Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods

Nitin Madnani*
University of Maryland, College Park

Bonnie J. Dorr**
University of Maryland, College Park

The task of paraphrasing is inherently familiar to speakers of all languages. Moreover, the task of automatically generating or extracting semantic equivalences for the various units of language—words, phrases, and sentences—is an important part of natural language processing (NLP) and is being increasingly employed to improve the performance of several NLP applications. In this article, we attempt to conduct a comprehensive and application-independent survey of data-driven phrasal and sentential paraphrase generation methods, while also conveying an appreciation for the importance and potential use of paraphrases in the field of NLP research. Recent work done in manual and automatic construction of paraphrase corpora is also examined. We also discuss the strategies used for evaluating paraphrase generation techniques and briefly explore some future trends in paraphrase generation.

1. Introduction

Although 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. In this survey, we make an attempt at such an elaboration. An important outcome of this survey is the discovery that there are a large variety of paraphrase generation methods, each with widely differing sets of characteristics, in terms of performance as well as ease of deployment. We also find that although many paraphrase methods are developed with a particular application in mind, all methods share the potential for more general applicability. Finally, we observe that the choice of the most appropriate method for an application depends on proper matching of the characteristics of the produced paraphrases with an appropriate method.

It could be argued that it is premature to survey an area of research that has shown promise but has not yet been tested for a long enough period (and in enough systems). However, we believe this argument actually strengthens the motivation for a survey

* Department of Computer Science and Institute for Advanced Computer Studies, A.V. Williams Bldg, University of Maryland, College Park, MD 20742, USA. E-mail: nmadnani@umiacs.umd.edu.
** Department of Computer Science and Institute for Advanced Computer Studies, A.V. Williams Bldg, University of Maryland, College Park, MD 20742, USA. E-mail: bonnie@umiacs.umd.edu.

Submission received: 16 December 2008; revised submission received: 30 November 2009; accepted for publication: 7 March 2010.

that can encourage the community to use paraphrases by providing an application-independent, cohesive, and condensed discussion of data-driven paraphrase generation techniques. We should also acknowledge related work that has been done on furthering the community's understanding of paraphrases. Hirst (2003) presents a comprehensive survey of paraphrasing focused on a deep analysis of the nature of a paraphrase. The current survey 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 remainder of this section formalizes the concept of a paraphrase, scopes out the coverage of this survey's discussion, and provides broader context and motivation by discussing applications in which paraphrase generation has proven useful, along with examples. Section 2 briefly describes the tasks of paraphrase recognition and textual entailment and their relationship to paraphrase generation and extraction. Section 3 forms the major contribution of this survey by examining various corpora-based techniques for paraphrase generation, organized by corpus type. Section 4 examines recent work done to construct various types of paraphrase corpora and to elicit human judgments for such corpora. Section 5 considers the task of evaluating the performance of paraphrase generation and extraction techniques. Finally, Section 6 provides a brief glimpse of the future trends in paraphrase generation and Section 7 concludes the survey with a summary.

1.1 What is a Paraphrase?

The concept of paraphrasing is most generally defined on the basis of the principle of semantic equivalence: 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.

Individual lexical items having the same meaning are usually referred to as **lexical paraphrases** or, more commonly, **synonyms**, for example, ⟨*hot*, *warm*⟩ and ⟨*eat*, *consume*⟩. However, lexical paraphrasing cannot be restricted strictly to the concept of synonymy. There are several other forms such as **hyperonymy**, where one of the words in the paraphrastic relationship is either more general or more specific than the other, for example, ⟨*reply*, *say*⟩ and ⟨*landlady*, *hostess*⟩.

The term **phrasal paraphrase** refers to phrasal fragments sharing the same semantic content. Although these fragments most commonly take the form of syntactic phrases (⟨*work on*, *soften up*⟩ and ⟨*take over*, *assume control of*⟩) they may also be patterns with linked variables, for example, ⟨*Y was built by X*, *X is the creator of Y*⟩.

Two sentences that represent the same semantic content are termed **sentential paraphrases**, for example, ⟨*I finished my work*, *I completed my assignment*⟩. Although 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 generate more interesting ones such as ⟨*He needed to make a quick decision in that situation*, *The scenario required him to make a split-second judgment*⟩. Culicover (1968) describes some common forms of sentential paraphrases.

1.2 Scope of Discussion

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 task, an unfortunate but necessary remedy is to impose certain limits on the scope of our discussion. In this survey, we will be restricting our discussion to only 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, we do include a relatively comprehensive list of references in this section for the work that is excluded from the survey.

For one, we do not discuss paraphrasing techniques that rely primarily on knowledge-based resources such as dictionaries (Wallis 1993; Fujita et al. 2004), hand-written rules (Fujita et al. 2007), and formal grammars (McKeown 1979; Dras 1999; Gardent, Amoia, and Jacquey 2004; Gardent and Kow 2005). We also refrain from discussing work on purely lexical paraphrasing which usually comprises various ways to cluster words occurring in similar contexts (Inoue 1991; Crouch and Yang 1992; Pereira, Tishby, and Lee 1993; Grefenstette 1994; Lin 1998; Gasperin et al. 2001; Glickman and Dagan 2003; Shimohata and Sumita 2005).[1] Exclusion of general lexical paraphrasing methods obviously implies that other lexical methods developed just for specific applications are also excluded (Bangalore and Rambow 2000; Duclaye, Yvon, and Collin 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; Hallett and Scott 2005; Power and Scott 2005). Finally, we also do not discuss the notion of near-synonymy (Hirst 1995; Edmonds and Hirst 2002).

1.3 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 applications we discuss subsequently, the use of paraphrases in the manner described may not yet be the norm. However, wherever applicable, we cite recent research that promises gains in performance by using paraphrases for these applications. Also note that we only discuss those paraphrasing techniques that can generate the types of paraphrases under examination in this survey: phrasal and sentential.

1.3.1 Query and Pattern Expansion. One of the most common applications of paraphrasing is the automatic generation of query variants for submission to information retrieval systems or of patterns for submission to information extraction systems. Culicover (1968) describes one of the earliest theoretical frameworks for query keyword expansion using paraphrases. One of the earliest works to implement this approach (Spärck-Jones and Tait 1984) generates several simple variants for compound nouns in queries submitted to a technical information retrieval system. For example:

Original : *circuit details*

Variant 1 : *details about the circuit*

Variant 2 : *the details of circuits*

1 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 3.1.

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 Hellman 2006; Metzler, Dumais, and Meek 2007; Shi and Yang 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, and 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 query (via a pivot-based sentential paraphrasing model employing bilingual parallel corpora, detailed in Section 3) 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 1 : *X completed its acquisition of Y*

Variant 2 : *X purchased Y*

1.3.2 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 given system output *S* 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 (Owczarzak et al. 2006; Zhou, Lin, and Hovy 2006).

In evaluation of document summarization, automatically generated summaries (**peers**) are also evaluated against reference summaries created by human authors (**models**). Zhou et al. (2006) 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.

1.3.3 Machine Translation. Besides being used in evaluation of machine translation systems, paraphrasing has also been applied to directly improve the translation process. Callison-Burch, Koehn, and Osborne (2006) 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 in 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 translation for the given phrase. Therefore, paraphrasing allows the translation system to properly handle phrases that it does not otherwise know how to translate.

Another important issue for statistical machine translation systems is that of **reference sparsity**. The fundamental problem that translation systems have to face is that there is no such thing as *the* correct translation for any sentence. In fact, any given source sentence can often be translated into the target language in many valid ways. Because there can be many "correct answers," almost all models employed by SMT systems require, in addition to a large bitext, a held-out development set comprising multiple high-quality, human-authored reference translations in the target language in order to tune their parameters relative to a translation quality metric. However, given

the time and cost implications of such a process, most such data sets usually have only a single reference translation. Madnani et al. (2007, 2008b) generate sentential paraphrases and use them to expand the available reference translations for such sets so that the machine translation system can learn a better set of system parameters.

2. Paraphrase Recognition and Textual Entailment

A problem closely related to, and as important as, generating paraphrases is that of assigning a quantitative measurement to the semantic similarity of two phrases (Fujita and Sato 2008a) 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; Marsi and Krahmer 2005a; Wu 2005; Joào, Das, and Pavel 2007a, 2007b; Das and Smith 2009; Malakasiotis 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 3) 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 the 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 2005b).

