Extracting Paraphrases from Aligned Corpora

by

Ali Ibrahim

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2002

Author . . . .
Department of Electrical Engineering and Computer Science
August 23, 2002

Certified by . . . .
Boris Katz
Principal Research Scientist
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Extracting Paraphrases from Aligned Corpora

by

Ali Ibrahim

Submitted to the Department of Electrical Engineering and Computer Science
on August 23, 2002, in partial fulfillment of the
requirements for the degree of
Master of Engineering

Abstract

Synonymy in word and expression is a problem in many natural language analysis tasks. While single word resources for synonymy exist such as thesauri and Wordnet, there are few resources for multiple word synonyms or paraphrases. We attempt to implement an unsupervised technique for automatically extracting paraphrases from aligned mono-lingual corpora. By comparing relations between similar words in two aligned sentences, we can extract paraphrases. Paraphrases which occur often and in different contexts are scored higher. While the final results were encouraging, the low density of paraphrases necessitates a much larger set of data. Such data is hard to obtain, because of the strict requirements of the alignment tool.

Thesis Supervisor: Boris Katz
Title: Principal Research Scientist

Acknowledgments

I would like to thank Boris Katz for supporting me academically, financially, and personally during my years as a UROP and while writing my thesis. I would like to thank Alton McFarland, Jimmy Lin, Sue Felshin, Luciano Castagnola, and Greg Marton for all their help during the research and writing of this thesis. The research would not have been possible without tools developed by Stefanie Tellex, Adam Oliner, and Daniel Loreto. Finally, I would like to acknowledge my parents for their continued support and love.

Contents

1 Introduction **8**

2 Background **13**

2.1 Query Expansion 13

2.1.1 Using Wordnet Senses 15

2.1.2 Using Automatically Constructed Thesauri 16

2.2 Automatic Paraphrase Extraction 16

2.2.1 Using Aligned Corpora 17

2.2.2 DIRT - Using Dependency Tree Paths as Paraphrases 19

3 Alignment **21**

3.1 Alignment in Bilingual Corpora 22

3.1.1 Gale-Church Alignment Algorithm 22

3.2 Alignment in Monolingual Corpora 24

3.2.1 Sentence Similarity Scoring 24

3.3 Evaluation of Alignment 25

4 Architecture **27**

4.1 Parsing 28

4.1.1 Link Grammar 28

4.1.2 Modifying Links 29

4.2 Named Entity Recognition 32

4.3 Paraphrase Generation 33

4.3.1 Finding Anchors 34
4.3.2 Generating Paths from Links 35
4.3.3 Path Generalization 36
4.3.4 Paraphrase Scoring 36

5 Evaluation **38**
5.1 Synonym Phrases Coverage 38
5.2 Synonym Phrases Accuracy 39

6 Future Work **41**
6.1 Improving Quality of Paraphrases 41
6.1.1 Using Different Representations for Paraphrases 41
6.1.2 Increasing the Amount of Data 42
6.2 Paraphrase Generalization 42
6.3 Deducing Syntactic and Morphological Rules 43
6.4 Evaluation of Usefulness of Synonym Phrases in IR 43

7 Conclusion **45**

List of Figures

2-1 Example of an aligned sentence 18
2-2 Dependency tree diagram 19

3-1 String alignment 23
3-2 Alignment precision for two different types of documents 25
3-3 Alignment recall for two different types of documents 26

4-1 Architecture diagram showing the 4 major stages of our system . . . 28

List of Tables

2.1 Example thesaurus entry from Schutze and Pederson's work 16

5.1 General paraphrase extraction statistics 38

5.2 Number of generated paraphrases grouped by length 39

5.3 Accuracy of generated paraphrases grouped by length 40

Chapter 1

Introduction

Information retrieval has received a lot of attention in the past few years. The popularity of search engines such as Google[1] highlights the demand for an efficient information retrieval tool. Today, a large amount of data is put into electronic format one way or another. The amount of data on the Internet is estimated to be in the terabytes, and the Internet holds a fraction of the data found on personal and commercial electronic devices. A lot of this data is unstructured. For example, free text and pictures contain a lot of useful knowledge. Free text is an especially important kind of expression since written language is used to convey many different types of information. However, human language is expressive and varied, with many different ways of representing a particular piece of information. Consequently, an ideal text information retrieval engine would be capable of retrieving multiple variations of an expression. Even though thesauri exist that contain single word variations, a manual dictionary of longer variations is difficult to create because of the large number of possible multi-word synonyms. The rest of this thesis concerns how one might go about automatically extracting multiple word synonyms from available text.

Information retrieval is concerned with retrieving the answer to a query from a large collection of documents, or *corpus*. The answer might be the entire document, a paragraph, a sentence, or a word. Often *document retrieval* refers to retrieving documents and *question answering* refers to retrieving a sentence or word. Informa-

[1] http://www.google.com

tion retrieval performance is usually measured along two different dimensions: recall and precision. Recall is the percentage of correct answers retrieved in response to a query from the total number of correct answers available in the corpus. Precision is the percentage of correct answers retrieved in response to a query from the total number of retrieved answers. Some researchers use a formula based on the precision and recall to come up with a single metric for performance. A common such formula is the F-measure (β is a parameter representing how important precision is relative to recall).

$F_{measure} = \frac{(\beta^2+1.0)*P*R}{(\beta^2*P)+R}$

Most information retrieval work has focused on free-text documents. While many natural language systems exist that can effectively parse short restricted pieces of text, there are few that are fast and general enough to deal with all free text. Frequently free text contains sentence fragments, speech, and obscure terminology. The simple solution is to use the "bag of words" approach. Free text is represented as a simple list of words. Answers are retrieved by finding documents which satisfy a boolean query. Early Internet information retrieval engines used this simple approach. In fact, most of today's Internet engines use this approach accompanied by sophisticated ranking algorithms (such as Google's evaluation of a page's referrers). The "bag of words" approach's performance can be improved by using some common information retrieval methods. Term weighting using TF[2] and IDF,[3] elimination of stop words (words with very high frequency), and inflectional and derivational stemming can help improve baseline performance. More complicated models for information retrieval have been developed, but use the same principle of using individual words as the basis for retrieval.

INQUERY [24] developed at Umass uses a probabilistic model as its foundation for information retrieval. Queries and documents are represented using Bayesian nets. Query and document nets can then be connected and evaluated to come up with a

[2]term frequency: the frequency of the occurrence of the term with respect to all occurrences of all terms

[3]inverse document frequency: the frequency of the term in unique documents with respect to all occurrences of that term

belief in a document depending on the query. The first version of INQUERY used many standard information retrieval techniques including named entity recognition, stop words, and stemming. The idea of Bayesian nets is interesting, because one can combine many different information retrieval techniques into one Bayesian net.

SMART [4], developed by Gerald Salton, uses a vector space model to represent documents and queries. Documents and queries are represented by vectors whose features are individual words. SMART eliminates stop words and weights the vector elements according to TF/IDF. Documents are scored against queries by looking at the similarity of the query and document vectors. A common similarity function is to calculate the cosine of the angle between the two vectors.

More complicated approaches exist which are variations on the document vector space model, such as LSI [4] which uses SVD (singular value decomposition) to reduce the dimensionality of the document and query vectors. However, basic information retrieval algorithms have remained the same over the past few decades.

The systems we have talked about so far focus on word-based approaches. Full-scale parsing is not usually considered for information retrieval for several reasons. First, researchers have found that parse trees often are too specific to be good features of documents. Second, even modern parsers have a difficult time correctly parsing sentence fragments, quotes, and long sentences from general domain corpora. Finally, it is not currently computationally feasible to parse large corpora. Naturally there are many systems that try to fall between word based approaches and full syntactic parsing.

Sapere [13] developed by Jimmy Lin at MIT uses Minipar[10] to produce dependency trees for sentences and then uses rules to extract simple semantic relationships such as possessive relations, subject-verb relations, verb-object relations, adjective-noun relations, and adverb-verb relations. Relations are represented by ternary expressions (triples such as [subject verb object]). There are several advantages of indexing relations over using a word-based approach. One is that relations are directional, meaning we can capture the difference between “x lent money to y” and “y lent money to x”. In addition, relations allow us to retrieve more easily a few words in response to a query by identifying the appropriate relation. This is useful, because finding a specific answer is important in question answering. Unfortunately, Sapere's performance is limited by the parser it uses, which while fast, is not comparable to the performance of a simple word-based indexing approach.

Current information retrieval technologies have many shortcomings. Anaphora resolution,[4] understanding hypernymy and hyponymy, and recognizing named entities are a few of the problems facing information retrieval. Many issues found in general artificial intelligence systems, such as logical deduction, and common sense are also present in information retrieval. The rest of this thesis discusses a system that aims to provide assistance in solving two common issues in information retrieval: polysemy and synonymy.

