Sentence Alignment for Monolingual Comparable Corpora

Regina Barzilay
Department of Computer Science
Cornell University
Ithaca, NY 14853
regina@cs.cornell.edu

Noemie Elhadad
Department of Computer Science
Columbia University
New York, NY 10027
noemie@cs.columbia.edu

Abstract

We address the problem of sentence alignment for monolingual corpora, a phenomenon distinct from alignment in parallel corpora. Aligning large comparable corpora automatically would provide a valuable resource for learning of text-to-text rewriting rules. We incorporate context into the search for an optimal alignment in two complementary ways: learning rules for matching paragraphs using topic structure and further refining the matching through local alignment to find good sentence pairs. Evaluation shows that our alignment method outperforms state-of-the-art systems developed for the same task.

1 Introduction

Text-to-text generation is an emerging area of research in NLP (Chandrasekar and Bangalore, 1997; Caroll et al., 1999; Knight and Marcu, 2000; Jing and McKeown, 2000). Unlike in traditional concept-to-text generation, text-to-text generation applications take a text as input and transform it into a new text satisfying specific constraints, such as length in summarization or style in text simplification. One exciting new research direction is the automatic induction of such transformation rules. This is a particularly promising direction given that there are naturally occurring examples of comparable texts that convey the same information yet are written in different styles. Presented with two such texts, one can pair sentences that convey the same information, thereby building a training set of rewriting examples for the domain to which the texts belong. We believe that automating this process will provide researchers in text-to-text generation with valuable training and testing resources, just as techniques to align multilingual parallel corpora boosted research in Machine Translation (MT).

In this paper, we address the task of aligning sentences in text pairs. We focus on monolingual comparable corpora, that is, texts in the same language (*e.g.*, English) that overlap in the information they convey. Stories about the same events from different press agencies and texts presenting the same information to experts and lay people are two examples.

In MT, the task of sentence alignment was extensively studied for parallel corpora.[1] A typical sentence alignment algorithm can be roughly described as a two-step process: (1) for each sentence pair compute a local similarity value, independently of the other sentences; (2) find an overall sequence of mapped sentences, using both the local similarity values and additional features.

In the case of monolingual corpora, step (2) might seem unnecessary. Since the texts share the same language, it would be enough to choose for local similarity a function based on lexical cues only and select sentence pairs with high lexical similarity. Even a simple lexical function (*e.g.*, one that counts word overlap) could produce an accurate alignment.

[1] Sentence alignment for comparable multilingual corpora was not addressed in previous research. Comparable corpora have primarily been used to build bilingual lexical resources (Fung and Yee, 1998).

After all, two sentences which share most of their words are likely to paraphrase each other. The problem is that there are many sentences that convey the same information but have little surface resemblance. As a result, simple word counts cannot distinguish the matching pair (A) in Figure 1 from the unrelated pair (B). An accurate local similarity measure would have to account for many complex paraphrasing phenomena. A simple, weak lexical similarity function alone is not sufficient.

(A)	· Petersburg served as the capital of Russia for 200 years. · For two centuries Petersburg was the capital of the Russian Empire.
(B)	· The city is also the country's leading port and center of commerce. · And yet, as with so much of the city, the port facilities are old and inefficient.

Figure 1: Sentence pairs from our corpus sharing two content words. (A) is a matching pair, (B) is not.

In MT, a weak similarity function is compensated for by searching for a globally optimal alignment, using dynamic programming or taking advantage of the geometric/positional or contextual properties of the text pair (Gale and Church, 1991; Shemtov, 1993; Melamed, 1999). But these techniques operate on the assumptions that there are limited insertions and deletions between the texts and that the order of the information is roughly preserved from one text to another.

Texts from comparable corpora, as opposed to parallel corpora, contain a great deal of "noise." In Figure 2 which plots the manually identified alignment for a text pair in our corpus, only a small fraction of the sentences got aligned (35 out of 31 $\times$ 270 sentence pairs), which illustrates that there is no complete information overlap. Consider two texts written by different press agencies: while both report on the same events, one may contain additional interviews and the other, background information. Another distinguishing characteristic of comparable corpora is that the order in which the information is presented can differ greatly from one text to another. Analysis of comparable texts in different domains (Paris, 1993; Barzilay et al., 2002) showed that there is wide variability in the order in which the same information can be presented. This is also illustrated in Figure 2.

