BSN is the one produced by the baseline translation system tuned using the single human-authored reference. The references (REF) are also shown for comparison. `[*]` denotes the translation preferred by Turkers.

SRC	Obama est vraiment culotté & il croit qu'il gagnera la plus haute fonction d'une nation qui compte en tout, deux gouverneurs et cinq sénateurs noirs !"
REF	Obama is being cheeky & he thinks he can win the highest position in a country which has had in total two governors and five senators that were black!"
BSN	Obama is really rich & it believes that it will build the leadership of a nation which in all, two governors and five senators black!
TRG*	Obama is really cheeky & he believes he would win the highest office of a nation that has a total of two governors and five senators are black!
SRC	Malgré ce pitoyable échec électoral, les têtes ne sont pas encore tombées, et le débat sur les questions personnelles aux sommets du CSU bavarois a été ajourné.
REF	Despite ignominious election failure heads have not rolled yet, personal issues are adjourned temporarily at the Bavarian CSU summits.
BSN	Despite this pitiful election, the warheads are not yet fallen, and the debate on the personal questions heights of Bavarian CSU was delayed.
TRG*	Despite this pitiful failure, electoral heads have not yet rolled, and debate on issues of personal summits at Bavarian CSU has been postponed.
SRC	L'il à facettes de l'abeille joue un rôle primordial, puisqu'il est capable de voir dans toutes les directions, dans un angle de 300 degrés.
REF	The compound structure of bees' eye plays a significant role, which makes the bees able to see in all directions, at an angle of 300 degrees.
BSN	The it to facets of the bee plays a role, because it is able to see in many directions, in a corner of 300 degrees.
TRG*	The faceted eye of the bee plays an important role, since it can see in all directions in angle of 300 degrees.
SRC	Celui qui croît en Dieu ressent-il moins la douleur ?
REF	Does it hurt less if you believe in God?
BSN	Anyone believes in God has less pain?
TRG*	Whoever believes in God, does he feel less pain?
SRC	Aujourd'hui, a débuté, à Paris, le Sommet Européen pour l'Égalité des chances, dont l'hôte est la France, comme représentant de la présidence actuelle de l'Union Européenne.
REF	The European Equality Summit has been opened today in Paris, which is organized by France this year, representing the Presidency-in-office of the European Union.
BSN*	Today has started, in Paris, the European summit for Equal Opportunities, whose host is France, as a representative of the current presidency of the European Union.
TRG	Today started in Paris for the European Equal Opportunities, with host as France, representing the current presidency of the European Union.

Figure A.2: Examples of translations produced for randomly chosen **French** (SRC) sentences. TRG is the translation produced by the SMT system tuned using **targeted** paraphrases and BSN is the one produced by the baseline translation system tuned using the single human-authored reference. The references (REF) are also shown for comparison. `[*]` denotes the translation preferred by Turkers.

SRC	Ich wohne am Rand Tokios in Richtung Kawasaki und fahre 40 Minuten mit dem Zug zur Tachibana Highschool.
REF	I live at the edge of Tokyo, towards Kawasaki, and the train ride to Tachibana High School is 40 minutes.
BSN	I live in Tokyo direction Kawasaki and take 40 minutes with the train to Tachibana Highschool.
UNT*	I live on the Tokyo edge towards Kawasaki and move 40 minutes by train to Tachibana Highschool.
SRC	Eine Ratte oder eine Schabe flieht bei Gefahr heißt das, dass sie auch Furcht empfindet?
REF	When in danger, a rat or roach will run away. Does it mean they experience fear, too?
BSN	A rat or a Schabe flees by danger that means that they also feel fears?
UNT*	A rat or a cockroach is fleeing when in danger, that means that they felt fear?
SRC	Auch die neue, durch die Lega eingebrachte Gesetzgebung gegen Einwanderung und zur schnelleren Ausweisung der illegalen Einwanderer wird als Zeichen des Klimawandels in Italien angeführt.
REF	Even the new legislation from the League that tightens the brakes on immigration and facilitates the expulsion of illegal immigrants is advanced as a paradigm of the change in climate in Italy.
BSN	Also the new brought in through the Lega legislation against immigration and to fast expulsion of the illegal immigrants as signs of the transformation in Italy.
UNT*	The new legislation put forward against immigration to speed up the deportation of illegal immigrants is seen as sign of climate change in Italy.
SRC	Die Liste der von den Fruchtsäften gehemmten Wirkstoffe ist lang.
REF	The list of active principles inhibited by grapefruit and other fruit juices is very long.
BSN*	The list of the hinder from the fruit juices agents is long.
UNT	The list of the fruit juices bad agents.
SRC	Zum ersten Mal in der Geschichte des Palazzo Chigi gibt es drehbare Türen für die Ein- und Ausgänge der Beamten
REF	It has never happened in all the history of the Chigi Palace that there were revolving doors at the employee entrance and exit.
BSN*	To first time in the history of the Palazzo Chigi there are doors for the entering and exiting of employees.
UNT	For the first time in the history of the Palazzo Chigi are there drehbare doors for the import and exits of officials.