Polysemy occurs when words have several meanings, creating ambiguity in a document or query. This phenomena is quite common in English. For example, "light" can mean the opposite of "heavy" or can refer to illumination. Resolving the meaning of ambiguous words can help improve the precision of information retrieval engines.

Synonymy is also quite common in English. For example, "tiny" and "small" are usually considered synonyms. One definition of synonymy is that two words or phrases are synonyms if they are interchangeable. However, because many words are polysemous, this definition is restrictive because synonyms are assumed to be context independent. We can modify this definition so that synonyms are interchangeable in a specific context. While few words have the exact same meaning, connotation, and usage, it is still important to recognize instances where words are interchangeable or close in meaning. If we can accurately tell what words are interchangeable in queries and documents, we should be able to improve recall in information retrieval engines. Synonymy and polysemy are related very closely as will be discussed in the next section.

This thesis seeks to address the lack of thesauri for common multiple word phrases by combining existing work on extracting paraphrases from free text. While there are many repositories for single word synonyms, there are very few sources of multiple

[4] resolving references

word synonyms. Multiple words synonyms can be thought of as paraphrases since they represent different ways of expressing an idea in a sentence. Multiple word synonyms have advantages and disadvantages over single word synonyms. The domain of multiple word synonymy is exponentially larger than that of single word synonyms; however, polysemy is less of an issue when using longer synonyms.

The paraphrase generation system extracts synonyms or paraphrases by examining different phrasings of the same semantic idea and extracting the corresponding dependency tree path fragments. These matching path fragments represent a paraphrase which is evaluated according to a scoring function. We envision use of such paraphrases in many natural language applications, however, the primary goal is to produce paraphrases useful in information retrieval.

This thesis' approach combines ideas from Dekang Lin [12] and Regina Barzilay [3]. Lin's path representation and basic premise is combined with Barzilay's idea of using aligned mono-lingual corpora. In this way, the computational complexity and antonymy problems in Lin's approach can be mediated by using a more restricted data set. On the other hand, Lin's path representation is more sophisticated than Barzilay's use of lists of words as contexts. In addition, Lin's approach allows for extraction of longer paraphrases.

Chapter 2

Background

Polysemy and synonymy have long been recognized as challenges in information retrieval. Researchers have explored using synonyms from different sources and indexing senses (particular word meaning) of individual words. Polysemy and synonymy are related problems in several ways. Words often have context dependent synonyms. Going back to our previous example of polysemy, "light" can have as a synonym "illumination", but not when it is used as the opposite of "heavy". If we add to a query synonyms of a word sense that is not contained in a sentence, then the precision of the information retrieval engine will suffer because we will retrieve incorrect documents. Therefore, the simple approach of including synonyms in a query (see below) does not necessarily improve information retrieval performance.

2.1 Query Expansion

Query expansion is the addition of terms to the query terms to improve information retrieval performance. We have previously discussed the need for the information retrieval engines to deal with synonymy in languages. Query expansion using synonyms is one solution to this problem. However, words often have many synonyms, many of which can be incompatible. If one were to expand all possible synonyms, the query would contain many undesirable words resulting in a loss of precision and speed for the information retrieval engine. To choose the correct synonyms to include in a query, one must determine the sense of the word in a particular context. This is the problem of word sense disambiguation (WSD). Word sense disambiguation is well-researched topic especially with regards to information retrieval. One's intuition is that word sense disambiguation would lead to an increase in information retrieval performance; however, this is not necessarily the case. First of all, query terms naturally disambiguate themselves. For example the query "shot the ball" contains the word "shot" which has several meanings, yet it is likely that if a document matches "shot" and "ball" in close proximity, then the correct meaning of "shot" (action in sports) will be represented. This fact reduces the potential benefit of word sense disambiguation. Secondly, a mistake in word sense disambiguation can be very costly resulting in the retrieval engine retrieving many incorrect documents.

Sanderson [18] conducted experiments to see if word sense disambiguation alone improves information retrieval performance. He used pseudo words, a mechanism introduced by Yarowsky [25]. Pseudo words are constructs such as "banana-rifle" whose actual meaning is one of the component words. By substituting pseudo words for one of their component words, Sanderson could test the effectiveness of word sense disambiguation without the cost of manually sense-tagging a corpus. The test corpus was chosen to be Reuters which does not have a set of standard queries. Consequently, Sanderson partitioned the documents by Reuters subject code and derived queries by examining the common words in a document. The query is assumed to match other documents with the same subject code. Sanderson then created a word sense disambiguator whose accuracy can be controlled. He found that a 90% accuracy word sense disambiguation is needed to achieve the same performance as without word sense disambiguation. The results of Sanderson have been criticized on several points (although most researchers agree a high degree of accuracy WSD algorithm is needed to be effective). One is that the use of subject codes as a determiner of similarity of documents is less than optimal. Secondly, the distribution of senses of pseudo-words is not necessarily representative of natural polysemy [19]. Finally, word senses are often similar, so picking the incorrect sense is less damaging than one might think from general pseudo words.

The effectiveness of using query expansion with word sense disambiguation in information retrieval has been hotly debated. Voorhees [22] conducted an experiment using Wordnet senses with the TREC-2 queries and corpus.[1] Wordnet was developed at Princeton University in 1993 [15]. A semantic net, Wordnet uses synonym sets as its basic unit of semantic meaning. Word forms can be contained in several synonym sets, or *synsets*. Phrases are not included as part of synsets. Wordnet contains a large collection of semantic relations between synsets including antonymy, hyponymy, hypernymy, and meronymy. Using a information retrieval engine similar to SMART, Voorhees manually added Wordnet senses and related senses (synonyms, hypernyms, hyponyms) to the test queries. She used human judgment to determine the word sense and whether there should be any extra synsets in the query. A set of tests were run with various levels of expansion for the above relations in Wordnet. The results are striking in that she showed there is little or no improvement with human word sense disambiguation accuracy with any of the different levels of expansion. However, she pointed out that since TREC-2's topic descriptions are fairly complete, maybe the query expansion's goal of providing better recall was disadvantaged. Voorhees also mentioned that Wordnet's incompleteness resulted in queries being weighted in favor of those terms which are contained in Wordnet.

2.1.1 Using Wordnet Senses

Mihalcea and Moldovan [14] expanded queries using Wordnet synset codes. To disambiguate a word form, they looked at its relationship through Wordnet with other word forms in the sentence. By doing so, they allowed words to disambiguate each other. They found that their disambiguation algorithm disambiguates 55% of words with 92.22% accuracy. Their information engine indexes dates, numbers, locations, titles, stemmed words, and synsets when available. Using the *Cranfield* collection, they found an increase in precision of 4% and an increase in recall of 16% using synset query expansion.

[1] TREC: Text REtrieval Conference

Word	Synonyms
accident	repair faulty personnel accidents exhaust equipped mishaps injuries sites

Table 2.1: Example thesaurus entry from Schutze and Pederson's work

2.1.2 Using Automatically Constructed Thesauri

Schutze and Pederson [19] constructed a thesaurus based on word co-occurrence. Words that appear together within 41 words of each other are said to co-occur. The co-occurrences were represented by a matrix where each entry represents the number of co-occurrences of the word for that column with the word for that row. Similarity of words was determined by the similarity of their corresponding columns. Obviously for a large number of words, such a matrix and the subsequent calculations would be computationally expensive. Schutze and Pederson used SVD to reduce the dimensionality of the matrix. The end result is a thesaurus (See Table 2.1) where similarity was measured by second order co-occurrence, i.e., the sharing of neighbors.

Schutze and Pederson used a word sense disambiguation algorithm which used the context of the sentence. However, they automatically determined possible senses of a word by clustering the context vectors retrieved from a corpus and creating a sense vector which represents each cluster. In this way, they hoped to avoid the coverage problem of Wordnet. A word sense is chosen simply by finding the "closest" (using a measure of similarity such as the cosine of the vectors) sense vector generated by the clustering algorithm. They evaluated their word sense disambiguation algorithm and thesaurus using the TREC-1 corpus. Allowing for multiple senses of a word to be used in the query they achieved an improvement of 14% over the baseline vector similarity model.

2.2 Automatic Paraphrase Extraction

Almost all of the information retrieval strategies for dealing with synonymy and polysemy use single word synonyms. Since single words often have many different meanings, word sense disambiguation is very important. However, we believe that to capture a larger range of linguistic phenomena, we must consider multiple word paraphrases. In some instances, multiple word paraphrases can be simply a single word synonym with the rest of the paraphrase acting as sense marker. For example, in the paraphrase ("enter", "pass through"), "through" in "pass through" serves to eliminate the sense of "pass" used in "pass the ball". However, sometimes multiple word paraphrases are beyond the scope of single word synonyms. In the paraphrase ("play", "have fun"), we need the phrase instead of single word to construct the paraphrase (this paraphrase is closer to an inference rule).