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 the Recognizing Textual Entailment (RTE) Challenges (Dagan, Glickman, and Magnini 2006; Bar-Haim et al. 2007; Sekine et al. 2007; Giampiccolo et al. 2008).

Looking towards the future, Dagan (2008) suggests 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. Although 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.*

Although both H_1 and H_2 are entailed by T, they are not strictly paraphrases of T because some of the semantic content has not been 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.

The relationship between paraphrasing and entailment is more tightly entwined than it might appear. Entailment recognition systems sometimes rely on the use of 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 data sets 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) from 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, McCarthy, and Lintean 2008).

Detection of semantic similarity and, to some extent, that of bidirectional entailment is usually an implicit part of paraphrase generation. However, given the interesting and diverse work that has been done in both these areas, we feel that any significant discussion beyond the treatment above merits a separate, detailed survey.

3. Paraphrasing with Corpora

In this section, we explore in detail the data-driven paraphrase generation approaches that have emerged and have 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.

We organize this section 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 organization, in our opinion, is the most instructive because 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, we believe that it would be very useful to explain the motivation behind distributional similarity, an extremely popular technique that can be used for paraphrase generation with several different types of corpora. We devote the following section to such an explanation.

3.1 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.

More specifically, Harris presents several empirical observations to support the hypothesis that such a structure exists naturally for language. Here, we closely quote these observations:

- 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 the distributions of the first pair will also be more alike than that of the latter pair.

Given these observations, it is relatively easy to characterize the concept of **distributional similarity**: words or phrases that share the same distribution—the same set of words in the same context in a corpus—tend to have similar meanings.

Figure 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). Although 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

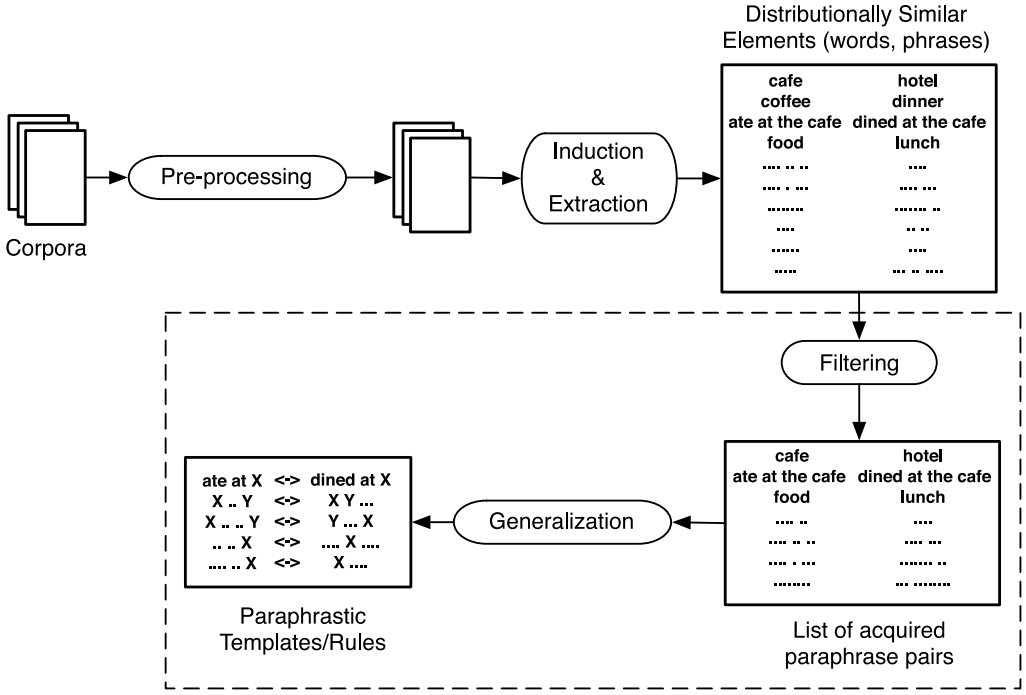

Figure 1
A general architecture for paraphrasing approaches leveraging the distributional similarity hypothesis.

elements of the pairs ⟨*boys*, *girls*⟩, ⟨*cats*, *dogs*⟩, ⟨*high*, *low*⟩ can occur in similar contexts but are not semantically equivalent.

3.2 Paraphrasing Using a Single Monolingual Corpus

In this section, we concentrate 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 of distributional similarity as outlined in the previous section. Besides the obvious caveat discussed previously regarding distributional similarity, we find that the other most important factor affecting the performance of these methods is the choice of distributional ingredients—that is, the features used to formulate the distribution of the extracted units. We consider three commonly used techniques 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 and the other two use features derived from additional semantic knowledge. Although the latter two methods are able to generate more sophisticated paraphrases by virtue of more specific and more informative ingredients, we find that 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. Although 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 the use of Web documents as the input corpus: It does not require 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, that is, a pair sharing one or more anchors, is counted. The final set of phrasal paraphrastic pairs is returned.

This algorithm embodies the spirit of the hypothesis of distributional similarity: It considers all words and phrases that are distributionally similar—those that 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

Algorithm 1 (Paşca and Dienes 2005). Induce a set of phrasal paraphrase pairs H with associated counts from a corpus of pre-processed Web documents.
Summary. Extract all n-grams from the corpus 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.

1: Let N represent a set of n-grams extracted from the corpus
2: $N \leftarrow \{\phi\}, H \leftarrow \{\phi\}$
3: **for** each sentence **E** in the corpus **do**
4: 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 $M_1 \leq M_2$
5: $N \leftarrow N \cup N_E$
6: **end for**
7: **for** each n-gram $\bar{e}$ in N **do**
8: Extract the subsequence C, such that $L_c \leq |C| \leq (|\bar{e}| - L_c - 1)$
9: Extract the subsequence A_L, such that $0 \leq |A_L| \leq (L_c - 1)$
10: Extract the subsequence A_R, such that $(|\bar{e}| - L_c) \leq |A_R| \leq (|\bar{e}| - 1)$
11: $A \leftarrow A_L + A_R$
12: Add the pair (A, C) to H
13: **end for**
14: **for** each subset of H with the same anchor A **do**
15: Exhaustively compare each pair of tuples (A, C_i) and (A, C_j) in this subset
16: Update the count of the candidate paraphrase pair (C_i, C_j) by 1
17: Update the count of the candidate paraphrase pair (C_j, C_i) by 1
18: **end for**
19: Output H containing paraphrastic pairs and their respective counts

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:[2]

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

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

Whereas single links between nodes in a dependency tree represent direct semantic relationships, a sequence of links, or a **path**, can be understood to represent an indirect relationship. Here, a path is named by concatenating the dependency relationships and lexical items along the way but excluding the lexical items at the end. 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. One dependency path that we could extract would be between the node *John* and the node *problem*. We start at *John* and see that the first item in the tree is the dependency relation *subject* that connects a noun to a verb and so we append that information to the path.[3] The next item in the tree is the word *found* and we append its lemma (*find*) to the path. Next is the semantic relation *object* connecting a verb to a noun and we append that. The process continues until we reach the other slot (the word *problem*) at which point we stop.[4] The extracted path is shown below the tree. Similarly, we can extract a path 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 an extended version of the distributional similarity hypothesis: 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 which are

2 Technically, these templates represent **inference rules**, such that the right-hand side can be inferred from the left-hand side but is not semantically equivalent to it. This form of inference is closely related to that exhibited in textual entailment. This work is primarily concerned with inducing such rules rather than strict paraphrases.