Figure A.3: Examples of translations produced for randomly chosen **German** (SRC) sentences. UNT is the translation produced by the SMT system tuned using **untargeted** paraphrases and BSN is the one produced by the baseline translation system tuned using the single human-authored reference. The references (REF) are also shown for comparison. `[*]` denotes the translation preferred by Turkers.

SRC	Nach dem steilen Abfall am Morgen konnte die Prager Börse die Verluste korrigieren.
REF	After a sharp drop in the morning, the Prague Stock Market corrected its losses.
BSN	After the steep waste at tomorrow the Prague stock exchange cannot correct the losses.
TRG*	After the steep waste in the morning, the Prague Stock Exchange losses corrected.
SRC	Trotz allem gibt es genügend Gründe dafür, warum man sich einen eigenständigen Player zulegen sollte.
REF	In spite of this, there are many reasons to get a separate MP3 player.
BSN	Despite that it sufficiently exists of reason for providing an independent player.
TRG*	In spite of everything, there are plenty of reasons why an MP3 player can be independent.
SRC	Hören Sie sich die vier Versionen der neuen tschechischen Nationalhymne an.
REF	Listen to the four renderings of the new version of the Czech national anthem.
BSN	The four versions of the new Czech national anthem listen to you.
TRG*	Listen to the four versions of the new Czech national anthem.
SRC	Doch dieses Mal ist Elena als seine Freundin mit dabei, und so sei alles anders, sagt er.
REF	But this time, he brought along his girlfriend Elena, and that changes everything, he says.
BSN	Yes this time Elena as his friend with in addition to, and so differently all, he says.
TRG*	But this time, Elena as his girlfriend with it, and so was everything else changed, he says.
SRC	Kongress macht Zugeständnis: US-Regierung darf 700 Milliarden Dollar in die Banken pumpen
REF	Congress yields: US government can pump 700 billion dollars into banks
BSN*	Congress makes concession: American government can pump 700 billion dollars in the bank
TRG	Congress makes concession: United States government allowed 700 billion U.S. dollars in the bank pumps

Figure A.4: Examples of translations produced for randomly chosen **German** (SRC) sentences. TRG is the translation produced by the SMT system tuned using **targeted** paraphrases and BSN is the one produced by the baseline translation system tuned using the single human-authored reference. The references (REF) are also shown for comparison. `[*]` denotes the translation preferred by Turkers.

Bibliography

Necip Fazil Ayan and Bonnie Dorr. Going Beyond AER: An Extensive Analysis of Word Alignments and Their Impact on MT. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 9–16, 2006.

Srinivas Bangalore and Owen Rambow. Corpus-based Lexical Choice in Natural Language Generation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 464–471, 2000.

Colin Bannard and Chris Callison-Burch. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 597–604, 2005.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor, editors. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*. 2007.

Yehoshua Bar-Hillel. *Aspects of Language: Essays and Lectures on Philosophy of Language, Linguistic Philosophy and Methodology of Linguistics*. The Magnes Press, Jerusalem, 1970.

Regina Barzilay and Lillian Lee. Bootstrapping Lexical Choice via Multiple-Sequence Alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

Regina Barzilay and Lillian Lee. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 16–23, 2003.