Multiple word paraphrase can be useful in many ways. Barzilay [3] uses paraphrases to help eliminate redundancies in sentences constructed by summarizing several other sentences. Lin [12] extracts paraphrases as a way of extracting inference rules.

2.2.1 Using Aligned Corpora

Regina Barzilay published a paper in 2001 [3] which is one of the bases of this thesis. Barzilay develops an automatic paraphrase extraction tool to aid in multi-document summarization; however, she notes the obvious uses in other natural language systems. Specifically, Barzilay uses the paraphrases to recognize redundant phrases in different documents. Paraphrases are either lexical or morpho-syntactic.

> Lexical:
> $sky <==> heavens$
>
> Morpho-syntactic:
> $DT_1 JJ NN_2 <==> DT_1 NN_2$
>
> The letters represent parts of speech and part of speech markers with the same subscript must refer to the same word.

Barzilay uses multiple translations of several foreign novels into English as the data for her paraphrase extraction tool. First, she aligns the translations at the sentence

People said "The Evening Noise is sounding, the sun is setting."
"The evening bell is ringing," people used to say.

Figure 2-1: Example of an aligned sentence

level using the Gale and Church alignment algorithm [6] with a weight function based on the number of common words. The result is a set of semantically equivalent sentences (Example in Figure 2-1). Using a sample of 127 aligned sentences, 120 (94.5%) were judged to be correct alignments.

The premise of her algorithm is that phrases occurring in the same context in semantically equivalent sentences, are paraphrases. This premise allows both single and multiple word paraphrases to be extracted. Contexts are a combination of the words occurring to the left and right of a possible paraphrase in a pair of aligned sentences. Barzilay uses ordered words as contexts instead of a more sophisticated mechanism such as parse tree fragments because it was difficult to find a broad coverage parser for literary novels. The paraphrase extraction tool uses a co-training algorithm to train a tester which can decide whether two phrases are paraphrases. First the algorithm learns the strength of positive and negative contexts using examples of positive and negative paraphrases. Paraphrase strength is then determined by looking at the percentage occurrence in positive contexts with respect to the total number of occurrences. Paraphrases whose strength is greater than a certain threshold are added to the list of known paraphrases. These two steps constitute one iteration, with each iteration adding to the number of paraphrases and refining the evaluation of each context. The co-training algorithm is seeded with an initial set of paraphrases of words or phrases occurring in both sentences. Since the sentences are assumed to be equivalent, it is reasonable to expect that equivalent words in the two sentences refer to the same idea. The algorithm terminates after reaching a certain number of paraphrases or after executing a certain number of iterations.

Using 11 translations of various foreign novels such as *Madame Bovary* and *20000 leagues under the sea*, Barzilay extracts 9483 lexical paraphrases and 25 morpho-syntactic paraphrases. Human judges checked a random selection of 500 paraphrases and gave precisions of 87.8% and 85.2%. Recall is very hard to judge, but Barzilay attempts to take a small sample of aligned sentences and have humans extract all the paraphrases. She finds about 69% coverage.

2.2.2 DIRT - Using Dependency Tree Paths as Paraphrases

Dekang Lin's work constitutes the second basis for this thesis. DIRT [12] finds pairs of dependency tree paths that are similar. Lin calls these pairs of paths *inference rules*. Dependency trees are sets of relations where each word may modify one other word (each relation consists of a head word and a modifier). Each word may be modified by several words. This set of relations can be represented as a tree as shown in Figure 2-2.

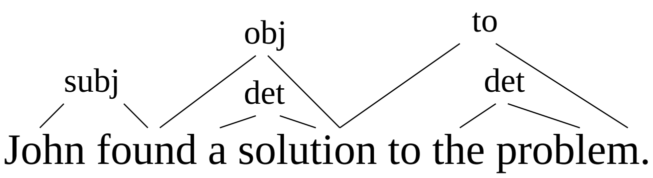

Figure 2-2: Dependency tree diagram

Lin's definition of inference rules is broader than that of paraphrases. Inference rules can be logical deductions of a phrase or vice-versa.

The central hypothesis of Lin's work is a variation of Harris's distributional hypothesis, which states that words appearing in similar contexts are similar. The hypothesis is extended to include that paths appearing in similar contexts are similar. The context for a dependency tree path is the end words of the path, which are called the SlotA and SlotB fillers. The similarity of paths is determined by the similarity of their slot fillers. Slot fillers are weighted according to their mutual information with the path. Lin only considers paths that have noun slot fillers, have lengths in a certain range, and occur with some frequency. This limits the number of paths considered and alleviates the computational complexity of considering every path.

Lin evaluates paraphrases by examining the usefulness of the paraphrases for the first 6 questions in TREC-8. He finds that the paraphrases cover a much broader range of phenomena than paraphrases generated by humans. There are two main problems in Lin's approach. First, the computational complexity is very high, because the similarity of every combination of paths must be calculated. Since the computational complexity is high and the amount of data needed for quality inference rules is high, the algorithm is very time consuming. Secondly, antonym, hypernym, and hyponym paths tend to occur in the same contexts; therefore, they are hard to distinguish from valid inference rules. For example, the inference rule (x lent money to y <==> x borrowed money from y) is not a valid inference rule; in fact they could be considered antonym paths. Unfortunately, these two paths will have similar x's and y's, namely people or companies.

Chapter 3

Alignment

Sentence alignment is the pairing of sentences in two different documents such that each pair of sentences represent the same semantic idea. For example, the following two sentences express roughly the same idea.[1]

```
At the period when these events took place, I had just returned from a
scientific research in the disagreeable territory of Nebraska, in the
United States.
```

```
During the period in which these developments were occurring, I had
returned from a scientific undertaking organized to explore the
Nebraska badlands in the United States.
```

Alignment has been mostly used in the context of parallel bilingual corpora in machine translation. Machine translation is the problem of automatically translating documents to a different language. Machine translation research uses alignment to pair up sentences between documents which are translations of each other. The pairs of sentences can then be used to learn the rules needed to translate from one language to another. In the case of machine translation, the documents aligned are strictly translated; each document contains roughly the same order of semantic ideas (and thus sentences to be matched). Consequently, alignment in the machine translation

[1]sentences taken from chapter 2 of *20000 Leagues under the Sea*

context is compared to a string alignment task, explained below. For the purposes of this thesis, alignment will also be used to describe pairing of strings without assuming any ordering of semantics in the source documents.

There is little previous work in using alignment for monolingual corpora, i.e, alignment between documents in the same language. Barzilay [3] uses alignment in the generation of paraphrases. A rough analogue of alignment is found in multi-document summarization where systems often try to extract fragments of documents related to a common theme. Each fragment is related to the common theme and thus to the other fragments. However, being related to a common theme implies a much looser definition of similarity than that of a representing the same semantic idea.

3.1 Alignment in Bilingual Corpora

Many machine translation algorithms are based on learning a translation model from existing data. There is a large number of existing translations of documents from one language to another. This data can be used by statistical machine translation systems to learn a translation model whether on the word or phrase level [16]. Initially we are given aligned documents; therefore, alignment on the sentence level improves the granularity of the translation alignment. Machine translation algorithms benefit from having smaller aligned units, creating a demand for accurate sentence alignment algorithms.

3.1.1 Gale-Church Alignment Algorithm

Gale and Church [6] describe a simple alignment algorithm to align parallel bilingual corpora. The algorithm is a variation of string alignment, in which two strings are aligned to maximize the match between them. Figure 3-1 shows a sample string alignment problem and solution.

The unit of matching in basic string alignment is a character. Characters can match equivalent characters. Gale and Church use the basic principles of string alignment, but rather than use characters they use strings (either sentences or paraabcdef, ahgbcief

a			b	c	d		e	f
a	h	g	b	c		i	e	f

Figure 3-1: String alignment

graphs) as the unit of alignment. Since strings which are translated do not match up exactly, they evaluate the match probability between strings. By looking at hand aligned data, they come up with model for determining the probability that two sentences are matches based on the similarity of their lengths. Surprisingly, this model is quite powerful for bilingual alignment.

The Gale and Church algorithm can delete sentences or substitute sentences. They use a dynamic programming algorithm $O(n^2)$ where n is the maximum of the number of strings in the two documents. At each step, the algorithm maximizes the probability of the match of the first i strings in document one and the first j strings in document two. The match probability is calculated by maximizing a match operation (deletion or substitution) plus the match probability of subproblem that the operation originated from. The cost of deletion is determined by constants or the length of the deleted string.