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

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

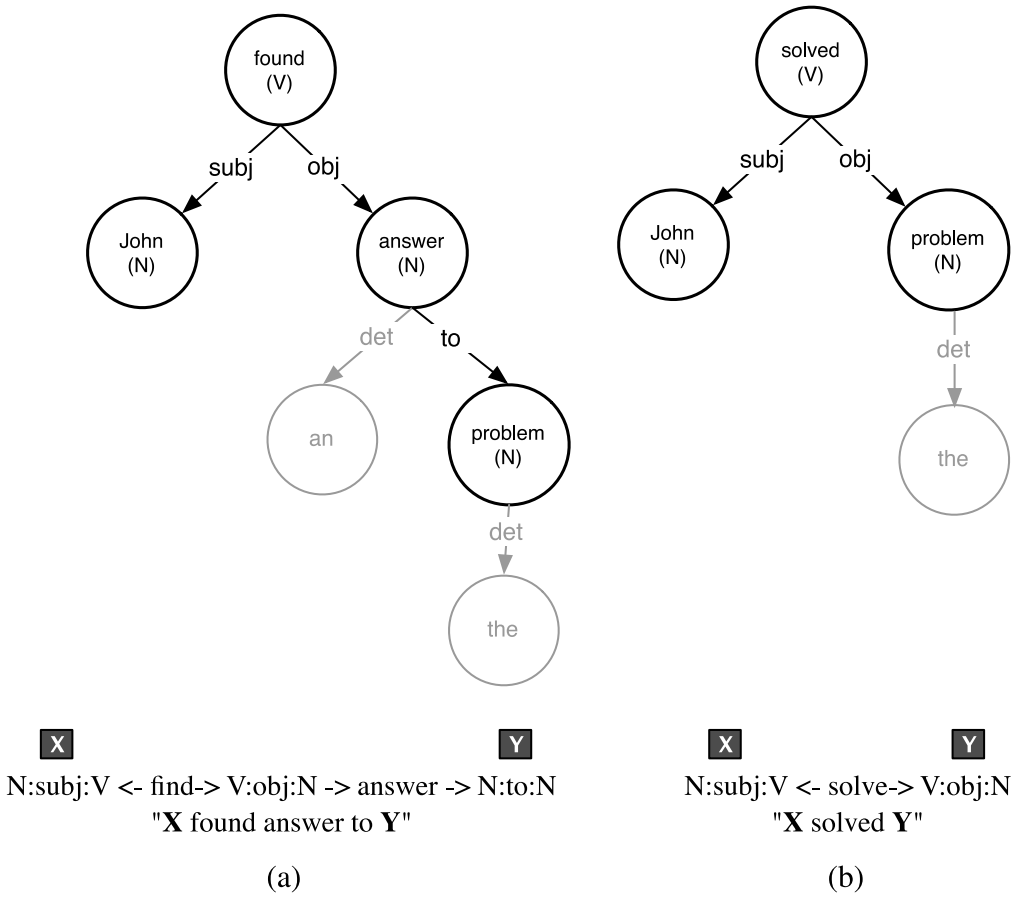

Figure 2
Two different dependency tree paths (a and b) that are considered paraphrastic because the same words (*John* and *problem*) are used to fill the corresponding slots (shown co-indexed) in both the paths. The implied meaning of each dependency path is also shown.

similar, insofar as such properties are concerned.[5] At the end, we have sets of paths (or inference rules) that are considered to have similar meanings by the algorithm.

The performance of their dependency-path based algorithm depends heavily on the root of the extracted path. For example, whereas 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).

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

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

Algorithm 2 (Lin and Pantel 2001). Produce inference rules from a parsed corpus.
Summary. Adapt Harris's (1954) hypothesis of distributional similarity for paths in dependency trees: If two tree paths have similar distributions such that they tend to link the same set of words, then they 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:

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

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**

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 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.[6]

6 The semantic relation R is deemed to be invariant under translation.

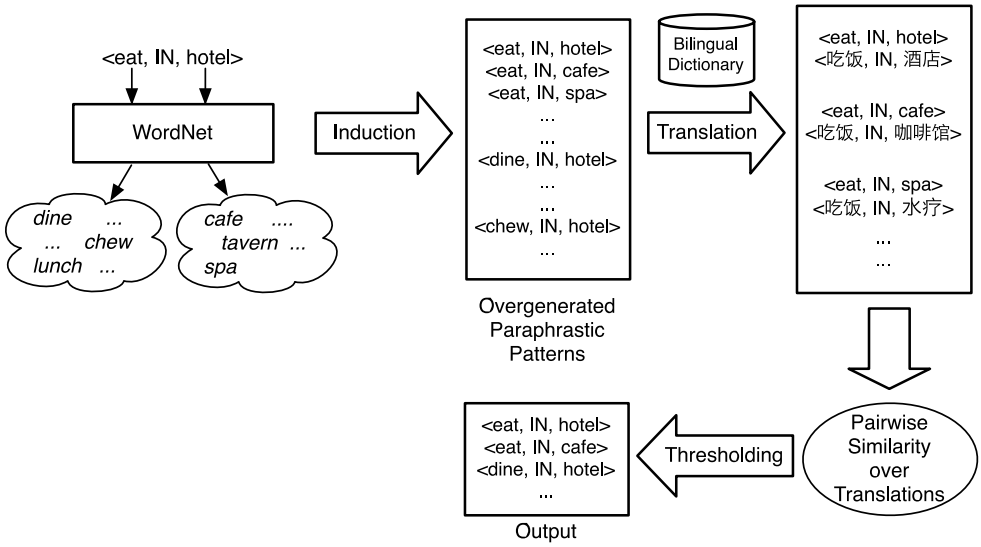

Figure 3
Using Chinese translations as the distributional elements to extract a set of English paraphrastic patterns from a large English corpus.

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 exceeds 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 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 are lower than those obtained by other methods.

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

3.3 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

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. We present four techniques 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.

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 components 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 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.*

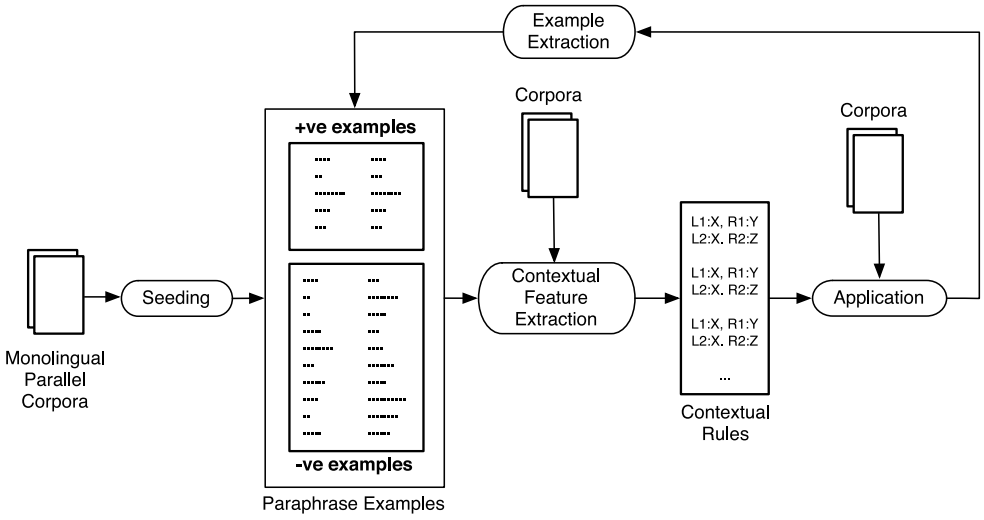

Figure 4
A bootstrapping algorithm to extract phrasal paraphrase pairs from monolingual parallel corpora.

⟨*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, we can extract 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 for the positive example ⟨*tried*, *tried*⟩. 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 to the left of the word *tried* is a personal pronoun (*he*) and the POS tag sequence to the right of *tired* is the preposition *to*. The second tuple is identical for this case. Note that the tags of identical tokens are indicated as such by subscripts 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 this 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 9,000 phrasal (including lexical) paraphrases are extracted from 11 translations of five 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. We present similar observations in Section 3.5 and highlight that although 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, Katz, and Lin (2003) propose an approach that applies a modified version of the dependency path distributional similarity algorithm proposed by Lin and Pantel (2001) to the same 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 approach simply because their technique 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 is 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, Knight, and Marcu (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, and 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.