Regina Barzilay and Kathleen McKeown. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 50–57, 2001.

Regina Barzilay and Kathleen R. McKeown. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*, 31(3):297–328, 2005.

Samuel Bayer, John Burger, Lisa Ferro, John Henderson, and Alexander Yeh. MITREs Submissions to the EU Pascal RTE Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, pages 41–44, 2005.

Doug Beeferman and Adam Berger. Agglomerative Clustering of a Search Engine Query Log. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data mining (KDD)*, 2000.

Anja Belz. That's Nice...What Can you Do With It? *Computational Linguistics*, 35(1):111–118, 2009.

Rahul Bhagat and Deepak Ravichandran. Large Scale Acquisition of Paraphrases for Learning Surface Patterns. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2008.

Wauter Bosma and Chris Callison-Burch. Paraphrase Substitution for Recognizing Textual Entailment. *Evaluation of Multilingual and Multi-modal Information Retrieval (LNCS-4730)*, pages 502–509, 2007.

Chris Brockett and William B. Dolan. Support Vector Machines for Paraphrase Identification and Corpus Construction. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.

Andrei Broder. On the Resemblance and Containment of Documents. In *Proceedings of the Compression and Complexity of Sequences*, 1997.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

Sabine Buchholz and Erwin Marsi. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, 2006.

Olivia Buzek, Philip Resnik, and Ben Bederson. Error Driven Paraphrase Annotation using Mechanical Turk. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 217–221, 2010.

Chris Callison-Burch. Syntactic Constraints on Paraphrases Extracted from Parallel Corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

Chris Callison-Burch. *Paraphrasing and Translation.* PhD thesis, School of Informatics, University of Edinburgh, 2007.

Chris Callison-Burch, David Talbot, and Miles Osborne. Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 176–183, 2004.

Chris Callison-Burch, Trevor Cohn, and Mirella Lapata. Annotation Guidelines for Paraphrase Alignment. Technical report, University of Edinburgh, 2006a. `http://www.dcs.shef.ac.uk/~tcohn/paraphrase_guidelines.pdf`.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. Improved Statistical Machine Translation Using Paraphrases. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2006b.

Chris Callison-Burch, Trevor Cohn, and Mirella Lapata. ParaMetric: An Automatic Evaluation Metric for Paraphrasing. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2008a.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, 2008b.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Josh Schroeder, and Cameron Shaw Fordyce, editors. *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio, June 2008c.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder, editors. *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Athens, Greece, March 2009.

Moses Charikar. Similarity Estimation Techniques from Rounding Algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, 2002.

Stanley F. Chen and Joshua Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. Technical report, Computer Science Group, Harvard University, 1998.

David Chiang. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228, 2007.

David Chiang, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. The Hiero Machine Translation System: Extensions, Evaluation, and Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2005.

J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20:3746, 1960.

Trevor Cohn and Mirella Lapata. Large Margin Synchronous Generation and its Application to Sentence Compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 73–82, 2007.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. Constructing Corpora for the Development and Evaluation of Paraphrase Systems. *Computational Linguistics*, 34, 2008.

Courtney Corley and Rada Mihalcea. Measuring the Semantic Similarity of Texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 2005.

Carolyn J. Crouch and Bokyung Yang. Experiments in Automatic Statistical Thesaurus Construction. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 77–88, 1992.

Ido Dagan. Invited Talk: It's Time for a Semantic Engine! In *Proceedings of the NSF Symposium on Semantic Knowledge Discovery, Organization and Use*, New York, 2008.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. *Machine Learning Challenges (LNCS-3944)*, pages 177–190, 2006.

Dipanjan Das and Noah A. Smith. Paraphrase Identification as Probabilistic Quasi-Synchronous Recognition. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 468–476, 2009.

Louise Deléger and Pierre Zweigenbaum. Extracting Lay Paraphrases of Specialized Expressions from Monolingual Comparable Medical Corpora. In *Proceedings of the ACL workshop on Building and Using Comparable Corpora*, 2009.