Gale and Church make a few other changes to the basic string alignment algorithm. They allow for two consecutive sentences to be treated as a unit and compared with another sentence (contraction). When calculating the match probability for combined sentences a penalty is assessed to show that is more likely sentences match one to one. The contraction operation is useful, because often times translators will split up or combine sentences while translating.

Gale and Church test their alignment algorithm with a set of 15 trilingual reports issued by the *Union Bank of Switzerland*. A judge aligned by hand the total 725 sentences and his alignment was checked by two additional judges. After determining that the primary judge's alignments could be used as a standard, Gale and Church find that the alignments produced by their algorithm were 95.8% correct.

3.2 Alignment in Monolingual Corpora

Alignment in monolingual corpora can use the same algorithm described by Gale and Church; however, it seems logical to exploit the fact that documents use the same language. The Gale and Church algorithm can be easily modified to use a different function to calculate the match probability. Alignment of the data in this thesis, which are multiple translations of a foreign novel, is complicated by two factors. First, the translations have more literary license than, for example, translations of official documents such as the *Canadian Hansards*.[2] Secondly, the parallelism is not direct, i.e., the texts are not derived from each other, but rather are derived from a common source, which adds a little more noise than found in common data sets used in bilingual machine translation such as the *Canadian Hansards*.

3.2.1 Sentence Similarity Scoring

Since both source documents are written in the same language, the Gale and Church match probability function can be improved. A simple extension is to factor in the number of common words between the two strings. Barzilay uses the same basic idea using an unspecified function. We come up with the following function used by our alignment program:

cost of substitution $= 1 - \frac{ncw}{anw}$

ncw: number of common words

anw: average number of words in two strings

This similarity scoring function seems to do well in experimental tests. For a match to occur, the substitution cost must be less than the cost of deleting both sentences. As a result, the performance of the alignment program is heavily dependent on the cost given to the delete operation. However, a lower deletion cost also causes some correct alignments to be ignored. The data used in this thesis was produced using a delete cost of 0.6.

[2] Canadian government transcripts written in both English and French

3.3 Evaluation of Alignment

Alignment is an important part of the paraphrase generation system. Since the system relies on the premise that sentences contain similar semantic ideas, the precision of the alignment directly affects the precision of the resulting paraphrases. In addition, the recall (percentage of available alignments extracted) determines the amount of data available for the paraphrase generation system. The alignment was tested on two very different sets of documents. The first set of documents is a pair of translations of one chapter from *20000 leagues under the Sea*. The second is a pair of articles from the Meter Corpus which are translations of a common Associated Press article. After some initial testing with alignment using the Meter Corpus, it was decided that the alignment of that type of data was not sufficiently accurate to be included as data for the paraphrase generation. The standard alignment for these sets of documents was produced by human judgment. Because there are no concrete measures of a correct alignment, this must suffice. Barzilay's reported results are included to measure the effectiveness of alignment implementation used in this thesis.

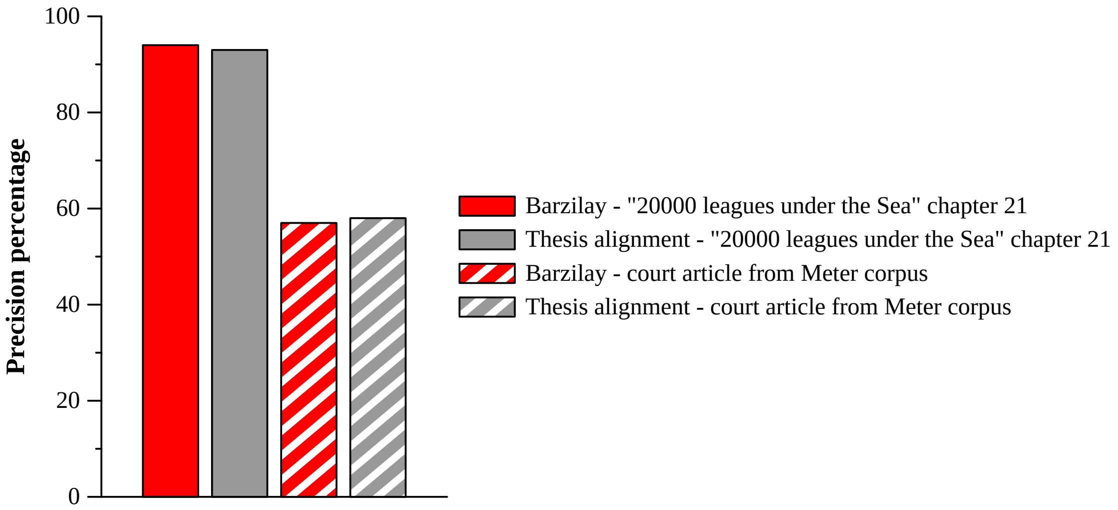

Figure 3-2: Alignment precision for two different types of documents

As seen in Figures 3-2 and 3-3, the alignment algorithm used for this thesis has similar performance to that of Barzilay's. The precision is quite good for the direct translations of the foreign novel, but is poor for the more out of order Meter articles.

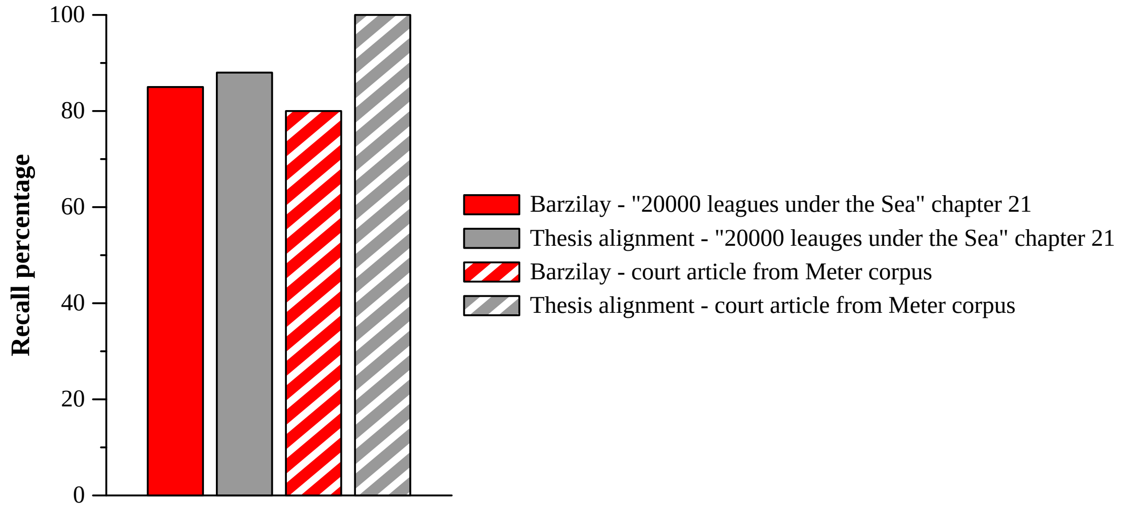

Figure 3-3: Alignment recall for two different types of documents

We can conclude that the alignment algorithm is well-suited for strict translations, but not as effective for out of order parallel corpora such as the Meter corpus. The effect of alignment performance on the overall performance of the system is an open issue and should be addressed later on. It is possible that incorrect alignments will not degrade the overall system performance by a large amount. For example, the number of anchors (discussed later) will be smaller in an incorrect alignment. In addition, an incorrect paraphrase from an incorrect alignment will not likely be repeated again, so its frequency should be low.

Chapter 4

Architecture

The crux of this thesis is the use of aligned corpora within DIRT's framework. The hypothesis in DIRT is constrained to aligned sentences and paths between equivalent words called anchors.

The computational complexity issues in DIRT are greatly mitigated because we need only compare two sentences at a time. In addition, the hypothesis is stronger with respect to similarity than DIRT's hypothesis, because not only are the slots of the paths equivalent, but the sentences the two paths are contained in are semantically similar. However, the hypothesis also narrows the available data sets, because we must depend on aligned corpora.

There are several other notable differences between Lin's work and our system. Lin uses Minipar [12], while our system uses the link parser (see Section 4.1.1). In addition, most of the restrictions on the paths that Lin examined are removed, since computational complexity is not an issue anymore.

The system acts in stages (see Figure 4-1) with the first stage being the alignment of parallel corpora. The sentences in the aligned corpora are then parsed using the link parser in the second stage. The third stage consists of modifying the link parser output to improve the quality of the links in various ways. The final stage extracts anchors from the aligned sentences, matches the paths between them, and tabulates the frequency of the paraphrases. Each stage produces XML output that can be read by the next stage. XML is used as the intermediary format because it is human read-able and many XML parsers exist for different programming languages. XML's lack of conciseness is not an issue because the parallel corpora are relatively small. Most of the stages are implemented in Java with some Perl mixed in. Since performance is not an issue, Java provided an ideal high level language to work with, while Perl helped implement some simple text processing tools.

