# Extracting Structural Paraphrases from Aligned Monolingual Corpora

**Ali Ibrahim  Boris Katz  Jimmy Lin**
MIT Artificial Intelligence Laboratory
200 Technology Square
Cambridge, MA 02139
{aibrahim,boris,jimmylin}@ai.mit.edu

## Abstract

We present an approach for automatically learning paraphrases from aligned monolingual corpora. Our algorithm works by generalizing the syntactic paths between corresponding anchors in aligned sentence pairs. Compared to previous work, structural paraphrases generated by our algorithm tend to be much longer on average, and are capable of capturing long-distance dependencies. In addition to a standalone evaluation of our paraphrases, we also describe a question answering application currently under development that could immensely benefit from automatically-learned structural paraphrases.

## 1 Introduction

The richness of human language allows people to express the same idea in many different ways; they may use different words to refer to the same entity or employ different phrases to describe the same concept. Acquisition of paraphrases, or alternative ways to convey the same information, is critical to many natural language applications. For example, an effective question answering system must be equipped to handle these variations, because it should be able to respond to differently phrased natural language questions.

While there are many resources that help systems deal with single-word synonyms, e.g., WordNet, there are few resources for multiple-word or domain-specific paraphrases. Because manually collecting paraphrases is time-consuming and impractical for large-scale applications, attention has recently focused on techniques for automatically acquiring paraphrases.

We present an unsupervised method for acquiring structural paraphrases, or fragments of syntactic trees that are roughly semantically equivalent, from aligned monolingual corpora. The structural paraphrases produced by our algorithm are similar to the S-rules advocated by Katz and Levin for question answering (1988), except that our paraphrases are automatically generated. Because there is disagreement regarding the exact definition of paraphrases (Dras, 1999), we employ that operating definition that structural paraphrases are roughly interchangeable within the specific configuration of syntactic structures that they specify.

Our approach is a synthesis of techniques developed by Barzilay and McKeown (2001) and Lin and Pantel (2001), designed to overcome the limitations of both. In addition to the evaluation of paraphrases generated by our method, we also describe a novel information retrieval system under development that is designed to take advantage of structural paraphrases.

## 2 Previous Work

There has been a rich body of research on automatically deriving paraphrases, including equating morphological and syntactic variants of technical terms (Jacquemin et al., 1997), and identifying equivalent adjective-noun phrases (Lapata, 2001). Unfortunately, both are limited in types of paraphrases that they can extract. Other researchers have explored distributional clustering of similar words (Pereira et al., 1993; Lin, 1998), but it is unclear to what extent such techniques produce paraphrases.[1]

Most relevant to this paper is the work of Barzilay and McKeown and the work of Lin and Pantel. Barzilay and McKeown (2001) extracted both single- and multiple-word paraphrases from a sentence-aligned corpus for use in multi-document summarization. They constructed an aligned corpus from multiple translations of foreign novels. From this, they co-trained a classifier that decided whether or not two phrases were paraphrases of each other based on their surrounding context. Barzilay and McKeown collected 9483 paraphrases with an average precision of $85.5\%$. However, $70.8\%$ of the paraphrases were single words. In addition, the paraphrases were required to be contiguous.

Lin and Pantel (2001) used a general text corpus to extract what they called *inference rules*, which we can take to be paraphrases. In their algorithm, rules are represented as dependency tree paths between two words. The words at the ends of a path are considered to be features of that path. For each path, they recorded the different features (words) that were associated with the path and their respective frequencies. Lin and Pantel calculated the similarity of two paths by looking at the similarity of their features. This method allowed them to extract inference rules of moderate length from general corpora. However, the technique is computationally expensive, and furthermore can give misleading results, i.e., paths having the *opposite* meaning often share similar features.

## 3 Approach

Our approach, like Barzilay and McKeown's, is built on the application of sentence-alignment techniques used in machine translation to generate paraphrases. The insight is simple: if we have pairs of sentences with the same semantic content, then the difference in lexical content can be attributed to variations in the surface form. By generalizing these differences we can automatically derive paraphrases. Barzilay and McKeown perform this learning process by only considering the local context of words and their frequencies; as a result, paraphrases must be contiguous, and in the majority of cases, are only one word long. We believe that disregarding the rich syntactic structure of language is an oversimplification, and that structural paraphrases offer several distinct advantages over lexical paraphrases. Long distance relations can be captured by syntactic trees, so that words in the paraphrases do not need to be contiguous. Use of syntactic trees also buffers against morphological variants (e.g., different inflections) and some syntactic variants (e.g., active vs. passive). Finally, because paraphrases are context-dependent, we believe that syntactic structures can encapsulate a richer context than lexical phrases.