Michael Denkowski and Alon Lavie. Extending the METEOR Machine Translation Metric to the Phrase Level. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.

Michael Denkowski, Hassan Al-Haj, and Alon Lavie. Turker-Assisted Paraphrasing for English-Arabic Machine Translation. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 66–70, 2010.

Bill Dolan and Ido Dagan, editors. *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*. June 2005.

William Dolan, Chris Quirk, and Chris Brockett. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2004.

William B. Dolan and Chris Brockett. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.

Mark Dras. A Meta-level Grammar: Redefining Synchronous TAG for Translation and Paraphrase. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 80–88, 1999.

Mark Dras and Kazuhide Yamamoto, editors. *Proceedings of the Third International Workshop on Paraphrasing*. 2005.

Florence Duclaye, François Yvon, and Olivier Collin. Learning Paraphrases to Improve a Question-Answering System. In *Proceedings of the EACL Workshop on Natural Language Processing for Question-Answering*, pages 35–41, 2003.

Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

Philip Edmonds and Graeme Hirst. Near-synonymy and Lexical Choice. *Computational Linguistics*, 28(2):105–144, 2002.

Noemie Elhadad and Komal Sutaria. Mining a Lexicon of Technical Terms and Lay Equivalents. In *Proceedings of the ACL BioNLP Workshop*, pages 49–56, 2007.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

Alexander Fraser and Daniel Marcu. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*, 33(3), 2007.

Atsushi Fujita and Kentaro Inui. A Class-oriented Approach to Building a Paraphrase Corpus. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.

Atsushi Fujita and Satoshi Sato. Computing Paraphrasability of Syntactic Variants Using Web Snippets. In *Proceedings of the International Joint Conference of Natural Language Processing (IJCNLP)*, pages 537–544, 2008a.

Atsushi Fujita and Satoshi Sato. A Probabilistic Model for Measuring Grammaticality and Similarity of Automatically Generated Paraphrases of Predicate Phrases. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 225–232, 2008b.

Atsushi Fujita, Kentaro Furihata, Kentaro Inui, Yuji Matsumoto, and Koichi Takeuchi. Paraphrasing of Japanese light-verb constructions based on Lexical Conceptual Structure. In *Proceedings of the ACL Workshop on Multiword Expressions: Integrating Processing*, pages 9–16, 2004.

Atsushi Fujita, Shuhei Kato, Naoki Kato, and Satoshi Sato. A Compositional Approach toward Dynamic Phrasal Thesaurus. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 151–158, 2007.

William A. Gale and Kenneth W. Church. A Program for Aligning Sentences in Bilingual Corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 177–184, 1991.

Claire Gardent and Eric Kow. Generating and Selecting Grammatical Paraphrases. In *Proceedings of the European Workshop on Natural Language Generation (ENLG)*, 2005.

Claire Gardent, Marilisa Amoia, and Evelyne Jacquey. Paraphrastic Grammars. In *Proceedings of the Second Workshop on Text Meaning and Interpretation*, 2004.

Konstantina Garoufi. Towards a Better Understanding of Applied Textual Entailment: Annotation and Evaluation of the RTE-2 Dataset. Master's thesis, Language Science and Technology, Saarland University, 2007.

Caroline Gasperin, P. Gamallo, A. Agustini, G. Lopes, and Vera de Lima. Using Syntactic Contexts for Measuring Word Similarity. In *Proceedings of the Workshop on Knowledge Acquisition & Categorization, ESSLLI*, 2001.

Danilo Giampiccolo, Hoa Dang, Ido Dagan, Bill Dolan, and Bernardo Magnini, editors. *Proceedings of the Text Analysis Conference (TAC): Recognizing Textual Entailment Track*. 2008.

Oren Glickman and Ido Dagan. Identifying Lexical Paraphrases From a Single Corpus: A Case Study for Verbs. In *Proceedings of the Conference on Recent Advantages in Natural Language Processing (RANLP)*, 2003.

G. Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Press, Boston, MA, 1994.

D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computational Science and Computational Biology*. Cambridge University Press, 1997.

Nizar Habash and Bonnie J. Dorr. A Categorial Variation Database for English. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 17–23, 2003.