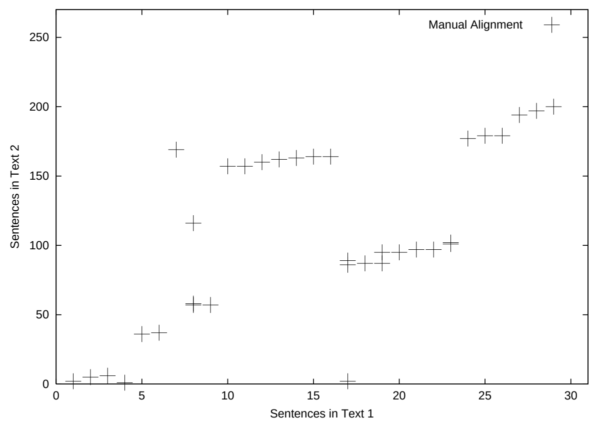

Figure 2: Manual alignment for a text pair in our corpus. A point in (x,y) indicates that the sentences x and y match.

We investigate a novel approach informed by text structure for sentence alignment. Our method emphasizes the search for an overall alignment, while relying on a simple local similarity function. We incorporate context into the search process in two complementary ways: (1) we map large text fragments using hypotheses learned in a supervised fashion and (2) we further refine the match through local alignment within mapping fragments to find sentence pairs. When the documents in the collection belong to the same domain and genre, the fragment mapping takes advantage of the topical structure of the texts. This structure is derived in an unsupervised fashion by analyzing commonalities among texts on each side of the comparable corpora separately. Our experiments show that our overall approach identifies even pairs with low lexical similarity. We also found that a fully unsupervised method using a minimalist representation of contextual information, *viz.*, paragraph-level lexical similarity, outperforms existing methods based on complex local similarity functions.

In the next section, we provide an overview of existing work on monolingual sentence alignment. Section 3 describes our algorithm. In sections 4 and 5, we report on data collection and evaluation.

2 Related Work

Most of the work in monolingual corpus alignment is in the context of summarization. In single document summarization, alignment between full documents and summaries written by humans is used to learn rules for text compression. Marcu (1999) computes sentence similarity using a cosine-based metric. Jing (2002) identifies phrases that were cut and pasted together using a Hidden Markov Model with features incorporating word identity and positioning within sentences, thereby providing an alignment of the document and its summary. However, both of these methods construct an alignment by looking at sentences one at a time, independently of the decisions made about other sentences. Because summaries often reuse original document text to a large extent, these methods achieve good results.

In the context of multidocument summarization, SimFinder (Hatzivassiloglou et al., 1999) identifies sentences that convey similar information across input documents to select the summary content. Even though the input documents are about the same subject, they exhibit a great deal of lexical variability. To address this issue, SimFinder employs a complex similarity function, combining features that extend beyond a simple word count and include noun phrase, proper noun, and WordNet sense overlap. Since many documents are processed in parallel, clustering is used to combine pairwise alignments. In contrast to our approach, SimFinder does not take the context around sentences into account.

3 Algorithm

Given a comparable corpus consisting of two collections and a training set of manually aligned text pairs from the corpus, the algorithm follows four main steps. Steps 1 and 2 take place at training time. Steps 3 and 4 are carried out when a new text pair ($Text_1$, $Text_2$) is to be aligned.

1. **Topical structure induction**: by analyzing multiple instances of paragraphs within the texts of each collection, the topics characteristic of the collections are identified through clustering. Each paragraph in the training set gets assigned the topic it verbalizes (Section 3.1.1.)
2. **Learning of structural mapping rules**: using the training set, rules for mapping paragraphs are learned in a supervised fashion (Section 3.1.2).
3. **Macro alignment**: given a new unseen pair ($Text_1$, $Text_2$), each paragraph is automatically assigned its topic. Paragraphs are mapped following the learned rules (Section 3.2).
4. **Micro alignment**: for each mapped paragraph pair, a local alignment is computed at the sentence level. The final alignment for the text pair is the union of all the aligned sentence pairs (Section 3.3).

3.1 Off-Line Processing

Given two sentences with moderate lexical similarity, we may not have enough evidence to decide accurately whether they should be aligned. Looking at the broader context they appear in can provide additional insight: if the *types* of information expressed in the contexts are similar, then the *specific* information expressed in the sentences is more likely to be the same. On the other hand, if the types of information in the two contexts are unrelated, chances are that the sentences should not be aligned. In our implementation, context is represented by the paragraphs to which the sentences belong.[2] Our goal in this phase is to learn rules for determining whether two paragraphs are likely to contain sentences that should be aligned, or whether, on the contrary, two paragraphs are unrelated and, therefore, should not be considered for further processing.

A potentially fruitful way to do so is to take advantage of the topical structure of texts. In a given domain and genre, while the texts relate different subjects, they all use a limited set of topics to convey information; these topics are also known as the Domain Communication Knowledge (Kittredge et al., 1991). For instance, most texts describing diseases will have topics such as "symptoms" or "treatment."[3] If the task is to align a disease description written for physicians and a text describing the same disease for lay people, it is most likely that sentences within the topic "symptoms" in the expert version will map to sentences describing the symptoms in the lay version rather than those describing treatment options. If we can automatically identify the topic each paragraph conveys, we can decide more accurately whether two paragraphs are related and should be mapped for further processing.