Based on aligned monolingual corpora, our technique for extracting paraphrases builds on Lin and Pantel's insight of using dependency paths (derived from parsing) as the fundamental unit of learning and using parts of those paths as features. Based on the hypothesis that paths between identical words in aligned sentences are semantically equivalent, we can extract paraphrases by scoring the path frequency and context. Our approach addresses the limitations of both Barzilay and McKeown's and Lin and Pantel's work: using syntactic structures allows us to generate structural paraphrases, and using aligned corpora renders the process more computationally tractable. The following sections describe our approach in greater detail.

### 3.1 Corpus Alignment

Multiple English translations of foreign novels, e.g., *Twenty Thousand Leagues Under the Sea* by Jules Verne, were used for extraction of paraphrases. Although translations by different authors differ slightly in their literary interpretation of the original text, it was usually possible to find corresponding sentences that have the same semantic content. Sentence alignment was performed using the Gale and Church algorithm (1991) with the following cost function:

cost of substitution $= 1 - \frac{ncw}{anw}$
ncw: number of common words
anw: average number of words in two strings

Here is a sample from two different translations of *Twenty Thousand Leagues Under the Sea*:

[1] For example, "dog" and "cat" are recognized to be similar, but they are obviously not paraphrases of one another.

> Ned Land tried the soil with his feet, as if to take possession of it.
> Ned Land tested the soil with his foot, as if he were laying claim to it.

To test the accuracy of our alignment, we manually aligned 454 sentences from two different versions of Chapter 21 from *Twenty Thousand Leagues Under the Sea* and compared the results of our automatic alignment algorithm against the manually generated "gold standard." We obtained a precision of 0.93 and recall of 0.88, which is comparable to the numbers (P.94/R.85) reported by Barzilay and McKeown, who used a different cost function for the alignment process.

### 3.2 Parsing and Postprocessing

The sentence pairs produced by the alignment algorithm are then parsed by the Link Parser (Sleator and Temperly, 1993), a dependency-based parser developed at CMU. The resulting parse structures are post-processed to render the links more consistent: Because the Link Parser does not directly identify the subject of a passive sentence, our postprocessor takes the object of the *by*-phrase as the subject by default. For our purposes, auxiliary verbs are ignored; the postprocessor connects verbs directly to their subjects, discarding links through any auxiliary verbs. In addition, subjects and objects within relative clauses are appropriately modified so that the linkages remained consistent with subject and object linkages in the matrix clause. For sentences involving verbs that have particles, the Link Parser connects the object of the verb directly to the verb itself, attaching the particle separately. Our postprocessor modifies the link structure so that the object is connected to the particle in order to form a continuous path. Predicate adjectives are converted into an adjective-noun modification link instead of a complete verb-argument structure. Also, common nouns denoting places and people are marked by consulting WordNet.

### 3.3 Paraphrase Extraction

The paraphrase extraction process starts by finding anchors within the aligned sentence pairs. In our approach, only nouns and pronouns serve as possible anchors. The anchor words from the sentence pairs are brought into alignment and scored by a simple set of ordered heuristics:

- Exact string matches denote correspondence.
- Noun and matching pronoun (same gender and number) denote correspondence. Such a match penalizes the score by 50%.
- Unique semantic class (e.g., places and people) denotes correspondence. Such a match penalizes the score by 50%.
- Unique part of speech (i.e., the only noun pair in the sentences) denotes correspondence. Such a match penalizes the score by 50%.
- Otherwise, attempt to find correspondence by finding longest common substrings. Such a match penalizes the score by 50%.
- If a word occurs more than once in the aligned sentence pairs, all possible combinations are considered, but the score for such a corresponding anchor pair is further penalized by 50%.

For each pair of anchors, a breadth-first search is used to find the shortest path between the anchor words. The search algorithm explicitly rejects paths that contain conjunctions and punctuation. If valid paths are found between anchor pairs in both of the aligned sentences, the resulting paths are considered candidate paraphrases, with a default score of one (subjected to penalties imposed by imperfect anchor matching).