Catalina Hallett and Donia Scott. Structural Variation in Generated Health Reports. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.

Zellig Harris. Distributional Structure. *Word*, 10(2):3.146–162, 1954.

Graeme Hirst. Near-synonymy and the Structure of Lexical Knowledge. In *Working notes of the AAAI Spring Symposium on Representation and Acquisition of Lexical Knowledge*, 1995.

Graeme Hirst. Paraphrasing Paraphrased, 2003. Invited talk at ACL International Workshop on Paraphrasing.

Eduard H. Hovy. *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1988.

Shudong Huang, David Graff, and George Doddington. Multiple-Translation Chinese Corpus. Linguistic Data Consortium, 2002.

Ali Ibrahim, Boris Katz, and Jimmy Lin. Extracting Structural Paraphrases from Aligned Monolingual Corpora. In *Proceedings of the International Workshop on Paraphrasing*, pages 57–64, 2003.

Adrian Iftene. *Textual Entailment*. PhD thesis, Faculty of Computer Science, University of Iaşi, 2009.

Diana Inkpen. A Statistical Model for Near-synonym Choice. *ACM Transactions on Speech and Language Processing*, 4(1):2, 2007.

Diana Inkpen and Graeme Hirst. Building and Using a Lexical Knowledge Base of Near-synonym Differences. *Computational Linguistics*, 32(2):223–262, 2006.

Naomi Inoue. Automatic Noun Classification by Using Japanese-English Word Pairs. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 201–208, 1991.

Kentaro Inui and Ulf Hermjakob, editors. *Proceedings of the Second International Workshop on Paraphrasing*. 2003.

Kentaro Inui and Masaru Nogami. A Paraphrase-Based Exploration of Cohesiveness Criteria. In *Proceedings of the European workshop on Natural Language Generation (ENLG)*, pages 1–10, 2001.

R. S. Jackendoff. The Status of Thematic Relations in Linguistic Theory. *Linguistic Inquiry*, 17:369–411, 1987.

R. S. Jackendoff. *Semantic Structures*. The MIT Press, Cambridge, MA, 1990.

Christian Jacquemin. Syntagmatic and Paradigmatic Representations of Term Variation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 341–348, 1999.

Cordeiro João, Gaël Dias, and Brazdil Pavel. A Metric for Paraphrase Detection. In *Proceedings of the The Second International Multi-Conference on Computing in the Global Information Technology*, 2007a.

Cordeiro João, Gaël Dias, and Brazdil Pavel. New Functions for Unsupervised Asymmetrical Paraphrase Detection. *Journal of Software*, 2(4), 2007b.

Rosie Jones, Benjamin Rey, Omid Madani, and Wile Greiner. Generating Query Substitutions. In *Proceedings of the World Wide Web Conference*, 2006.

David Kauchak and Regina Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 455–462, 2006.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of Machine Translation Summit*, 2005.

Philipp Koehn. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2003.

Stanley Kok and Chris Brockett. Hitting the Right Paraphrases in Good Time. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.

Irene Langkilde. Forest-based statistical sentence generation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2000.

Alon Lavie and Abhaya Agarwal. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, 2007.

Dekang Lin. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 768–774, 1998.

Dekang Lin and Lin Pantel. DIRT - Discovery of Inference Rules from Text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2001.

Adam Lopez. Statistical Machine Translation. *ACM Comput. Surv.*, 40(3):1–49, 2008.

Adam Lopez and Philip Resnik. Word-Based Alignment, Phrase-Based Translation: What's the Link? In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 90–99, 2006.

Nitin Madnani and Bonnie J. Dorr. Generating Phrasal & Sentential Paraphrases: A Survey of Data-Driven Methods. *Computational Linguistics*, 2010.

Nitin Madnani, Necip Fazil Ayan, Philip Resnik, and Bonnie J. Dorr. Using Paraphrases for Parameter Tuning in Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007.

Nitin Madnani, Philip Resnik, Bonnie J. Dorr, and Richard Schwartz. Are Multiple Reference Translations Necessary? Investigating the Value of Paraphrased Reference Translations in Parameter Optimization. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, 2008a.