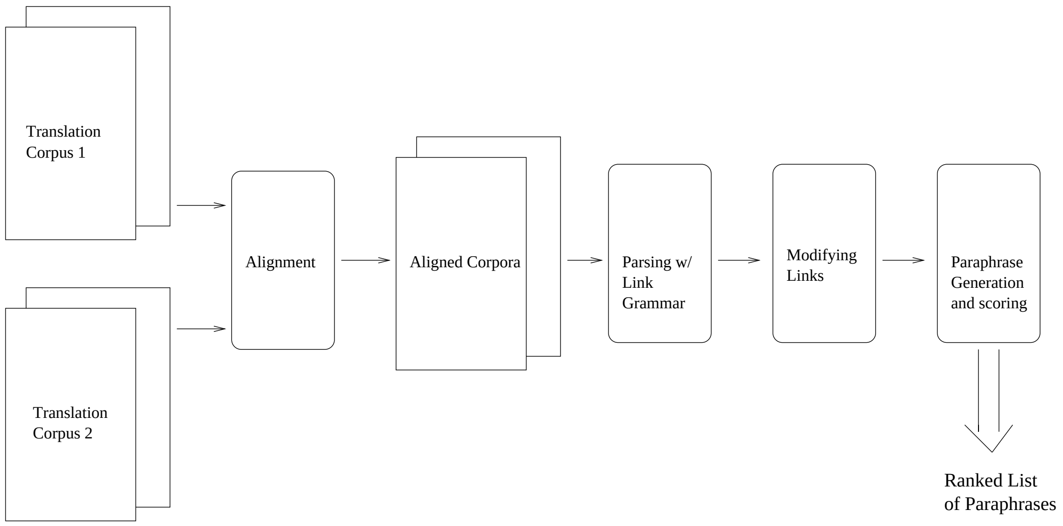

Figure 4-1: Architecture diagram showing the 4 major stages of our system

While the system could have been implemented as one large program with all the stages intertwined, the pipeline format allowed for a shorter development cycle. The parsing stage is the main bottleneck; therefore, it was worthwhile to separate it from the other tasks. This allowed quicker debugging of the heart of the system: construction and evaluation of paraphrases.

4.1 Parsing

4.1.1 Link Grammar

Link Grammar [20] was developed by Sleator and Temperley at CMU. Link Grammar is a grammatical system which produces a set of links between words in a sentence. Sleator shows that the basic functionality of a link grammar is in fact a context-free grammar; however, a postprocessor that filters possible linkages to deal with some linguistic phenomena is not context-free. The links produced follow 3 basic rules:

1. The links satisfy the local requirements of each word.
2. The links do not cross. (Planarity)
3. Every word is connected. (Completeness)

The grammar is defined by the linking requirements for each word. For example, "a" requires one end of a determiner link.[1] Linking rules can be defined for classes of words, allowing for simple addition of new words. Morphological heuristics are used to guess at the part of speech of words that have not been seen before.

Example Link Grammar linkage:

```
             +---Js---+
 +--Ss-+-MVp+  +--Ds-+
 |     |    |  |     |
John went.v to the park.n
```

The links give a straightforward way to calculate the paths between words. Even though there are parsers which are more sophisticated, link grammar is a good choice because of two qualities: broad coverage and partial parsing (producing partial linkages). Broad coverage and partial parsing are especially important due to the demanding nature of the literary corpora used by the system.

4.1.2 Modifying Links

Link Grammar provides a simple dependency parser which is used as the basis for the paraphrase paths; however, it is not optimal for providing terse dependency rep-

[1] A short summary of link types can be found at http://www.link.cs.cmu.edu/link/dict/summarize-links.html

resentations. For example, Link Grammar is constrained to link every word, even words which are not usually desirable in paraphrases such as auxiliary verbs and conjunctions. In addition, Link Grammar's link representation contains syntactic information, such as the subtype of a relation, which we ignore to generalize our paraphrases.

To provide a framework for modifying the link parser linkages, a tool was written to allow the linkages to be changed according to a rule based system. The rule based system was based on Jess, a Java expert system, and rules were written to modify the link parser linkages. These rules did not attempt to satisfy the link parser requirement of planarity.[2]

Classes of rules:

1. Auxiliary Verbs

   The paraphrases constructed are assumed to be tense independent. While this is not always true,[3] it is generally a good assumption. With that in mind, auxiliary and modal verb links are bypassed.

   ```
     +--Ss-+--If--+--PP-+--Os-+
     |     |      |     |     |
   John must.v have.v loved Mary
   ```

   is converted to

   ```
     +-------Ss---------+--Os-+
     |                  |     |
   John must.v have.v loved Mary
   ```

2. Passives

   The link parser does not correctly indicate the subject and object when it encounters a passive structure. However, the link parser does recognize the passive

[2] planarity means the links do not cross

[3] Reducing the paraphrase (John had arrived when Mary arrived, John arrived before Mary) to (John arrived when Mary arrived, John arrived before Mary) is not correct

verb structure through the relationship between the auxiliary verb and primary verb. We would like for the relationships between the subject and object to be correctly identified in a passive structure, so we relink the sentence when a passive construction is observed. The object is taken to be the original subject. The subject is allowed to be missing or contained in an agent construction such as “by John Doe”.

```
              +--------MVp--------+
              |     +---Js---+    |
  +-Ss-+--Pv--+-MVp-+  +--Ds-+    +-Js+
  |    |      |     |  |     |    |   |
John was.v taken.v to the mall.n by Mary
```

is converted to

```
              +--------S-------------+
              |     +---Js---+       |
  +-O---------+-MVp-+  +--Ds-+       +
  |           |     |  |     |       |
John was.v taken.v to the mall.n by Mary
```

3. Relative Clauses

   The link parser differentiates between structures with relative clauses and those without by including relative links. We wish to bypass these links so as to make the representation of subject and object relations consistent throughout all sentences.

```
       +---------Ss--------+
       +------Bs-----+     |
 +--Ds-+---R--+--RS--+     +--Pa-+
 |     |      |      |     |     |
the boat.n that.r sank.v was.v red.a
```

is converted to

```
       +---------Ss--------+
       +------S------+     |
 +--Ds-+             |     +--Pa-+
 |     |             |     |     |
the boat.n that.r sank.v was.v red.a
```

4.2 Named Entity Recognition

The system recognizes 3 categories of named entities: proper nouns, people, and places.

Proper nouns are recognized by observing the link parser output. The link parser connects proper nouns with the following links G (connects proper nouns), DG (connects proper nouns with determiners), JG(connects prepositions to proper nouns), and MG (allows prepositions to modify proper nouns). Nouns that are connected with G links are combined into one proper noun (Currently DG, JG, or MG links are not modified).

Example linkage with a proper noun:

```
                                      +------Jp------+
                                      |   +----Dsu---+
    +----G---+--Ss--+-Pg*b-+--MVp--+   |   +---A--+
    |        |      |      |       |   |   |      |
Hollywood Studios is.v casting.v for.p a new.a film.n
```

is converted to

```
                                      +------Jp------+
                                      |   +----Dsu---+
    +--------Ss-----+-Pg*b-+--MVp--+   |   +---A--+
    |               |      |       |   |   |      |
Hollywood_Studios is.v casting.v for.p a new.a film.n
```

People and places are recognized by simply looking them up in a list extracted from Wordnet (see Section 2.1). For example, words whose every synset had "person" as a hypernym are recognized as people words (fireman, mayor, etc. . .). Furthermore, a list of male and female names compiled by hand is used to determine gender if possible. Named Entity Recognition is used in two different ways. First, the named entities are used to help infer anchors (discussed in the next section). Second, named entity recognition helps us determine the part of speech of some words, which the link parser cannot recognize. Finally, the named entities' categories are used to substitute nouns in the paraphrase paths when possible. The rationale behind substituting named entities in the paraphrases is that it is likely that paraphrases do not depend on a specific name or place (exceptions to this include idioms and sayings).

4.3 Paraphrase Generation

Following the algorithm in Lin's work [12], the first step in paraphrase generation is to find the anchors, which is more complicated in the context of aligned data, because we want to find matching anchors in each pair of aligned sentences. Next, the anchors are used to generate the paraphrases. The paraphrases are then reduced and scored individually and with respect to the other occurrences of the paraphrases in the corpora.