Scores of candidate paraphrases take into account two factors: the frequency of anchors with respect to a particular candidate paraphrase and the variety of different anchors from which the paraphrase was produced. The initial default score of any paraphrase is one (assuming perfect anchor matches), but for each additional occurrence the score is incremented by $\frac{1}{2}^n$, where $n$ is the number of times the current set of anchors has been seen. Therefore, the effect of seeing new sets of anchors has a big initial impact on the score, but the additional increase in score is subjected to diminishing returns as more occurrences of the same anchor are encountered.

| | count |
|---|---|
| aligned sentences | 27479 |
| parsed aligned sentences | 25292 |
| anchor pairs | 43974 |
| paraphrases | 5925 |
| unique paraphrases | 5502 |
| gathered paraphrases (score $\geq 1.0$) | 2886 |

Table 1: Summary of the paraphrase generation process

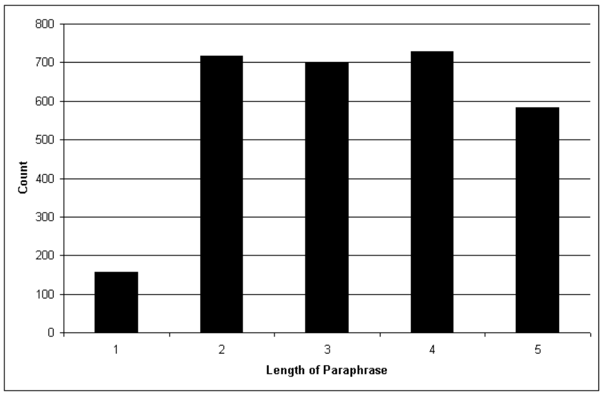


Figure 1: Distribution of paraphrase length

## 4 Results

Using the approach described in previous sections, we were able to extract nearly six thousand different paraphrases (see Table 1) from our corpus, which consisted of two translations of *20,000 Leagues Under the Sea*, two translations of *The Kreutzer Sonata*, and three translations of *Madame Bouvary*.

Our corpus was essentially the same as the one used by Barzilay and McKeown, with the exception of some short fairy tale translations that we found to be unsuitable. Due to the length of sentences (some translations were noted for their paragraph-length sentences), the Link Parser was unable to produce a parse for approximately eight percent of the sentences. Although the Link Parser is capable of producing partial linkages, accuracy deteriorated significantly as the length of the input string increased. The distribution of paraphrase length is shown in Figure 1. The length of paraphrases is measured by the number of words that it contains (discounting the anchors on both sides).

To evaluate the accuracy of our results, 130 unique paraphrases were randomly chosen to be assessed by human judges. The human assessors were specifically asked whether they thought the paraphrases were roughly interchangeable with each other, given the context of the genre. We believe that the genre constraint was important because some paraphrases captured literary or archaic uses of particular words that were not generally useful. This should not be viewed as a shortcoming of our approach, but rather an artifact of our corpus. In addition, sample sentences containing the structural paraphrases were presented as context to the judges; structural paraphrases are difficult to comprehend without this information.

| Evaluator | Precision |
|---|---|
| Evaluator 1 | 36.2% |
| Evaluator 2 | 40.0% |
| Evaluator 3 | 44.6% |
| Average | 41.2% |

Table 2: Summary of judgments by human evaluators for 130 unique paraphrases

A summary of the judgments provided by human evaluators is shown in Table 2. The average precision of our approach stands at just over forty percent; the average length of the paraphrases learned was 3.26 words long. Our results also show that judging structural paraphrases is a difficult task and inter-assessor agreement is rather low. All of the evaluators agreed on the judgments (either positive or negative) only 75.4% of the time. The average correlation constant of the judgments is only 0.66.

The highest scoring paraphrase was the equivalence of the possessive morpheme *'s* with the preposition *of*. We found it encouraging that our algorithm was able to induce this structural paraphrase, complete with co-indexed anchors on the ends of the paths, i.e., $A$'s $B \Longleftrightarrow B$ of $A$. Some other interesting examples include:[2]

$$A_1 \xleftrightarrow{*} \text{liked} \xleftrightarrow{\texttt{O}} A_2 \Longleftrightarrow$$
$$A_1 \xleftrightarrow{*} \text{fond} \xleftrightarrow{\texttt{OF}} \text{of} \xrightarrow{\texttt{J}} A_2$$

*Example:* The clerk liked Monsieur Bovary. $\Longleftrightarrow$

[2]Brief description of link labels: `S`: subject to verb; `O`: object to verb; `OF`: certain verbs to *of*; `K`: verbs to particles; `MV`: verbs to certain modifying phrases. See Link Parser documentation for full descriptions.

| Score Threshold | Avg. Precision | Avg. Length | Count |
|---|---|---|---|
| $\geq 1.0$ | 40.2% | 3.24 | 130 |
| $\geq 1.25$ | 46.0% | 2.88 | 58 |
| $\geq 1.5$ | 47.8% | 2.22 | 23 |
| $\geq 1.75$ | 38.9% | 1.67 | 12 |

Table 3: Breakdown of our evaluation results

The clerk was fond of Monsieur Bovary.