As a pre-processing step, any group (of 11 sentences) that contains sentences longer than 45 words is 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 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

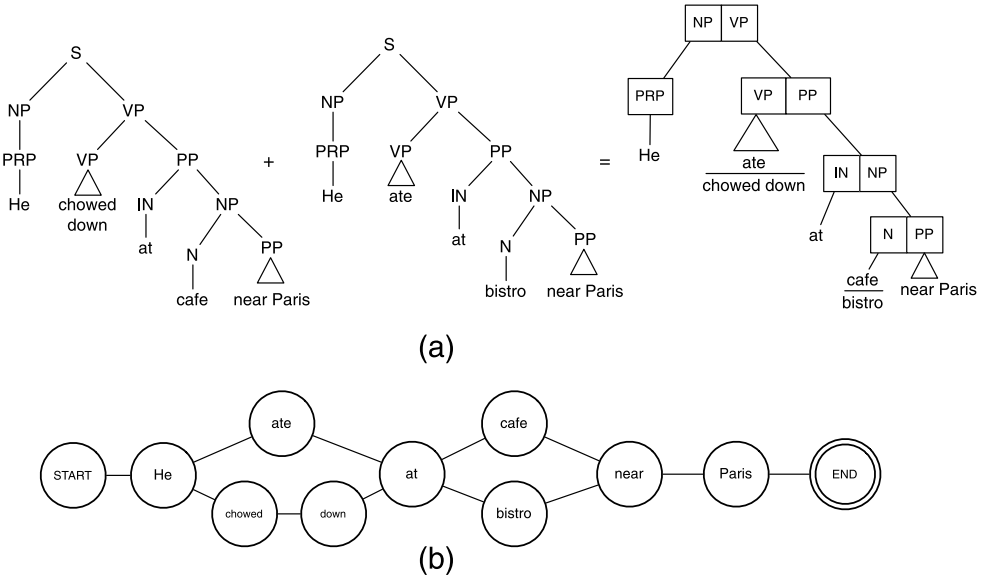

Figure 5
The merging algorithm. (a) How the merging algorithm works for two simple parse trees to produce a shared forest. Note that for clarity, not all constituents are expanded fully. Leaf nodes with two entries represent paraphrases. (b) The word lattice generated by linearizing the forest in (a).

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 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 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 because no form of generalization is performed to convert lattices into patterns.

- **Requirement of a large number of human-written translations.** Each of the lattices described 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 four references, which would certainly lead to much sparser word lattices and smaller numbers 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, Graff, and Doddington 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, Brockett, and Dolan (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, Quirk, Brockett, and Dolan's approach to sentential paraphrase generation may be expressed in terms of the typical channel model equation for statistical machine translation:

$$\mathbf{E_p^*} = \arg\max_{\mathbf{E_p}} P(\mathbf{E_p}|\mathbf{E}) \tag{1}$$

The equation denotes the search for the optimal paraphrase $\mathbf{E_p}$ for a given sentence $\mathbf{E}$. We may use Bayes' Theorem to rewrite this as:

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

where $P(\mathbf{E_p})$ is an n-gram language model providing a probabilistic estimate of the fluency of a hypothesis $\mathbf{E_p}$ and $P(\mathbf{E}|\mathbf{E_p})$ is the translation model, 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 because unsupervised induction of word alignments typically requires a 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 on-line 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, the authors 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 following examples show that the proposed method can work well:

S_1: *In only 14 days, U.S. researchers have created an artificial bacteria-eating virus 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, we refer the reader to Dolan, Quirk, and Brockett (2004) and, more specifically, to Section 4. Algorithm 3 shows how to

---

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.

1: $M \leftarrow \{\phi\}$
2: Compute lexical replacement probabilities $P(e_1|e_2)$ from all sentence pairs in C via IBM Model 1 estimation
3: 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 $a_i \in \{0 \dots n\}, m = |\mathbf{E_1}|, n = |\mathbf{E_2}|$
4: **for** each word-aligned sentence pair $(\mathbf{E_1}, \mathbf{E_2})_\mathbf{a}$ in C **do**
5: 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) $\forall i \in \{1, \dots, |\bar{e_2}|\}\ \exists j \in \{1, \dots, |\bar{e_1}|\},\ e_{2,i} \overset{\mathbf{a}}{\sim} e_{1,j}$

6: Add all extracted pairs to M
7: **end for**
8: **for** each paraphrase pair $(\bar{e_1}, \bar{e_2})$ in M **do**
9: Compute $P(\bar{e_1}|\bar{e_2}) = \prod_{e_1^j \in \bar{e_1}} \sum_{e_2^k \in \bar{e_2}} P(e_1^j|e_2^k)$
10: **end for**
11: Output M containing paraphrastic pairs and associated probabilities

---

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 **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.

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 = the 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 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, Brockett, and Dolan (2004) 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

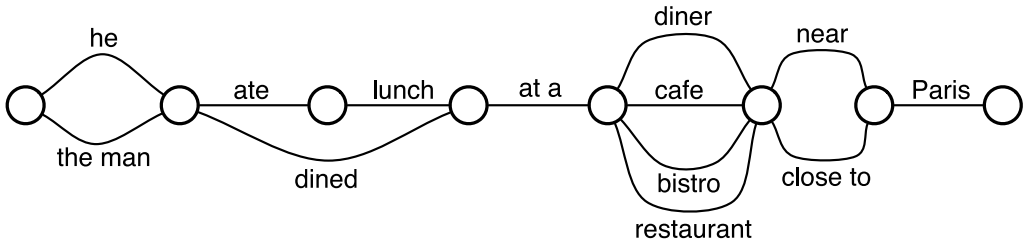

Figure 6
A paraphrase generation lattice for the sentence *He ate lunch at a cafe near Paris*. Alternate paths between various nodes represent phrasal replacements. The probability values associated with each edge are not shown for the sake of clarity.

sentence, and there is no limit on the number of sentences that can be paraphrased.[7] 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. In other words, 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 these limitations combined lead to paraphrases that, although grammatically sound, contain very little variety. Most sentential paraphrases that are generated involve little more than simple substitutions of words and short phrases. In Section 3.5, we will discuss other approaches 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.

3.4 Paraphrasing Using Monolingual Comparable Corpora

Whereas it is clearly to our advantage to have monolingual parallel corpora, such corpora are usually not very readily 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 because 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.

In this section, we describe 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,

7 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.

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, find the most "similar" article from the other set, based on a similarity measure computed over the named entities appearing in the two articles.
2. From each 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) gets its own variable. For example, the following sentence:[8]

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

   may give us the following two patterns, among others:

   ⟨PERSON⟩ *of* ⟨ORGANIZATION⟩ *was promoted*
   ⟨PERSON⟩ *was promoted to* ⟨POST⟩
3. Find all sentences in the two newswire corpora that match these 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.
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, the algorithm 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 (see the discussion regarding Paşca and Dienes [2005] in Section 3.2). However, it has some obvious disadvantages. The authors manually evaluate the technique by generating paraphrases for two specific

8 Although the authors provide motivating examples in Japanese (transliterated into romaji) in their paper, we choose to use English here for the sake of clarity.

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. In addition, if the average number of entities in sentences is low, the likelihood of creating incorrect paraphrases is confirmed to be higher.

Let us now 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 some details of how 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 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 three or more protein or DNA sequences, is as follows:[9]

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, that is, several distinct parallel paths, may correspond to template arguments and can be collapsed into one 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 of 30% is used, 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 7 shows how a very simple word lattice may be generalized into a slotted lattice.

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

9 For more details on MSA algorithms, refer to Gusfield (1997) and Durbin et al. (1998).

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 $W_{C_i,k}$, that is, the nodes that are shared by a majority ($\geq$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 pre-stipulated 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

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.

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

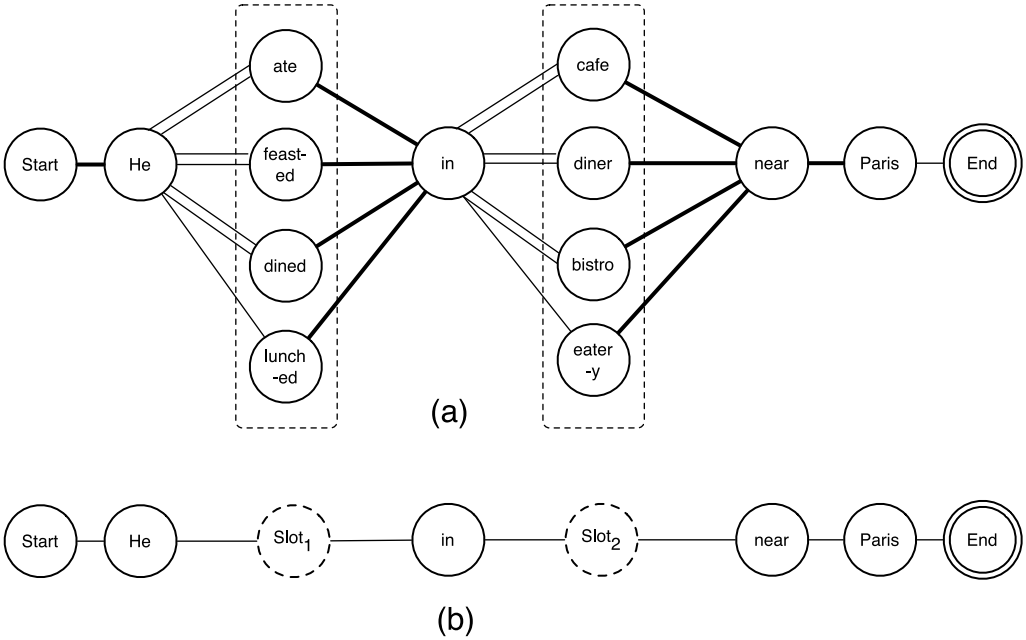

Figure 7
An example showing the generalization of the word lattice (a) into a slotted lattice (b). The word lattice is produced by aligning seven sentences. Nodes having in-degrees > 1 occur in more than one sentence. Nodes with thick incoming edges occur in all sentences.

multiple rewritings (paraphrases) for S. Rewriting in this context is a simple matter of substitution: For each slot in the matching lattice, we know not only the argument from the sentence that fills it but also the slot in the corresponding rewriting lattice.