4.3.1 Finding Anchors

Anchors are the basis for generating paraphrases. An anchor is a pair of words in opposite aligned sentences that refer to the same semantic idea. Here, we restrict anchors to verbs and nouns since those represent clear semantic meanings. In most cases, anchors are the same word in two different sentences. For this simple case, the anchors can be extracted by intersecting the set of words in the two sentences. However, it is desirable to recognize anchors which are not lexically equivalent. Proper names can usually be written in several ways. Translators also deviate from the anaphora used by an author. In some cases, we can infer with reasonable certainty that two words are an anchor pair, while in other cases we can only guess. The first step in finding anchors is to extract the nouns and verbs in both sentences. Then, anchors are extracted using a simple multiple pass algorithm:

1. We construct a list of nouns and verbs for each aligned sentence. We then intersect the two lists to obtain words which appear in both sentences. We impose several constraints on the intersected list: the words must not be pronouns and must be unique, i.e., not have a duplicate in the same sentence. These anchors are fairly certain.

2. If there is one female noun word pair or male noun word pair, it is assumed it is an anchor pair.

3. If there is one person noun word pair or place noun word pair, it is assumed it is an anchor pair.

4. If there is one noun word pair, it is assumed it is an anchor pair.

5. We repeat the first step, without the constraints, and allow words which are

not exact matches, but contain long common substrings. These anchor pairs are marked as uncertain, and paraphrases based on them will be scored differently.

Illustration of algorithm:

Aligned sentences:

```
The dentist went to the city to visit his parents.
He visited his parents in the town.
```

1. A list of nouns from each sentence is generated:

Sentence 1	Sentence 2
dentist	He
city	parents
parents	town

2. "parents" occurs in both sentences without duplicates and thus the anchor (parent, parent) is created. Each word in the anchor pair is deleted from the list of nouns since they cannot be involved in any more anchors.

3. "city" and "town" are recognized as places. Because they are the only two places in their respective sentences, they are determined to be anchors. Similarly "dentist" and "He" are the only people unmatched in their respective sentences. Both anchor pairs (city, town) and (dentist, he) are generated and removed from the list.

4.3.2 Generating Paths from Links

Once we have a list of anchors, we can start to generate the paraphrases. For each pair of different anchors, we have 2 words in each sentence. We then try to find the shortest path between those two words, subject to minimum and maximum path length constraints. We also do not allow paths to use certain relations such as conjunction relations and comma relations because they serve structural purposes and are not strong semantic links. We generate the path using a breadth-first search. Since we are treating the links as bidirectional we must be careful to avoid cycles, so we keep the path we have generated so far. If valid paths are found between the same two anchors pairs, then the paths are packaged as a paraphrase and are examined more closely in the path reduction phase.

Example of paraphrase at this point (anchors are [gate, gate] and [nurse, nurse]):

```
gate <--J<-- of <--M<-- noise <--J<-- at -->CO--> <person>
Example of paraphrase in lexical form: at the noise of
<=====>
gate <--J<-- of <--M<-- sound <--J<-- at -->CO--> <person>
Example of paraphrase in lexical form: at the sound of
```

4.3.3 Path Generalization

Path generalization in the current system is limited to replacing people and places found in paths with their named entity placeholder: "<person>" and "<place>" respectively. Path generalization is important in creating broad coverage paraphrases and is a topic for further research.

4.3.4 Paraphrase Scoring

Paraphrase scoring is heavily dependent on how the paraphrase will be used. The scoring in this system is geared towards use of the paraphrases in information retrieval. An ideal paraphrase in information retrieval would be context independent and close in meaning. Consequently the scoring takes two factors into account when scoring a paraphrase: the frequency of the occurrence of the paraphrase and the variety of anchors from which the paraphrase was produced. The second criterion is used as an indicator for the variety of contexts the paraphrase appeared in. The initial score for any paraphrase is 1. For each additional paraphrase occurrence the score is incremented by the $\frac{1}{2}^n$ where n is the number of times the current set of anchors has been seen before. In addition, each increment receives a 50% penalty if the anchors were pronouns or were not exact matches.

Chapter 5

Evaluation

Table 5.1 contains general statistics for the various phases of paraphrase generation. We can see that the link parser was unable to parse one of the phrases in approximately 8% of the aligned regions. Since we allowed partial parsing, the parse usually failed due to an extremely long or difficult sentence. We can also see that only about 10% of the paraphrases were duplicates, which hinders our ability to discriminate between good and bad paraphrases.

5.1 Synonym Phrases Coverage

The coverage of this system refers to the number of extracted paraphrases compared to the number of paraphrases that were available to be learned. The paraphrase coverage is difficult to determine since the research topic is to determine how to extract paraphrases from aligned corpora. Barzilay[3] tries to determine coverage by having human judges manually extract paraphrases from a subset of the aligned data. Lin[12]

Total number of aligned sentences	27479
Total number of parsed aligned sentences	25292
Total number of anchors in sentences	99040
Total number of paraphrases	13275
Total number of unique paraphrases	12309

Table 5.1: General paraphrase extraction statistics

Length of paraphrase	Number of unique paraphrases
1	396
2	1520
3	1593
4	1472
5	1144
6	812
Total	6937

Table 5.2: Number of generated paraphrases grouped by length

notes that humans have a hard time extracting all possible paraphrases. It is easy to overlook paraphrases as common sense. We can compare our results to Barzilay's results since they come from the same data source. Barzilay's algorithm produced 9483 paraphrases, 6714 (70.8%) of which were single word paraphrases. Table 5.2 is the breakdown of number of paraphrases produced by our system, grouped by length, above a threshold of 1.0, and with path length less than 7. The length of the paraphrase is taken to be the length of longer of the two phrases in the paraphrase. The maximum length was restricted because longer paraphrases were very inaccurate.

The average paraphrase length was 3.56. While the number of paraphrases is smaller than that generated by Barzilay, our approach is promising for the extraction of moderate length paraphrases.

5.2 Synonym Phrases Accuracy

The accuracy of paraphrases is likewise difficult to measure. The evaluation of paraphrases is largely subjective and depends on the criteria used. In this case, we decide whether a paraphrase is correct by asking whether one path can be substituted for another in a large number of contexts. 99 paraphrases were chosen at random and were evaluated by a human judge. 44 paraphrases were judged to be correct as seen in Table 5.3.

Example of acceptable paraphrase:

Length of paraphrase	Accuracy
1	1/1 (100%)
2	12/19 (63%)
3	19/35 (54%)
4	3/10 (30%)
5	8/20 (40%)
6	1/12 (8%)
Total	44/99 (44%)

Table 5.3: Accuracy of generated paraphrases grouped by length

```
put -->K--> on <====> wear

Example: John put on his sweater <====> John wore his sweater
```

Example of unacceptable paraphrase:

```
arrangement -->MV--> in <====> of
```

Appendix A contains some interesting paraphrases generated by our system. The accuracy results show that the accuracy decreased with the length of the paraphrase. This makes sense, since larger paths are more likely to be affected by an incorrect parse. In general, the precision of the paraphrases was lower than expected. The main culprit seemed to be the questionable semantics of the links on paths. A more semantically deep parser than the link parser might help the path quality. In addition, further modification of the link parser linkages might also help improve the semantics of the link relations. Another problem was generating paraphrases from incorrect parses especially those for long sentences with many clauses. This is unavoidable given the very difficult nature of the corpus. Incorrect identification of anchors was also a problem. Errors due to misalignments were very rare, which makes sense since a misalignment with the same phrases is unlikely to happen again.

Chapter 6

Future Work

There are several avenues of future research to consider. Improvements in alignment and paraphrase generation can increase the precision and coverage of paraphrases. In addition, it is important to explore the applications of paraphrases in natural language applications such as information retrieval.

6.1 Improving Quality of Paraphrases

Since we are focusing on information retrieval applications, the quality of paraphrases will refer to their precision and coverage.

6.1.1 Using Different Representations for Paraphrases

Our system uses paths between anchor words as the basis of paraphrases. However, there are many cases where words outside of the path are important to the paraphrase. Another representation such as a tree might prove to be useful in representing paraphrases. I surmise it might be difficult to handle trees because the number of possible trees one could generate is quite large and one would have to choose from among them the ones that are useful. In addition to improving the representation itself, we can explore how the parsing of the sentences can be optimized to produce the most meaningful paths between words.

6.1.2 Increasing the Amount of Data

The current alignment algorithm only allows for alignment of similar documents with little or no reordering. This limits the documents that can be aligned and therefore restricts the amount of data available. Since the quality and coverage of the paraphrases is closely related to the amount of data available, a more general alignment algorithm should improve the overall performance of the system.

The alignment algorithm described here was mainly used to align strict translations of foreign novels. A more general algorithm might be able to align newspaper articles, encyclopedia entries, or dictionary entries on the same topic. This would greatly increase the amount of data available to the system. These other sources contain more differences and, therefore, only a subset of the sentences might be aligned.

A simple extension of the current algorithm to consider out of order substitutions will both be slower and less precise. It is tempting to increase the precision by decreasing the delete cost; however, doing so would only produce sentences which are very similar, which is not useful in extracting paraphrases. A more successful approach might be to consider the sentence and its context, the hypothesis being that general discourse rules create ordering and proximity dependencies between sentences. An alternate approach is to research incorporating some of the thematic extraction methods from multi-document summarization into an alignment tool.