$A_1 \xleftrightarrow{\texttt{s}} \text{rush} \xrightarrow{\texttt{K}} \text{over} \xrightarrow{\texttt{MV}} \text{to} \xrightarrow{\texttt{J}} A_2 \Longleftrightarrow$
$A_1 \xleftrightarrow{\texttt{s}} \text{run} \xrightarrow{\texttt{MV}} \text{to} \xrightarrow{\texttt{J}} A_2$
*Example:* And he rushed over to his son, who had just jumped into a heap of lime to whiten his shoes. $\Longleftrightarrow$ And he ran to his son, who had just precipitated himself into a heap of lime in order to whiten his boots.

$A_1 \xleftrightarrow{\texttt{s}} \text{put} \xrightarrow{\texttt{K}} \text{on} \xrightarrow{\texttt{O}} A_2 \Longleftrightarrow$
$A_1 \xleftrightarrow{\texttt{s}} \text{wear} \xrightarrow{\texttt{O}} A_2$
*Example:* That is why he puts on his best waistcoat and risks spoiling it in the rain. $\Longleftrightarrow$ That's why he wears his new waistcoat, even in the rain!

$A_1 \xleftrightarrow{*} \text{fit} \xrightarrow{\texttt{MV}} \text{to} \xrightarrow{\texttt{I}} \text{give} \xrightarrow{\texttt{O}} A_2 \Longleftrightarrow$
$A_1 \xleftrightarrow{*} \text{appropriate} \xrightarrow{\texttt{MV}} \text{to} \xrightarrow{\texttt{I}} \text{supply} \xrightarrow{\texttt{O}} A_2$
*Example:* He thought fit, after the first few mouthfuls, to give some details as to the catastrophe. $\Longleftrightarrow$ After the first few mouthfuls he considered it appropriate to supply a few details concerning the catastrophe.

A more detailed breakdown of the evaluation results can be seen in Table 3. Increasing the threshold for generating paraphrases tends to increase their precision, up to a certain point. In general, the highest ranking structural paraphrases consisted of single word paraphrases of prepositions, e.g., *at* $\Longleftrightarrow$ *in*. Our algorithm noticed that different prepositions were often interchangeable, which is something that our human assessors disagreed widely on. Beyond a certain threshold, the accuracy of our approach actually decreases.

## 5 Discussion

An obvious first observation about our algorithm is the dependence on parse quality; bad parses lead to many bogus paraphrases. Although the parse results from the Link Parser are far from perfect, it is unclear whether other purely statistical parsers would fare any better, since they are generally trained on corpora containing a totally different genre of text. However, future work will most likely include a comparison of different parsers.

Examination of our results show that a better notion of constituency would increase the accuracy of our results. Our algorithm occasionally generates non-sensical paraphrases that cross constituent boundaries, for example, including the verb of a subordinate clause with elements from the matrix clause. Other problems arise because our current algorithm has no notion of verb phrases; it often generates near misses such as *fail* $\Longleftrightarrow$ *succeed*, neglecting to include *not* as part of the paraphrase.

However, there are problems inherent in paraphrase generation that simple knowledge of constituency alone cannot solve. Consider the following two sentences:

John made out gold at the bottom of the well.
John discovered gold near the bottom of the well.

Which structural paraphrases should we be able to extract?

*made out X at Y* $\Longleftrightarrow$ *discovered X near Y*
*made out X* $\Longleftrightarrow$ *discovered X*
*at X* $\Longleftrightarrow$ *near X*

Arguably, all three paraphrases are valid, although opinions vary more regarding the last paraphrase. What is the optimal level of structure for paraphrases? Obviously, this represents a tradeoff between specificity and accuracy, but the ability of structural paraphrases to capture long-distance relationships across large numbers of lexical items complicates the problem. Due to the sparseness of our data, our algorithm cannot make a good decision on what constituents to generalize as variables; naturally, greater amounts of data would alleviate this problem. This current inability to decide on a good "scope" for paraphrasing was a primary reason why we were unable to perform a strict evaluation of recall. Our initial attempts at generating a gold standard for estimating recall failed because human judges could not agree on the boundaries of paraphrases.