As far as the quality of acquired paraphrases is concerned, this approach easily outperforms almost all other sentential paraphrasing approaches surveyed in this article. 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, the construction and generalization of lattices may become computationally expensive when dealing with much larger corpora.

We can also compare and contrast Barzilay and Lee's work and the work from Section 3.3 that seems most closely related: that of Pang, Knight, and Marcu (2003). Both take sentences grouped together in a cluster and align them into a lattice using a particular algorithm. Pang, Knight, and Marcu 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 because their input corpus has no pre-defined 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, Knight, and Marcu 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, Knight, and Marcu 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, Knight, and Marcu 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). If a word has the tag I-NP, we can infer 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 this 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. (2006) 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 would not 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.

3.5 Paraphrasing Using Bilingual Parallel Corpora

In the last decade, there has been a resurgence in research on statistical machine translation. There has 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. In this section, we look at such approaches that rely on the preservation of meaning across languages and try to recover said meaning by using cues from the second 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 (or word) alignment techniques.

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, Och, and Marcu 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 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 correspond to the same foreign phrase and infer them all to be paraphrases of each other.[10] For example, the English paraphrase pair ⟨*effectively contained, under control*⟩ is obtained from Figure 8 by pivoting on the Chinese phrase 有效 遏制, shown underlined for both matrices.

10 Note that it would have been equally easy to pivot on the English side and generate paraphrases in the other language instead.

Algorithm 5 (Bannard and Callison-Burch 2005). Generate set M of monolingual paraphrase pairs given a bilingual parallel 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 **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 $p(\bar{e}_j|\bar{e}_l) = \sum_{\bar{f}} p(\bar{e}_j|\bar{f}) p(\bar{f}|\bar{e}_l)$
end for
Output M containing paraphrastic pairs and associated probabilities

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:

$$p(\bar{e}_j|\bar{e}_k) = \sum_{\bar{f}} p(\bar{e}_j, \bar{f}|\bar{e}_k) \approx \sum_{\bar{f}} p(\bar{e}_j|\bar{f}) p(\bar{f}|\bar{e}_k)$$

where both $p(\bar{e}_j|\bar{f})$ and $p(\bar{f}|\bar{e}_k)$ can be computed using maximum likelihood estimation as part of the bilingual phrasal extraction process:

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

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

Bannard and Callison-Burch (2005) 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 used for aligning phrase pairs. The authors find via manual evaluation that the quality of the phrasal

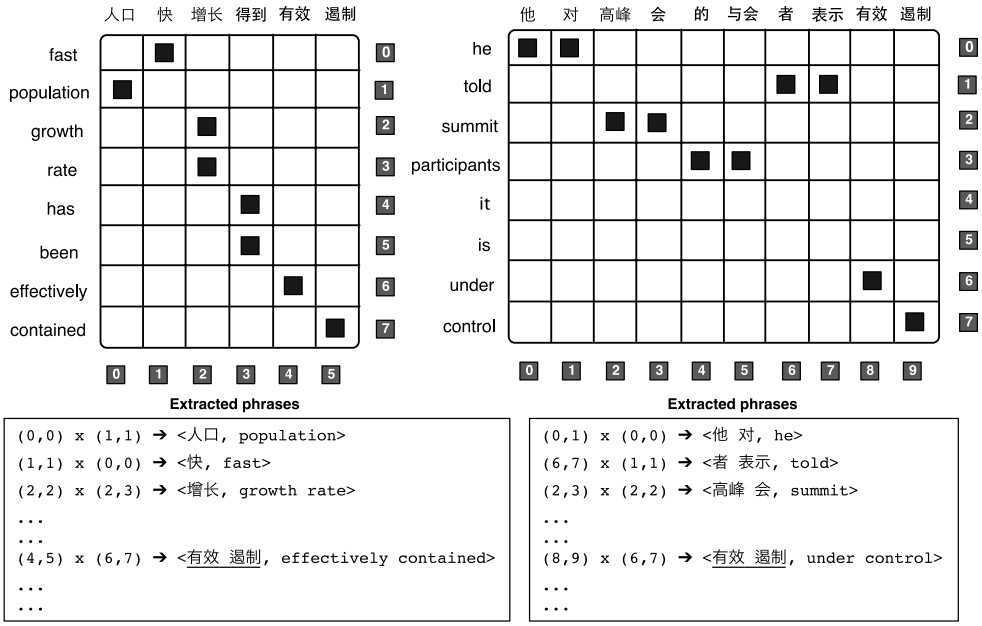

Figure 8
Extracting consistent bilingual phrasal correspondences from the shown sentence pairs. $(i_1, j_1) \times (i_2, j_2)$ denotes the correspondence $\langle f_{i_1} \ldots f_{j_1}, e_{i_2} \ldots e_{j_2} \rangle$. Not all extracted correspondences are shown.

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, Talbot, and Osborne 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 susceptible to the shortcomings of the word alignment techniques. To determine how noisy automatic word alignments affect paraphrasing done via bilingual corpora, we inspected a sample of paraphrase pairs that were extracted when using Arabic—a language significantly different from English—as the pivot language.[11] In this study, we 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 of one of the words in the phrases and cannot really be considered paraphrases. Examples: ⟨*ten ton*, *ten tons*⟩, ⟨*caused clouds*, *causing clouds*⟩.

11 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 herein to the bilingual phrase pairs.

(b) **Approximate Phrasal Paraphrases**. These are pairs that only share 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 were 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 (b) is incorrectly induced alignments between the English and Arabic words, and hence, phrases. Therefore, a good portion of subsequent work on paraphrasing using bilingual corpora, as discussed below focuses on using additional machinery or evidence to cope with the noisy alignment process. Before we continue, we believe it would be useful to draw a connection between Bannard and Callison-Burch's (2005) work and that of Wu and Zhou (2003) as discussed in Section 3.2. 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.[12] 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 likelihood estimation is employed to compute the two component probabilities:

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

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

12 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`.

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 our manual inspection of pivot-based paraphrases above. Indeed, the authors conduct a manual evaluation to show that the syntactically constrained paraphrase pairs are better than those produced 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.

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

An obvious extension of the Callison-Burch style approach is to use the collection of pivoted English-to-English phrase pairs to generate *sentential* paraphrases for new sentences. Madnani et al. (2008a) combine the pivot-based approach to paraphrase acquisition with a well-defined English-to-English translation model that is then used in an (unmodified) SMT system, yielding sentential paraphrases by means of "translating" input English sentences. However, instead of fully lexicalized phrasal correspondences as in (Bannard and Callison-Burch 2005), the fundamental units of translation (and paraphrasing) are **hierarchical** phrase pairs. The latter can be extracted from the former by replacing aligned sub-phrases with non-terminal symbols. For example, given the **initial** phrase pair ⟨增长 得到 有效 遏制, *growth rate has been effectively contained*⟩, the hierarchical phrase pair ⟨X_1 有效 X_2, X_1 *has been* X_2⟩ can be formed.[13] Each hierarchical phrase pair can also have certain features associated with it that are estimated via maximum likelihood estimation during the extraction process. Such phrase pairs can formally be considered the rules of a bilingual **synchronous context-free grammar** (SCFG). Translation with SCFGs is equivalent to parsing the string in the source language using these rules to generate the highest-scoring tree and then reading off the tree in target order. For the purposes of this survey, it is sufficient to state that efficient

13 The process of converting an initial phrase into a hierarchical one is subject to several additional constraints on the lengths of the initial and hierarchical phrases and the number and position of non-terminals in the hierarchical phrase.

methods to extract such rules, to estimate their features, and to translate with them are now well established. For more details on building SCFG-based models and translating with them, we refer the reader to (Chiang 2006, 2007).