Nitin Madnani, Philip Resnik, Bonnie J. Dorr, and Richard Schwartz. Applying Automatically Generated Semantic Knowledge: A Case Study in Machine Translation. In *Proceedings of the NSF Symposium on Semantic Knowledge Discovery, Organization and Use*, New York, 2008b.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

Erwin Marsi and Emiel Krahmer. Explorations in Sentence Fusion. In *Proceedings of the European Workshop on Natural Language Generation*, 2005a.

Erwin Marsi and Emiel Krahmer. Classification of Semantic Relations by Humans and Machines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 2005b.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 381–390, 2009.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. Discriminative Corpus Weight Estimation for Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–717, 2009.

Aurélien Max. Local Rephrasing Suggestions for Supporting the Work of Writers. In *Proceedings of the 6th International Conference on Natural Language Processing (GoTAL)*, 2008.

Kathleen R. McKeown. Paraphrasing Using Given and New Information in a Question-Answer System. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 1979.

Dan Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT press, 2001.

I. Dan Melamed. A Word-to-Word Model of Translational Equivalence. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 490–497, 1997.

Donald Metzler, Susan Dumais, and Christopher Meek. Similarity Measures for Short Segments of Text. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, 2007.

Haitao Mi, Liang Huang, and Qun Liu. Forest-based translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 192–199, 2008.

Mehryar Mohri and Michael Riley. An Efficient Algorithm for the nbest-strings Problem. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 1313–1316, 2002.

Akiko Murakami and Tetsuya Nasukawa. Term Aggregation: Mining Synonymous Expressions using Personal Stylistic Variations. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 806–812, 2004.

NIST. NIST Open Machine Translation (MT) Evaluation. Information Access Division, 2008. http://www.nist.gov/speech/tests/mt/.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, 2007.

D. W. Oard. The surprise langauge exercises. *ACM Transactions on Asian Language Information Processing*, 2(3), 2003.

Franz Och and Hermann Ney. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4), 2004.

Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2003.

Franz Josef Och and Hermann Ney. Improved Statistical Alignment Models. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 440–447, 2000.

Joesph Olive. Global Autonomous Language Exploitation. DARPA/IPTO Proposer Information Pamphlet, 2005.

Mari Ostendorf, Ashvin Kannan, Steve Austin, Owen Kimball, Richard Schwartz, and Jan Robin Rohlicek. Integration of Diverse Recognition Methodologies through Re-evaluation of N-best Sentence Hypotheses. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 83–87, 1991.

Karolina Owczarzak, Declan Groves, Josef Van Genabith, and Andy Way. Contextual Bitext-derived Paraphrases in Automatic MT Evaluation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 86–93, 2006.

Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2003.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. ISP: Learning Inferential Selectional Preferences. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2007.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2002.

Marius Paşca and Péter Dienes. Aligning Needles In A Haystack: Paraphrase Acquisition Across The Web. In *Proceedings of the International Joint Conference of Natural Language Processing (IJCNLP)*, pages 119–130, 2005.

Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional Clustering of English Words. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 183–190, 1993.

Michael J. D. Powell. An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. *The Computer Journal*, 7: 155–162, 1965.

Richard Power and Donia Scott. Automatic Generation of Large-scale Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, page 5764, 2005.

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes*. Cambridge University Press, Cambridge, 1986.

Chris Quirk, Chris Brockett, and William Dolan. Monolingual Machine Translation for Paraphrase Generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.

Lance Ramshaw and Mitch Marcus. Text Chunking Using Transformation-Based Learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, 1995.

Deepak Ravichandran and Eduard Hovy. Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2002.

Deepak Ravichandran, Patrick Pantel, and Eduard H. Hovy. Randomized Algorithms and NLP: Using Locality Sensitive Hash Function for High Speed Noun Clustering. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 622–629, 2005.

Philip Resnik. Exploiting Hidden Meanings: Using Bilingual Text for Monolingual Annotation. *Lecture Notes in Computer Science 2945: Computational Linguistics and Intelligent Text Processing*, pages 283–299, 2004.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O. Mittal, and Yi Liu. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2007.