The accuracy of our structural paraphrases is highly dependent on the corpus size. As can be seen from the numbers in Table 1, paraphrases are rather sparse—nearly 93% of them are unique. Without adequate statistical evidence, validating candidate paraphrases can be very difficult. Although our data spareness problem can be alleviated simply by gathering a larger corpus, the type of parallel text our algorithm requires is rather hard to obtain, i.e., there are only so many translations of so many foreign novels. Furthermore, since our paraphrases are arguably genre-specific, different applications may require different training corpora. Similar to the work of Barzilay and Lee (2003), who have applied paraphrase generation techniques to comparable corpora consisting of different newspaper articles about the same event, we are currently attempting to solve the data sparseness problem by extending our approach to non-parallel corpora.

We believe that generating paraphrases at the structural level holds several key advantages over lexical paraphrases, from the capturing of long-distance relationships to the more accurate modeling of context. The paraphrases generated by our approach could prove to be useful in any natural language application where understanding of linguistic variations is important. In particular, we are attempting to apply our results to improve the performance of question answering system, which we will describe in the following section.

## 6 Paraphrases and Question Answering

The ultimate goal of our work on paraphrases is to enable the development of high-precision question answering system (cf. (Katz and Levin, 1988; Soubbotin and Soubbotin, 2001; Hermjakob et al., 2002)). We believe that a knowledge base of paraphrases is the key to overcoming challenges presented by the expressiveness of natural languages. Because the same semantic content can be expressed in many different ways, a question answering system must be able to cope with a variety of alternative phrasings. In particular, an answer stated in a form that differs from the form of the question presents significant problems:

**When did Colorado become a state?**
(1a) Colorado became a state in 1876.
(1b) Colorado was admitted to the Union in 1876.
**Who killed Abraham Lincoln?**
(2a) John Wilkes Booth killed Abraham Lincoln.
(2b) John Wilkes Booth ended Abraham Lincoln's life with a bullet.

In the above examples, question answering systems have little difficulty extracting answers if the answers are stated in a form directly derived from the question, e.g., (1a) and (2a); simple keyword matching techniques with primitive named-entity detection technology will suffice. However, question answering systems will have a much harder time extracting answers from sentences where they are not obviously stated, e.g., (1b) and (2b). To relate question to answers in those examples, a system would need access to rules like the following:

*X became a state in Y* $\Longleftrightarrow$
*X was admitted to the Union in Y*
*X killed Y* $\Longleftrightarrow$ *X ended Y's life*

We believe that such rules are best formulated at the syntactic level: structural paraphrases represent a good level of generality and provide much more accurate results than keyword-based approaches.

The simplest approach to overcoming the "paraphrase problem" in question answering is via keyword query expansion when searching for candidate answers:

```
(AND X became state) ⟺
   (AND X admitted Union)
(AND X killed) ⟺
   (AND X ended life)
```

The major drawback of such techniques is overgeneration of bogus answer candidates. For example, it is a well-known result that query expansion based on synonymy, hyponymy, etc. may actually degrade performance if done in an uncontrolled manner (Voorhees, 1994). Typically, keyword-based query expansion techniques sacrifice significant amounts of precision for little (if any) increase in recall.

The problems associated with keyword query expansion techniques stem from the fundamental deficiencies of "bag-of-words" approaches; in short, they simply cannot accurately model the semantic content of text, as illustrated by the following pairs of sentences and phrases that have the same word content, but dramatically different meaning:

(3a) The bird ate the snake.
(3b) The snake ate the bird.
(4a) the largest planet's volcanoes
(4b) the planet's largest volcanoes
(5a) the house by the river
(5b) the river by the house
(6a) The Germans defeated the French.
(6b) The Germans were defeated by the French.

The above examples are nearly indistinguishable in terms of lexical content, yet their meanings are vastly different. Naturally, because one text fragment might be an appropriate answer to a question while the other fragment may not be, a question answering system seeking to achieve high precision must provide mechanisms for differentiating the semantic content of the pairs.