Once a set of bilingual hierarchical rules has been extracted along with associated features, the pivoting trick can be applied to infer monolingual hierarchical paraphrase pairs (or paraphrastic patterns). However, the patterns are not the final output and are actually used as rules from a monolingual SCFG grammar in order to define an English-to-English translation model. Features for each monolingual rule are estimated in terms of the features of the bilingual pairs that the rule was inferred from. A sentential paraphrase can then be generated for *any* given sentence by using this model along with an *n*-gram language model and a regular SMT decoder to paraphrase (or monolingually translate) *any* sentence just as one would translate bilingually.

The primary advantage of this approach is the ability to produce good quality sentential paraphrases by leveraging the SMT machinery to address the noise issue. However, although the decoder and the language model do serve to counter the noisy word alignment process, they do so only to a degree and not entirely.

Again, we must draw a connection between this work and that of Quirk, Brockett, and Dolan (2004) (discussed in Section 3.3) because both treat paraphrasing as monolingual translation. However, as outlined in the discussion of that work, Quirk, Brockett, and Dolan use a relatively simplistic translation model and decoder which leads to paraphrases with little or no lexical variety. In contrast, Madnani et al. use a more complex translation model and an unmodified state-of-the-art SMT decoder to produce paraphrases that are much more diverse. Of course, the reliance of the latter approach on automatic word alignments does inevitably lead to much noisier sentential paraphrases than those produced by Quirk, Brockett, and Dolan.

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. In order to extract paraphrases, they sample random paths in the graph from any English node to another. Note that the traditional pivot step is equivalent to 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 to a different pair of six languages. Similar to the single-table case, each phrase in each table is represented as a node in a 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*

In general, each edge is associated with a weight corresponding to the bilingual phrase translation probability. Random walks are then sampled from the graph in such a way 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. Astute readers will make the following 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 described in Callison-Burch (2008) without requiring a parser.

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 that for the syntactically constrained pivot-based approach. However, they perform no comparisons of their coverage with the original pivot-based approach (Bannard and Callison-Burch 2005).

4. Building Paraphrase Corpora

Before we present some specific techniques from the literature that have been employed to evaluate paraphrase generation methods, it is important to examine some recent work that has been done on constructing paraphrase corpora. As part of this work, human subjects are generally asked to judge whether two given sentences are paraphrases of each other. We believe that a detailed examination of this manual evaluation task provides an illuminating insight into the nature of a paraphrase in a practical, rather than a theoretical, context. In addition, it has obvious implications 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 5,801 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 aspects of the task were 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. For example:

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 Minelli's 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]

Although 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 2,031 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 (rewriting light-verb constructions, for example) and then manually create a set of morpho-syntactic paraphrasing rules and patterns describing each type of paraphrasing phenomenon. A paraphrase generation system

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

using these rules (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. 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 the evaluation of paraphrase recognition techniques.

Most recently, Cohn, Callison-Burch, and Lapata (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 SMT systems that were illustrated in Section 3.5. 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 we have already described elsewhere in this survey: (1) the sentence pairs judged equivalent from the MSRP Corpus: (2) the Multiple Translation Chinese (MTC) corpus of multiple human-written translations of Chinese news stories used by Pang, Knight, and Marcu (2003); and (3) 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

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

alignments, if present). Two parts were said to 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, we refer the reader to (Callison-Burch, Cohn, and Lapata 2006).

Extensive experiments are conducted to measure inter-annotator agreements and obtain good agreement values but they 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, because 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 lexicalized 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 the authors show that they can learn paraphrastic patterns such as those shown in Figure 9.

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

5. Evaluation of Paraphrase Generation

Whereas other language processing tasks such as machine translation and document summarization usually have multiple annual community-wide evaluations using

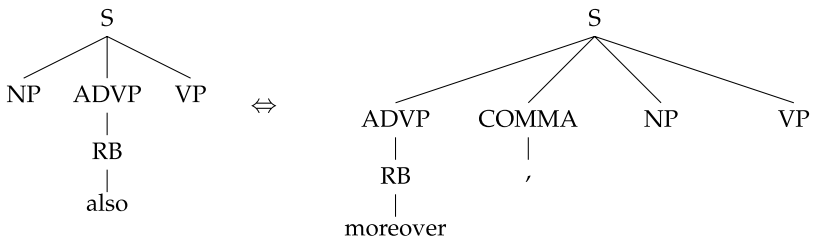

Figure 9
An example of syntactically motivated paraphrastic patterns that can be extracted from the paraphrase corpus constructed by Cohn, Callison-Burch, and Lapata (2008).

standard test sets and manual as well as automated metrics, the task of automated paraphrasing does not. An obvious reason for 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. We believe that the primary reason 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 (for example: 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 survey—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, Katz, and Lin 2003; Pang, Knight, and Marcu 2003). A more direct approach is to substitute 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 as such only in some contexts and not others. Szpektor, Shnarch, and Dagan (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, Brockett, and Dolan 2004). An intrinsic evaluation of this form must employ the usual methods for avoiding any bias and for maximizing inter-judge agreement. In addition, we believe that, given the difficulty of this 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. A number of these approaches 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, Koehn, and Osborne 2006) and others as described in Section 1.

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 paraphrases is essentially the task of measuring semantic similarity or of paraphrase recognition, all of those metrics, including the ones discussed in Section 2, can be employed here.

Most recently, Callison-Burch, Cohn, and Lapata (2008) 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 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 (P_{align}) and recall (R_{align}) measures are computed as follows:

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

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

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 8.

Although 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, Knight, and Marcu (2003) and Quirk, Brockett, and Dolan (2004) for generating sentential paraphrases from monolingual parallel corpora and described in Section 3.3 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.

There has also been some preliminary progress toward using standardized test sets for 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 available.[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 herein, 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

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

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 fora where members of the community may share their ideas with their colleagues and receive valuable feedback. In recent years, a number of such fora have 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 represents an extremely important step toward countering the fragmented usage pattern described previously.

6. Future Trends

It is important for any survey to provide a look to the future of the surveyed task and general trends for the corresponding research methods. We identify several such trends 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 its 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, the 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 similarity 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). Because 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, Pantel, and Hovy 2005). We also believe that the influence of the Web will extend to other avenues of paraphrase generation such as the aforementioned extrinsic evaluation or lack thereof. For example, Fujita and Sato (2008b) 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 proposed by Madnani et al. (2007). A manual inspection reveals that using multiple sources of information yields paraphrases with much higher accuracy. We believe that such exploitation of multiple types of resources and their combinations 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. For example, 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 the former 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 we saw in the preceding pages of this survey. Despite the obvious convenience of using SMT components for the purpose of monolingual translation, we must consider that doing so usually requires additional work to deal with the added noise due to the nature of such components. We believe 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 components 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 for 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. Because 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.

We also believe that a large-scale annual community-wide evaluation should become a trend since it is required to foster further research in, and use of, paraphrase extraction and generation. Although there have been recent workshops and tasks on paraphrasing and entailment as discussed in Section 5, 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 2009).

7. Summary

Over the last two decades, there has been much research on paraphrase extraction and generation within a number of research communities 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 its particular host 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.

In this article, we have presented a comprehensive survey of the task 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 paraphrasing in the field of NLP research. We show that there are a large variety

of paraphrase generation methods and each such method has a very different set of characteristics, in terms of both its performance and its ease of deployment. We also observe that whereas most of the methods in this survey 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.

References

Ayan, Necip Fazil and Bonnie Dorr. 2006. Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proceedings of ACL/COLING*, pages 9–16, Sydney.

Bangalore, Srinivas and Owen Rambow. 2000. Corpus-based lexical choice in natural language generation. In *Proceedings of ACL*, pages 464–471, Hong Kong.

Bannard, Colin and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*, pages 597–604, Ann Arbor, MI.

Bar-Haim, Roy, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor, editors. 2007. *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*, Venice.

Barzilay, Regina and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of EMNLP*, pages 164–171, Philadelphia, PA.

Barzilay, Regina and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of HLT-NAACL 2003*, pages 16–23, Edmonton.

Barzilay, Regina and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of ACL*, pages 50–57, Toulouse.

Barzilay, Regina and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Bayer, Samuel, John Burger, Lisa Ferro, John Henderson, and Alexander Yeh. 2005. MITREs submissions to the EU Pascal RTE Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, pages 41–44, Southampton, U.K.

Beeferman, Doug and Adam Berger. 2000. Agglomerative clustering of a search engine query log. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 407–416, Boston, MA.