[2] Texts without adequate paragraph marking could be segmented using tools such as TextTiling (Hearst, 1994).

[3] We use the term topic differently than it is commonly used in the topic detection task– there, a "topic" would designate which disease is described.

In the field of text generation, methods for representing the semantic structure of texts have been investigated through text schemata (McKeown, 1985) or rhetorical structures (Mann and Thompson, 1987). In our framework, we want to identify the different topics of the text, but we are not concerned with the relations holding between them or the order in which they typically appear. We propose to identify the topics typical to each collection in the comparable corpus by using clustering, such that each cluster represents a topic in the collection.

The process of learning paragraph mapping rules is accomplished in two stages: first, we identify the topics of each collection, $Corpus_1$ and $Corpus_2$, and label each paragraph with its specific topic. Second, using a training set of manually aligned text pairs, we learn rules for mapping paragraphs from $Corpus_1$ to $Corpus_2$. Two paragraphs are considered mapped if they are likely to contain sentences that should be aligned.

3.1.1 Vertical Paragraph Clustering

We perform a clustering at the paragraph level for each collection. We call this stage Vertical Clustering because all the paragraphs of all the documents in $Corpus_1$ get clustered, independently of $Corpus_2$; the same goes for the paragraphs in $Corpus_2$. At this stage, we are only interested in identifying the topics of the texts in each collection, each cluster representing a topic.

We apply a hierarchical complete link clustering. Similarity is a simple cosine measure based on the word overlap of the paragraphs, ignoring function words. Since we want to group together paragraphs that convey the same type of information across the documents in the same collection, we replace all the text-specific attributes, such as proper names, dates and numbers, by generic tags.[4] This way, we ensure that two paragraphs are clustered not because they relate the same specific information, but rather, because they convey the same type of information (an example of two automatically clustered paragraphs is shown in Figure 3). The number of clusters for each collection is a parameter tuned on our training set (see Section 4).

[4] We crudely consider any words with a capital letter a proper name, except for each sentence's first word.

Lisbon has a mild and equable climate, with a mean annual temperature of 63 degree F (17 degree C). The proximity of the Atlantic and the frequency of sea fogs keep the atmosphere humid, and summers can be somewhat oppressive, although the city has been esteemed as a winter health resort since the 18th century. Average annual rainfall is 26.6 inches (666 millimetres).

Jakarta is a tropical, humid city, with annual temperatures ranging between the extremes of 75 and 93 degree F (24 and 34 degree C) and a relative humidity between 75 and 85 percent. The average mean temperatures are 79 degree F (26 degree C) in January and 82 degree F (28 degree C) in October. The annual rainfall is more than 67 inches (1,700 mm). Temperatures are often modified by sea winds. Jakarta, like any other large city, also has its share of air and noise pollution.

Figure 3: Two automatically clustered paragraphs in the same collection (without date, number, and name substitution).

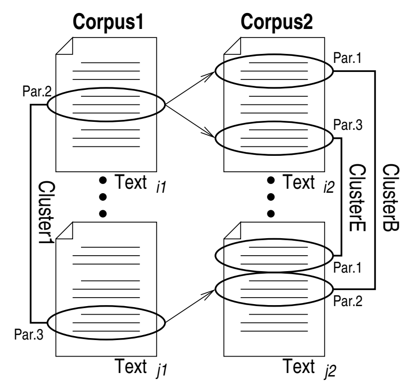

Figure 4: The training set for the paragraph mapping step. An arrow between two paragraphs indicates they contain at least one aligned sentence pair.

3.1.2 Horizontal Paragraph Mapping

Once the different topics, or clusters, are identified inside each collection, we can use this information to learn rules for paragraph mapping (Horizontal Mapping between texts from $Corpus_1$ and texts from $Corpus_2$). Using a training set of text pairs, manually aligned at the sentence level, we consider two paragraphs to map each other if they contain at least one aligned sentence pair (see Figure 4).

Our problem can be framed as a classification task: given training instances of paragraph pairs (P, Q) from a text pair, classify them as mapping or not. The features for the classification are the lexical similarity of P and Q, the cluster number of P, and the cluster number of Q. Here, similarity is again a simple cosine measure based on the word overlap of the two paragraphs.[5] These features are weak indicators by themselves. Consequently, we use the publicly-available classification tool BoosTexter (Singer and Schapire, 1998) to combine them accurately.[6]

3.2 Macro Alignment: Find Candidate Paragraph(s)

At this stage, the clustering and training are completed. Given a new unseen text pair ($Text_1$, $Text_2$), the goal is to find a sentence alignment between them. Two sentences with very high lexical similarity are likely to be aligned. We allow such pairs in the alignment independently of their context. This step allows us to catch the "easy" paraphrases. We focus next on how our algorithm identifies the less obvious matching sentence pairs.