6.2 Paraphrase Generalization

Paraphrase generalization is the reduction of paraphrases to their most basic form. Often the paraphrases are encumbered with extra relations that do not contribute to the paraphrase. Since we want the paraphrases to be as general as possible, it is useful to recognize which parts of the paraphrase are unnecessary.

Unfortunately, this is often a difficult problem because some semantic understanding is needed. For example, it is difficult to differentiate cases in which the prepositional phrase is needed and cases where the prepositional phrase can be eliminated. Sometimes the preposition is needed to pick the correct sense of a single word.

Prepositions not quite important:

```
fling -->MV--> on <====> throw -->MV--> on
Example: John flung on his hat <====> John threw on his hat
```

Prepositions somewhat important:

```
at -->J--> door -->M--> of <===> outside
Example: John stood at the door of the room <====>
 John stood outside the room.
```

6.3 Deducing Syntactic and Morphological Rules

After extracting a large number of paraphrases, there likely will be many paraphrases which follow syntactic or morphological rules. The aggregation and development of paraphrase rules could significantly increase the coverage of the paraphrase system. For example, paraphrases can be the results of alternation of verb forms and be applied to any verb in a verb group. Beth Levin [9] studied many different verb alternation rules organized by their application to specific verb groups. However, it is difficult to automatically recognize verb groups because they are often organized according to the semantics of a verb.

6.4 Evaluation of Usefulness of Synonym Phrases in IR

Generation of paraphrases is not a particularly useful end; therefore, it is imperative to look at possible applications of paraphrases in natural language systems. Barzilay [3] uses paraphrases in multiple document summarization. Since the paraphrases are based on link relations, it is natural to examine the application of synonym phrases in an information retrieval system such as Sapere [13].

Sapere is an ideal information retrieval tool to test this system's paraphrases on because the paraphrases are based on link relations and thus can be indexed within

Sapere's relational framework. A simple experiment would be to expand a query with all the matching paraphrases and measure the performance of the information retrieval system.

Chapter 7

Conclusion

With the wealth of information available in electronic format, a lot of it free text, it is more important than ever to improve information retrieval. Query expansion can help improve information retrieval by alleviating the problems of polysemy and synonymy. Current techniques focus on query expansion with single word synonyms; however, more complicated inferences and meanings often are represented by multiple word paraphrases. The purpose of this thesis was to construct a system capable of unsupervised generation of both single and multiple word synonyms.

The final results are promising; however, the dearth of data greatly limited the possible scope of the generated paraphrases. Since longer paraphrases are less frequent, we need as much aligned data as we can obtain. Alternative sources of aligned sentences such as variations of Associated Press articles can provide us with a constant stream of aligned data. Other sources such as encyclopedia entries can also be mined for aligned sentences.

Kids naturally pick up different ways of expressing an idea through interaction with the real world. Currently, computers cannot use the real world as a reference point for language; therefore, we need a simpler reference point. By providing a common reference point for expressions, aligned sentences can provide a wealth of information about different sentence structures and allow insight into variation in language. A more comprehensive analysis of produced paraphrases to recognize generalities and derive syntactic and morphological rules could significantly improve the quality and breadth of the generated paraphrases. Further testing in actual information retrieval engines will also help us learn what types of multiple word paraphrases are useful and how much they can help us.

Appendix A

The first line of each paraphrase is “***” followed by the score, paraphrase, and example context sentences (up to a maximum of 3).

```
23.195976129180096
’s
<====>
of


length <--O<-- ’s <--S<-- mammal <=====> length -->M--> of -->J--> mammal
From their simultaneous observations, they were able to estimate the
mammal’s minimum length at more than 350 English feet; this was
because both the Shannon and the Helvetia were of smaller dimensions,
although each measured 100 meters stem to stern.
:::::
In these simultaneous observations they thought themselves justified
in estimating the minimum length of the mammal at more than three hundred
and fifty feet, as the Shannon and Helvetia were of smaller dimensions
than it, though they measured three hundred feet over all.


animal -->YS--> ’s -->D--> existence <=====>
animal <--J<-- of <--M<-- existence
On the cetacean question no doubts arose in his mind, and he didn’t
allow the animal’s existence to be disputed aboard his vessel.
:::::
On the question of the monster there was no doubt in his mind, and he
would not allow the existence of the animal to be disputed on board.


frigate -->YS--> ’s -->D--> maneuver <=====>
frigate <--J<-- of <--M<-- manoeuvre
```

Meanwhile I was astonished at the frigate's maneuvers.

:::::

However, I was astonished at the manoeuvres of the frigate.

lip <--D<-- 's <--YS<-- <person> <=====> lip -->M--> of -->J--> <person>

A half smile curled the commander's lips; then, in a calmer tone: "Professor Aronnax," he replied, "do you dare claim that your frigate wouldn't have chased and cannonaded an underwater boat as readily as a monster?"

:::::

A half-smile curled the lips of the commander: then, in a calmer tone: "Monsieur Aronnax," he replied, "dare you affirm that your frigate would not as soon have pursued and cannonaded a submarine boat as a monster?"

***

1.75

cast

<====>

throw

lamp -->S--> cast -->O--> sort <=====> lamp -->S--> throw -->O--> sort

Our lamps cast a sort of brilliant twilight over the area, making inordinately long shadows on the seafloor.

:::::

Our lamps threw over this place a sort of clear twilight that singularly elongated the shadows on the ground.

flame -->S--> cast -->O--> light <=====> flame -->S--> throw -->O--> light

She lay on her back and stretched out her arms.The flames in the

fireplace cast a merry, flickering light on the ceilings.
:::::
The flame of the fire threw a joyous light upon the ceiling; she turned on her back, stretching out her arms.

***
1.75
attain
<====>
reach

unicorn -->S--> attain -->O--> length <=====>
unicorn -->S--> reach -->O--> length
"The common narwhal, or unicorn of the sea, often attains a length of sixty feet.
:::::
"The common narwhale, or sea unicorn, often reaches a length of sixty feet.

<person> -->S--> attain -->O--> ideal <=====>
<person> -->S--> reach -->O--> ideal
Emma was privately pleased to feel that she had so very quickly attained this ideal of ethereal languor, inaccessible to mediocre spirits.
:::::
Emma was secretly pleased that she had reached at a first attempt the rare ideal of pale lives, never attained by mediocre hearts.

***
2.5
beam -->M--> of
<====>

ray -->M--> of

taper <--J<-- of <--M<-- beam -->S--> seem <=====>

seem <--S<-- ray -->M--> of -->J--> taper

And the beams of the two wax tapers burning on the chest of drawers seemed to her like dazzling emanations of divine light.

:::::

The curtains of the alcove floated gently round her like clouds, and the rays of the two tapers burning on the night-table seemed to shine like dazzling halos.

<person> <--J<-- of <--M<-- beam <--AN<-- level <=====>

level -->AN--> ray -->M--> of -->J--> <person>

The shades of evening had begun to fall; the level beams of the sun, shining through the branches, dazzled her eyes.

:::::

Evening shadows were falling, and the level rays of the sun streamed through the branches and dazzled her eyes.

<person> <--J<-- of <--M<-- beam -->S--> pulling <=====>

pulling <--S<-- ray -->M--> of -->J--> <person>

The beam of light that shone up directly from below was pulling the weight of her body towards the abyss.

:::::

The rays of bright light reflected directly up to her from below were pulling the weight of her body toward the abyss.

***

1.5

enter

<====>

pass -->MV--> into

<person> -->S--> enter -->O--> corridor <=====>

<person> -->S--> pass -->MV--> into -->J--> <person>

She entered the corridor into which the laboratory door opened.

:::::

She passed into the hall off which opened the laboratory door.

<person> -->S--> enter -->O--> passage <=====>

<person> -->S--> pass -->MV--> into -->J--> <person>

She entered the passage, where the laboratory door was.

:::::

She passed into the hall off which opened the laboratory door.

***

1.25

avert

<====>

turn -->K--> away

calamity -->S--> avert -->O--> head <=====>

calamity -->S--> turn -->K--> away -->K--> <person>

At the word "nurse" which brought back her adulteries and her calamities, Madame Bovary averted her head, as though another, stronger, poison had risen to her mouth and filled her with revulsion.

:::::

And at this name, that carried her back to the memory of her adulteries and her calamities, Madame Bovary turned away her head, as at the loathing of another bitterer poison that rose to her mouth.

calamity -->S--> avert -->O--> head <=====>

calamity -->S--> turn -->K--> away -->K--> <person>

At the word "nurse" which brought back her adulteries and her calamities, Madame Bovary averted her head, as though another, stronger, poison had risen to her mouth and filled her with revulsion.