Belz, Anja. 2009. That's nice...what can you do with it? *Computational Linguistics*, 35(1):111–118.

Bhagat, Rahul and Deepak Ravichandran. 2008. Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL*, pages 674–682, Columbus, OH.

Bosma, Wauter and Chris Callison-Burch. 2007. Paraphrase substitution for recognizing textual entailment. In *Evaluation of Multilingual and Multimodal Information Retrieval*, Lecture Notes in Computer Science, Volume 4730, Springer-Verlag, pages 502–509.

Brockett, Chris and William B. Dolan. 2005. Support vector machines for paraphrase identification and corpus construction. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 1–8, Jeju Island.

Broder, Andrei. 1997. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*, pages 21–29, Salemo.

Brown, Peter F., John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Buchholz, Sabine and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York, NY.

Callison-Burch, Chris. 2007. *Paraphrasing and Translation*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Callison-Burch, Chris. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP*, pages 196–205, Waikiki, HI.

Callison-Burch, Chris, Trevor Cohn, and Mirella Lapata. 2006. Annotation guidelines for paraphrase alignment. Technical report, University of Edinburgh. `http://www.dcs.shef.ac.uk/~tcohn/paraphrase_guidelines.pdf`.

Callison-Burch, Chris, Trevor Cohn, and Mirella Lapata. 2008. ParaMetric: An automatic evaluation metric for paraphrasing. In *Proceedings of COLING*, pages 97–104, Manchester.

Callison-Burch, Chris, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of NAACL*, pages 17–24, New York, NY.

Callison-Burch, Chris, David Talbot, and Miles Osborne. 2004. Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proceedings of ACL*, pages 176–183, Barcelona.

Charikar, Moses. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 380–388, Montréal.

Chiang, David. 2006. An Introduction to Synchronous Grammars. Part of a tutorial given at ACL. Sydney, Australia.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:3746.

Cohn, Trevor, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34:597–614.

Cohn, Trevor and Mirella Lapata. 2007. Large margin synchronous generation and its application to sentence compression. In *Proceedings of EMNLP-CoNLL*, pages 73–82, Prague.

Corley, Courtney and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, MI.

Crouch, Carolyn J. and Bokyung Yang. 1992. Experiments in automatic statistical thesaurus construction. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 77–88, Copenhagen, Denmark.

Culicover, P. W. 1968. Paraphrase generation and information retrieval from stored text. *Mechanical Translation and Computational Linguistics*, 11(1–2):78–88.

Dagan, Ido. 2008. Invited Talk: It's time for a semantic engine! In *Proceedings of the NSF Symposium on Semantic Knowledge Discovery, Organization and Use*, pages 20–28, New York, NY.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges*, Lecture Notes in Computer Science, Volume 3944, Springer-Verlag, pages 177–190.

Das, Dipanjan and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL/IJCNLP*, pages 468–476, Singapore.

Deléger, Louise and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the ACL Workshop on Building and Using Comparable Corpora*, pages 2–10, Singapore.

Dolan, Bill and Ido Dagan, editors. 2005. *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor, MI.

Dolan, William, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING*, pages 350–356, Geneva.

Dolan, William B. and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 9–16, Jeju Island.

Dras, Mark. 1999. A Meta-level grammar: Redefining synchronous TAG for translation and paraphrase. In *Proceedings of ACL*, pages 80–88, College Park, MD.

Dras, Mark and Kazuhide Yamamoto, editors. 2005. *Proceedings of the Third International Workshop on Paraphrasing*, Jeju Island.

Duclaye, Florence, François Yvon, and Olivier Collin. 2003. Learning paraphrases to improve a question-answering system. In *Proceedings of the EACL Workshop on Natural Language Processing for*

Question-Answering, pages 35–41, Budapest.
Durbin, Richard, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.
Edmonds, Philip and Graeme Hirst. 2002. Near-synonymy and lexical choice. *Computational Linguistics*, 28(2):105–144.
Elhadad, Noemie and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Proceedings of the ACL BioNLP Workshop*, pages 49–56, Prague.
Fraser, Alexander and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.
Fujita, Atsushi, Kentaro Furihata, Kentaro Inui, Yuji Matsumoto, and Koichi Takeuchi. 2004. 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, Barcelona.
Fujita, Atsushi and Kentaro Inui. 2005. A Class-oriented approach to building a paraphrase corpus. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 25–32, Jeju Island.
Fujita, Atsushi, Shuhei Kato, Naoki Kato, and Satoshi Sato. 2007. A compositional approach toward dynamic phrasal thesaurus. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 151–158, Prague.
Fujita, Atsushi and Satoshi Sato. 2008a. A probabilistic model for measuring grammaticality and similarity of automatically generated paraphrases of predicate phrases. In *Proceedings of COLING*, pages 225–232, Manchester.
Fujita, Atsushi and Satoshi Sato. 2008b. Computing paraphrasability of syntactic variants using Web snippets. In *Proceedings of IJCNLP*, pages 537–544, Hyderabad.
Gale, William A. and Kenneth W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of ACL*, pages 177–184, Berkeley, CA.
Gardent, Claire, Marilisa Amoia, and Evelyne Jacquey. 2004. Paraphrastic grammars. In *Proceedings of the Second Workshop on Text Meaning and Interpretation*, pages 73–80, Barcelona.
Gardent, Claire and Eric Kow. 2005. Generating and selecting grammatical paraphrases. In *Proceedings of the European Workshop on Natural Language Generation*, pages 49–57, Abderdeen.
Garoufi, Konstantina. 2007. *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.
Gasperin, Caroline, P. Gamallo, A. Agustini, G. Lopes, and Vera de Lima. 2001. Using syntactic contexts for measuring word similarity. In *Proceedings of the Workshop on Knowledge Acquisition and Categorization, ESSLLI*, pages 18–23, Helsinki.
Giampiccolo, Danilo, Hoa Dang, Ido Dagan, Bill Dolan, and Bernardo Magnini, editors. 2008. *Proceedings of the Text Analysis Conference (TAC): Recognizing Textual Entailment Track*, Gaithersburg, MD.
Glickman, Oren and Ido Dagan. 2003. Identifying lexical paraphrases from a single corpus: A case study for verbs. In *Recent Advantages in Natural Language Processing (RANLP'03)*, pages 81–90, Borovets.
Grefenstette, G. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Press, Dordrecht.
Gusfield, D. 1997. *Algorithms on Strings, Trees, and Sequences: Computational Science and Computational Biology*. Cambridge University Press, Cambridge.
Hallett, Catalina and Donia Scott. 2005. Structural variation in generated health reports. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 33–40, Jeju Island.
Harris, Zellig. 1954. Distributional structure. *Word*, 10(2):3.146–162.
Hearst, Graeme. 1995. Near-synonymy and the structure of lexical knowledge. In *Working notes of the AAAI Spring Symposium on Representation and Acquisition of Lexical Knowledge*, Stanford, CA.
Hirst, Graeme. 2003. Paraphrasing paraphrased. Unpublished invited talk at the *ACL International Workshop on Paraphrasing*, Sapporo, Japan.
Hovy, Eduard H. 1988. *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.
Huang, Shudong, David Graff, and George Doddington. 2002. Multiple-translation chinese corpus. Linguistic Data Consortium. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T01`

Ibrahim, Ali, Boris Katz, and Jimmy Lin. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the International Workshop on Paraphrasing*, pages 57–64, Sapporo.

Iftene, Adrian. 2009. *Textual Entailment*. Ph.D. thesis, Faculty of Computer Science, University of Iaşi.

Inoue, Naomi. 1991. Automatic noun classification by using Japanese-English word pairs. In *Proceedings of ACL*, pages 201–208, Berkeley, CA.

Inui, Kentaro and Ulf Hermjakob, editors. 2003. *Proceedings of the Second International Workshop on Paraphrasing*. Association for Computational Linguistics, Sapporo.

Inui, Kentaro and Masaru Nogami. 2001. A paraphrase-based exploration of cohesiveness criteria. In *Proceedings of the European Workshop on Natural Language Generation (ENLG'01)*, pages 1–10, Toulouse.

Jacquemin, Christian. 1999. Syntagmatic and paradigmatic representations of term variation. In *Proceedings of ACL*, pages 341–348, College Park, MD.

João, Cordeiro, Gaël Dias, and Brazdil Pavel. 2007a. A metric for paraphrase detection. In *Proceedings of the Second International Multi-Conference on Computing in the Global Information Technology*, page 7, Guadeloupe.