Lorenza Romano, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. Investigating a Generic Paraphrase-based Approach for Relation Extraction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 409–416, 2006.

Vasile Rus, Philip M. McCarthy, and Mihai C. Lintean. Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In *Proceedings of the 21st International FLAIRS Conference*, 2008.

Mehran Sahami and Timothy D. Heilman. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. In *Proceedings of the World Wide Web Conference*, 2006.

Satoshi Sekine. Automatic Paraphrase Discovery Based on Context and Keywords Between Ne Pairs. In *Proceedings of the International Workshop on Paraphrasing*, pages 80–87, 2005.

Satoshi Sekine. On-demand Information Extraction. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 731–738, 2006.

Satoshi Sekine, Kentaro Inui, Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini, editors. *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. 2007.

Libin Shen, Jinxi Xu, and Ralph Weischedel. String-to-Dependency Machine Translation. *Computational Linguistics*, page To appear, 2010.

Siwei Shen, Dragomir R. Radev, Agam Patel, and Güneş Erkan. Adding Syntax to Dynamic Programming for Aligning Comparable Texts for the Generation of Paraphrases. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 747–754, 2006.

Xiaodong Shi and Christopher C. Yang. Mining Related Queries from Web Search Engine Query Logs Using an Improved Association Rule Mining Model. *Journal of the American Society for Information Science and Technology*, 58(12), 2007.

Mitsuo Shimohata and Eiichiro Sumita. Acquiring Synonyms from Monolingual Comparable Texts. In *Proceedings of the International Joint Conference of Natural Language Processing (IJCNLP)*, pages 233–244, 2005.

Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. Automatic Paraphrase Acquisition from News Articles. In *Proceedings of the Human Language Technology Conference (HLT)*, 2002.

Matthew Snover, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, 2006.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate. *Machine Translation*, 23(2–3):117–127, 2009.

Frank K. Soong and Eng-Fong Huang. A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition. In *Proceedings of the HLT workshop on Speech and Natural Language*, pages 12–19, 1990.

Karen Spärck-Jones and J. I. Tait. Automatic Search Term Variant Generation. *Journal of Documentation*, 40(1):50–66, 1984.

Mark Steedman, editor. *Surface Structure and Interpretation (Linguistic Inquiry Monograph No. 30)*. MIT Press, 1996.

S. Strassel, C. Cieri, A. Cole, D. DiPersio, M. Liberman, X. Ma, M. Maamouri, and K. Maeda. Integrated linguistic resources for language exploitation technologies. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2006.

Idan Szpektor, Eyal Shnarch, and Ido Dagan. Instance-based Evaluation of Entailment Rule Acquisition. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2007.

Takaaki Tanaka, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors. *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*. 2004.

Özlem Uzuner and Boris Katz. Capturing Expression Using Linguistic Information. In *Proceedings of the Meeting of the American Association for Artificial Intelligence (AAAI)*, 2005.

Peter Wallis. Information Retrieval Based on Paraphrase. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 118–126, 1993.

Dekai Wu. Recognizing Paraphrases and Textual Entailment Using Inversion Transduction Grammars. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 2005.

Hua Wu and Ming Zhou. Synonymous Collocation Extraction Using Translation Information. In *Proceedings of the ACL Workshop on Multiword Expressions: Integrating Processing*, pages 120–127, 2003.

Omar F. Zaidan and Chris Callison-Burch. Feasibility of human-in-the-loop minimum error rate training. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 52–61, 2009.

Ying Zhang and Stephan Vogel. Measuring Confidence Intervals for the Machine Translation Evaluation Metrics. In *Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, 2004.

Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. Combining Multiple Resources to Improve SMT-based Paraphrasing Model. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 1021–1029, 2008.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. Application-driven Statistical Paraphrase Generation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 834–842, 2009.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. Re-evaluating Machine Translation Results With Paraphrase Support. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 77–84, 2006a.

Liang Zhou, Chin-Yew Lin, Dragos Stefan Muntenau, and Eduard Hovy. ParaEval: Using Paraphrases to Evaluate Summaries Automatically. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2006b.