:::::

At the sound of this name, which brought back the memory of her sins and her calamities, Madame Bovary turned away her head, as though her gorge had risen at the taste of a poison more virulent and more bitter than that other.

***

1.25

thought -->MV--> about

<====>

thought -->OF--> of

<person> -->S--> thought -->MV--> about -->J--> <person> <=====>

<person> -->S--> thought -->OF--> of -->J--> <person>

She thought very little about him now.

:::::

She scarcely thought of him now.

<person> -->S--> thought -->MV--> about -->J--> <person> <=====>

<person> -->S--> thought -->OF--> of -->J--> <person>

She thought very little about him now.

:::::

She never thought of him now.

***

1.0

rush -->K--> over -->K--> to

<====>

run -->MV--> to

<person> <--J<-- to <--K<-- over <--K<-- rush <--S<-- <person> <=====>

<person> <--J<-- to <--MV<-- run <--S<-- <person>

And he rushed over to his son, who had just jumped into a heap of lime to whiten his shoes.

:::::

And he ran to his son, who had just precipitated himself into a heap of lime in order to whiten his boots.

***

1.5

from -->J--> amount -->M--> of

<====>

from -->J--> quantity -->M--> of

food <--J<-- of <--M<-- amount <--J<-- from <--MV<-- <person> <=====>

food <--J<-- of <--M<-- quantity <--J<-- from <--MV<-- <place>

Madame Lefrancois could tell it from the amount of food he left on his plate.

:::::

Madame Lefrancois could see that well enough from the quantity of food he left on his plate.

food <--J<-- of <--M<-- amount <--J<-- from <--MV<-- saw <=====>

food <--J<-- of <--M<-- quantity <--J<-- from <--MV<-- <place>

He was sadder than ever, as Madame Lefrancois saw from the amount of

food he left on his plate.
:::::
Madame Lefrancois could see that well enough from the quantity of food
he left on his plate.

***
1.25
to -->J--> amelioration -->M--> of
<====>
to -->J--> improvement -->M--> of

apply -->MV--> to -->J--> amelioration -->M--> of -->J--> soil <=====>
apply -->MV--> to -->J--> improvement -->M--> of -->J--> soil
Apply yourselves, above all, to the amelioration of the soil, to good
manures, to the development of the equine, bovine, ovine, and porcine races.
:::::
Apply yourselves above all to the improvement of the soil, to rich
fertilizers, to the development of fine breeds, equine, bovine, ovine
and porcine.

apply -->MV--> to -->J--> amelioration -->M--> of -->J--> soil <=====>
apply -->MV--> to -->J--> improvement -->M--> of -->J--> soil
Apply yourselves, above all, to the amelioration of the soil, to good
manures, to the development of the equine, bovine, ovine and porcine races.
:::::
Apply yourselves above all to the improvement of the soil, to rich
fertilizers, to the development of fine breeds, equine, bovine, ovine
and porcine.

***

1.25

discover -->MV--> at

<====>

make -->K--> out -->K--> at

<person> -->S--> discover -->MV--> at -->J--> bottom <=====>

<person> -->S--> make -->K--> out -->K--> at -->J--> bottom

Finally he discovered a small "R" at the bottom of the second page.

:::::

He looked, and after a time he made out a little R at the bottom of the second page.

<person> -->S--> discover -->MV--> at -->J--> bottom <=====>

<person> -->S--> make -->K--> out -->K--> at -->J--> bottom

At last he discovered a small R at the bottom of the second page.

:::::

He looked, and after a time he made out a little R at the bottom of the second page.

***

1.0

convey -->O--> impression -->M--> leave -->MV--> under

<====>

retrace -->O--> impression -->M--> leave -->MV--> under

<person> <--J<-- under <--MV<-- leave <--M<-- impression <--O<-- convey <--I<-- can <=====>

<person> <--J<-- under <--MV<-- leave <--M<-- impression <--O<-- retrace <--I<-- can

And now, how can I convey the impressions left on me by this stroll under the waters.

:::::

And now, how can I retrace the impression left upon me by that walk under the waters?

***

1.0

show -->K--> off

<====>

was -->P--> in -->J--> contrast -->M--> to

hair -->S--> show -->K--> off -->K--> body <=====>

hair -->S--> was -->P--> in -->J--> contrast -->M--> to -->J--> body

Their woolly hair, with a reddish tinge, showed off on their black shining bodies like those of the Nubians.

:::::

Their woolly, red-tinted hair was in sharp contrast to their bodies, which were black and glistening like those of Nubians.

***

1.25

for <--MV<-- name

<====>

after <--MV<-- call <--P<-- be <--I<-- to <--TO<-- wanted

child -->M--> name -->MV--> for -->J--> <person> <=====>

child <--O<-- wanted -->TO--> to -->I--> be -->P--> call -->MV--> after -->J--> <person>

Charles wanted the child named for its mother, Emma was opposed.

:::::

```
Charles wanted the child to be called after her mother; Emma opposed this.

child -->M--> name -->MV--> for -->J--> <person> <=====>
child <--O<-- wanted -->TO--> to -->I--> be -->P--> call -->MV-->
after -->J--> <person>
Charles wanted the child named for its mother, Emma was opposed.
:::::
Charles wanted the child to be called after her mother.
```

Bibliography

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (DL'00)*, 2001.

[2] E. Agichtein, S. Lawrence, and L. Gravano. Learning search engine specific query transformations for question answering. In *Proceedings of the 10th International World-Wide Web Conference (WWW10)*, 2001.

[3] Regina Barzilay and Kathleen McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the ACL/EACL, Toulouse*, 2001.

[4] Christos Faloutsos and Douglas W. Oard. A survey of information retrieval and filtering methods. Technical Report CS-TR-3514, 1995.

[5] Sumio Fujita. Discriminative power and retrieval effectiveness of phrasal indexing terms. In *Proceedings of the ACL 2000 Workshop on Recent Advances in NLP and IR*, 2000.

[6] William A. Gale and Kenneth Ward Church. A program for aligning sentences in bilingual corpora. In *Meeting of the Association for Computational Linguistics*, pages 177–184, 1991.

[7] Christian Jacquemin, Judith L. Klavans, and Evelyne Tzokermann. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proceedings of the Thirty-fifth Annual Meeting of the Association for Computational Linguistics ((E)ACL 1997)*, 1997.

[8] B. Katz and B. Levin. Exploiting lexical regularities in designing natural language systems. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING '88)*, 1988.

[9] Beth Levin. *English Verb Classes and Alternations*. The University of Chicago Press, Chicago and London, 1993.

[10] D. Lin. Dependency-based evaluation of minipar. In *In Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation, Granada, Spain*, 1998.

[11] Dekang Lin. Extracting collocations from text corpora. In *Workshop on Computational Terminology*, 1998.

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

[13] Jimmy J. Lin. Indexing and retrieving natural language using ternanry expressions. Master's thesis, Massachussetts Institute of Technology, 2001.

[14] Rada Mihalcea and Dan Moldovan. Semantic indexing using wordnet senses. In *Proceedings of ACL 2000 Workshop on Recent Advances in NLP and IR*, 2000.

[15] George Miller, Richard Beckwith, Christiane Felbaum, Derek Gross, and Katherine Miller. Introduction to wordnet: An online lexical database.

[16] Franz Josef Och and Hermann Ney. Statistical machine translation.

[17] Stephen D. Richardson, William B. Dolan, and Lucy Vanderwende. Mindnet: Acquiring and structuring semantic information from text. Technical Report TR-98-23, Microsoft Research, 1998.

[18] M. Sanderson. Word-sense disambiguation and information retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Reasearch and Development in Information Retrieval*, 1997.

[19] H. Schutze and J. Pederson. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995.

[20] Daniel Sleator and Davy Temperly. Parsing english with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technology*, 1993.

[21] Ozlem Uzuner. Word sense disambiguation applied to information retrieval. Master's thesis, Massachussetts Institute of Technology, 1998.

[22] Ellen Voorhees. On expanding query vectors with lexically related words. In *SIGIR '93, Proceedings of the sixteenth annual international ACM SIGIR Conference on Research and Development in information retrieval*, 1993.

[23] Peter Wallis. Information retrieval based on paraphrase. In *Proceedings of PACLING Conference*, 1993.

[24] Callan J.P.and Croft W.B. and Harding S.M. The INQUERY retrieval system. In *Proceedings of 3rd International Conference on Database and Expert Systems Applications*, 1992.

[25] D. Yarowsky. One sense per collaction. In *ARPA Human Language Technology Workshop*, 1993.

[26] Chengxiang Zhai, Xiang Tong, Natasa Milic-Frayling, and David A. Evans. Evaluation of syntactic phrase indexing - CLARIT NLP track report. In *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, 1996.