João, Cordeiro, Gaël Dias, and Brazdil Pavel. 2007b. New functions for unsupervised asymmetrical paraphrase detection. *Journal of Software*, 2(4):12–23.

Jones, Rosie, Benjamin Rey, Omid Madani, and Wile Greiner. 2006. Generating query substitutions. In *Proceedings of the World Wide Web Conference*, pages 387–396, Edinburgh.

Kauchak, David and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of HLT-NAACL*, pages 455–462, New York, NY.

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

Kok, Stanley and Chris Brockett. 2010. Hitting the right paraphrases in good time. In *Proceedings of NAACL*, Los Angeles, CA.

Lin, Dekang. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of ACL-COLING*, pages 768–774, Montréal.

Lin, Dekang and Lin Pantel. 2001. DIRT - Discovery of Inference Rules from Text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328, San Francisco, CA.

Lopez, Adam. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3):1–49.

Lopez, Adam and Philip Resnik. 2006. Word-based alignment, phrase-based translation: What's the link? In *Proceedings of AMTA*, pages 90–99, Boston, MA.

Madnani, Nitin, Necip Fazil Ayan, Philip Resnik, and Bonnie J. Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 120–127, Prague.

Madnani, Nitin, Philip Resnik, Bonnie J. Dorr, and Richard Schwartz. 2008a. Applying automatically generated semantic knowledge: A case study in machine translation. In *Proceedings of the NSF Symposium on Semantic Knowledge Discovery, Organization and Use*, pages 60–61, New York, NY.

Madnani, Nitin, Philip Resnik, Bonnie J. Dorr, and Richard Schwartz. 2008b. Are multiple reference translations necessary? Investigating the value of paraphrased reference translations in parameter optimization. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 143–152, Waikiki, HI.

Malakasiotis, Prodromos. 2009. Paraphrase recognition using machine learning to combine similarity measures. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 27–35, Singapore.

Marsi, Erwin and Emiel Krahmer. 2005a. Classification of semantic relations by humans and machines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 1–6, Ann Arbor, MI.

Marsi, Erwin and Emiel Krahmer. 2005b. Explorations in sentence fusion. In *Proceedings of the European Workshop on Natural Language Generation*, pages 109–117, Aberdeen.

Max, Aurélien. 2008. Local rephrasing suggestions for supporting the work of writers. In *Proceedings of GoTAL*, pages 324–335, Gothenburg.

McKeown, Kathleen R. 1979. Paraphrasing using given and new information in a question-answer system. In *Proceedings of ACL*, pages 67–72, San Diego, CA.

Melamed, Dan. 2001. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, Cambridge, MA.

Melamed, I. Dan. 1997. A word-to-word model of translational equivalence. In *Proceedings of ACL*, pages 490–497, Madrid.

Metzler, Donald, Susan Dumais, and Christopher Meek. 2007. Similarity measures for short segments of text. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 16–27, Rome.

Mohri, Mehryar and Michael Riley. 2002. An efficient algorithm for the n-best-strings problem. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP'02)*, pages 1313–1316, Denver, CO.

Murakami, Akiko and Tetsuya Nasukawa. 2004. Term aggregation: Mining synonymous expressions using personal stylistic variations. In *Proceedings of COLING*, pages 806–812, Geneva.

NIST. 2009. NIST Open Machine Translation (MT) Evaluation. Information Access Division. `http://www.nist.gov/speech/tests/mt/`.

Nivre, Joakim, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague.

Owczarzak, Karolina, Declan Groves, Josef Van Genabith, and Andy Way. 2006. Contextual bitext-derived paraphrases in automatic MT evaluation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 86–93, New York, NY.

Pang, Bo, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of HLT-NAACL*, pages 102–109, Edmonton.

Pantel, Patrick, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning inferential selectional preferences. In *Proceedings of NAACL*, pages 564–571, Rochester, NY.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, PA.

Paşca, Marius and Péter Dienes. 2005. Aligning needles in a haystack: Paraphrase acquisition across the Web. In *Proceedings of IJCNLP*, pages 119—130, Jeju Island.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of ACL*, pages 183–190, Columbus, OH.

Power, Richard and Donia Scott. 2005. Automatic generation of large-scale paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 57–64, Jeju Island.

Quirk, Chris, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, pages 142–149, Barcelona.

Ramshaw, Lance and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Cambridge, MA.

Ravichandran, Deepak and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*, pages 41–47, Philadelphia, PA.

Ravichandran, Deepak, Patrick Pantel, and Eduard H. Hovy. 2005. Randomized algorithms and NLP: Using locality sensitive hash function for high speed noun clustering. In *Proceedings of ACL*, pages 622–629, Ann Arbor, MI.

Riezler, Stefan, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O. Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of ACL*, pages 464–471, Prague.

Romano, Lorenza, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. 2006. Investigating a generic paraphrase-based approach for relation extraction. In *Proceedings of EACL*, pages 409–416, Trento.

Rus, Vasile, Philip M. McCarthy, and Mihai C. Lintean. 2008. Paraphrase identification with lexico-syntactic graph subsumption. In *Proceedings of the 21st International FLAIRS Conference*, pages 201–206, Coconut Grove, FL.

Sahami, Mehran and Timothy D. Heilman. 2006. A Web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the World Wide Web Conference*, pages 377–386, Edinburgh.

Sekine, Satoshi. 2005. Automatic paraphrase discovery based on context and keywords between NE pairs. In *Proceedings of the International Workshop on Paraphrasing*, pages 80–87, Jeju Island, South Korea.

Sekine, Satoshi. 2006. On-demand information extraction. In *Proceedings of COLING-ACL*, pages 731–738, Sydney.

Sekine, Satoshi, Kentaro Inui, Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini, editors. 2007. *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics, Prague.

Shen, Siwei, Dragomir R. Radev, Agam Patel, and Güneş Erkan. 2006. Adding syntax to dynamic programming for aligning comparable texts for the generation of paraphrases. In *Proceedings of ACL-COLING*, pages 747–754, Sydney.

Shi, Xiaodong and Christopher C. Yang. 2007. Mining related queries from Web search engine query logs using an improved association rule mining model. *JASIST*, 58(12):1871–1883.

Shimohata, Mitsuo and Eiichiro Sumita. 2005. Acquiring synonyms from monolingual comparable texts. In *Proceedings of IJCNLP*, pages 233–244, Jeju Island.

Shinyama, Y., S. Sekine, K. Sudo, and R. Grishman. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT*, pages 313–318, San Diego, CA.

Soong, Frank K. and Eng-Fong Huang. 1990. 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, Hidden Valley, PA.

Spärck-Jones, Karen and J. I. Tait. 1984. Automatic search term variant generation. *Journal of Documentation*, 40(1):50–66.

Steedman, Mark, editor. 1996. *Surface Structure and Interpretation (Linguistic Inquiry Monograph No. 30)*. MIT Press, Cambridge, MA.

Szpektor, Idan, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proceedings of ACL*, pages 456–463, Prague.

Tanaka, Takaaki, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors. 2004. *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*. Association for Computational Linguistics, Barcelona.

Uzuner, Özlem and Boris Katz. 2005. Capturing expression using linguistic information. In *Proceedings of AAAI*, pages 1124–1129, Pittsburgh, PA.

Wallis, Peter. 1993. Information retrieval based on paraphrase. In *Proceedings of the 3rd Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 118–126, Vancouver.

Wu, Dekai. 2005. Recognizing paraphrases and textual entailment using inversion transduction grammars. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 25–30, Ann Arbor, MI.

Wu, Hua and Ming Zhou. 2003. Synonymous collocation extraction using translation information. In *Proceedings of the ACL Workshop on Multiword Expressions: Integrating Processing*, pages 120–127, Sapporo.

Zhao, Shiqi, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of ACL/AFNLP*, pages 834–842, Singapore.

Zhao, Shiqi, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. 2008. Combining multiple resources to improve SMT-based paraphrasing model. In *Proceedings of ACL-08: HLT*, pages 1021–1029, Columbus, OH.

Zhou, Liang, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating machine translation results with paraphrase support. In *Proceedings of EMNLP*, pages 77–84, Sydney.

Zhou, Liang, Chin-Yew Lin, Dragos Stefan Muntenau, and Eduard Hovy. 2006. ParaEval: Using paraphrases to evaluate summaries automatically. In *Proceedings of HLT-NAACL*, pages 447–454, New York, NY.