While paraphrase techniques at the keyword-level vastly overgenerate, paraphrase techniques at the phrase-level undergenerate, that is, they are often too specific. Although paraphrase rules can easily be formulated at the string-level, e.g., using regular expression matching and substitution techniques (Soubbotin and Soubbotin, 2001; Hermjakob et al., 2002), such a treatment fails to capture important linguistic generalizations. For example, the addition of an adverb typically does not alter the validity of a paraphrase; thus, a phrase-level rule "*X killed Y*" $\Longleftrightarrow$ "*X ended Y's life*" would not be able to match an answer like "John Wilkes Booth *suddenly* ended Abraham Lincoln's life with a bullet". String-level paraphrases are also unable to handle syntactic phenomenona like passivization, which are easily captured at the syntactic level.

We believe that answering questions at level of syntactic relations, that is, matching parsed representations of questions with parsed representations of candidates, addresses the issues presented above. Syntactic relations, basically simplified versions of dependency structures derived from the Link Parser, can capture significant portions of the meaning present in text documents, while providing a flexible foundation on which to build machinery for paraphrases.

Our position is that question answering should be performed at the level of "key relations" in addition to keywords. We have begun to experiment with relations indexing and matching techniques described above using an electronic encyclopedia as the test corpus. We identified a particular set of linguistic phenomena where relation-based indexing can dramatically boost the precision of a question answering system (Katz and Lin, 2003). As an example, consider a sample output from a baseline keyword-based IR system:

**What do frogs eat?**
(R1) Alligators eat many kinds of small animals that live in or near the water, including fish, snakes, frogs, turtles, small mammals, and birds.
(R2) Some bats catch fish with their claws, and a few species eat lizards, rodents, birds, and frogs.
(R3) Bowfins eat mainly other fish, frogs, and crayfish.
(R4) Adult frogs eat mainly insects and other small animals, including earthworms, minnows, and spiders.
...
(R32) Kookaburras eat caterpillars, fish, frogs, insects, small mammals, snakes, worms, and even small birds.

Of the 32 sentences returned, only (R4) correctly answers the user query; the other results answer a different question—"What eats frogs?" A bag-of-words approach fundamentally cannot differentiate between a query in which the frog is in the subject position and a query in which the frog is in the object position. Compare this to the results produced by our relations matcher:

**What do frogs eat?**
(R4) Adult frogs eat mainly insects and other small animals, including earthworms, minnows, and spiders.

By examining subject-verb-object relations, our system can filter out irrelevant results and return only the correct responses.

We are currently working on combining this relations-indexing technology with the automatic paraphrase generation technology described earlier. For example, our approach would be capable of automatically learning a paraphrase like *X eat Y* $\Longleftrightarrow$ *Y is a prey of X*; a large collection of such paraphrases would go a long way in overcoming the brittleness associated with a relations-based indexing scheme.

## 7 Contributions

We have presented a method for automatically learning structural paraphrases from aligned monolingual corpora that overcomes the limitation of previous approaches. In addition, we have sketched how this technology can be applied to enhance the performance of a question answering system based on indexing relations. Although we have not completed a task-based evaluation, we believe that the ability to handle variations in language is key to building better question answering systems.

## 8 Acknowledgements

This research was funded by DARPA under contract number F30602-00-1-0545 and administered by the Air Force Research Laboratory.

## References

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

Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*.

Mark Dras. 1999. *Tree Adjoining Grammar and the Reluctant Paraphrasing of Text*. Ph.D. thesis, Macquarie University, Australia.

William A. Gale and Kenneth Ward Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL-1991)*.

Ulf Hermjakob, Abdessamad Echihabi, and Daniel Marcu. 2002. Natural language based reformulation resource and Web exploitation for question answering. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*.

Christian Jacquemin, Judith L. Klavans, and Evelyne Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-1997)*.

Boris Katz and Beth Levin. 1988. Exploiting lexical regularities in designing natural language systems. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING-1988)*.

Boris Katz and Jimmy Lin. 2003. Selectively using relations to improve precision in question answering. In *Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering*.

Maria Lapata. 2001. A corpus-based account of regular polysemy: The case of context-sensitive adjectives. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*.

Dekang Lin and Patrick Pantel. 2001. DIRT—discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Dekang Lin. 1998. Extracting collocations from text corpora. In *Proceedings of the First Workshop on Computational Terminology*.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-1991)*.

Daniel Sleator and Davy Temperly. 1993. Parsing English with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technology*.

Martin M. Soubbotin and Sergei M. Soubbotin. 2001. Patterns of potential answer expressions as clues to the right answers. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

Ellen M. Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1994)*.