For each paragraph in each text, we identify the cluster in its collection it is the closest to. Similarity between the paragraph and each cluster is computed the same way as in the Vertical Clustering step. We then apply mapping classification to find the mapping paragraphs in the text pair (see Figure 5).

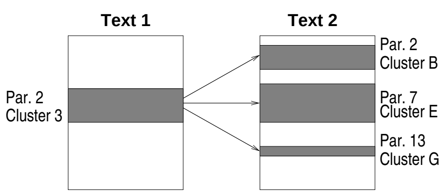

Figure 5: Macro Alignment: a paragraph in $Text_1$ and its mapped candidate paragraphs in $Text_2$.

3.3 Micro Alignment: Find Sentence Pair(s)

Once the paragraph pairs are identified in ($Text_1$, $Text_2$), we want to find, for each paragraph pair, the (possibly empty) subsets of sentence pairs which constitute a good alignment. Context is used in the following way: given two sentences with moderate similarity, their proximity to sentence pairs with high similarity can help us decide whether to align them or not.

To combine the lexical similarity (again using cosine measure) and the proximity feature, we compute local alignments on each paragraph pair, using dynamic programming. The local alignment we construct fits the characteristics of the data we are considering. In particular, we adapt it to our framework to allow many-to-many alignments and some flips of order among aligned sentences. Given sentences i and j, their local similarity $sim(i,j)$ is:

$$sim(i,j) = cos(i,j) - mismatch\_penalty$$

The weight $s(i,j)$ of the optimal alignment between the two sentences is computed as follows:

$$s(i,j) = max \begin{cases} s(i,j-1) - skip\_penalty \\ s(i-1,j) - skip\_penalty \\ s(i-1,j-1) + sim(i,j) \\ s(i-1,j-2) + sim(i,j) + sim(i,j-1) \\ s(i-2,j-1) + sim(i,j) + sim(i-1,j) \\ s(i-2,j-2) + sim(i,j-1) + sim(i-1,j) \end{cases}$$

The mismatch penalty penalizes sentence pairs with very low similarity measure, while the skip penalty prevents *only* sentence pairs with high similarity from getting aligned.

4 Evaluation Setup

The Data. We compiled two collections from the Encyclopedia Britannica and Britannica Elementary. In contrast to the long (up to 15-page) detailed articles of the Encyclopedia Britannica, Britannica Elementary contains one- to two-page entries targeted towards children. The elementary version generally contains a subset of the information presented in the comprehensive version, but there are numerous cases when the elementary entry contains additional or more up-to-date pieces of information.[7] The two collections together exhibit many instances of complex rewriting.

We collected 103 pairs of comprehensive/elementary city descriptions. We set aside a testing set of 11 text pairs. The rest (92 pairs) was used for the Vertical Clustering. Nine text pairs were used for training (see Table 1 for statistics).

Human Annotation. Each text pair in the training and testing sets was annotated by two annotators.[8] In our guidelines to them, we defined two sentences as aligned if they contain at least one clause

[5] At this stage, we want to match on text-specific information, unlike in the Vertical Clustering. We therefore use the original text, without any substitution, to compute the similarity.

[6] Because BoosTexter cannot form conjunctive hypotheses, we add a feature which encodes the combination of two cluster numbers.

[7] Britannica Elementary is a new feature of the encyclopedia, not all entries in the original Britannica have been fully updated.

[8] All the annotators were native speakers of English. The authors did not take part in the annotation.

	Sentences			Paragraphs		
	Min	Max	Avg	Min	Max	Avg
Comp. Train	87	313	180	19	59	37
Comp. Test	138	308	200	32	63	43
Elem. Train	34	64	47	8	12	10
Elem. Test	27	75	45	6	16	10

Table 1: Statistics for the training and testing sets for the comprehensive and elementary versions.

Range	**Training**	**Testing**
0%–40%	149 (46.6%)	127 (45.2%)
40%–70%	103 (32.2%)	96 (34.2%)
70%–100%	68 (21.2%)	58 (20.6%)

Table 2: Distribution of manually aligned sentence pairs among different similarity ranges.

that expresses the same information. We allowed many-to-many alignments. On average, each annotator spent 50 minutes per text pair. While the annotators agreed for most of the sentence pairs they identified, there were some cases of disagreement. Alignment is a tedious task, and sentence pairs can easily be missed even by a careful human annotator. For each text pair, a third annotator went through contested sentence pairs, deciding on a case-by-case basis whether to include it in the alignment. Overall, 320 sentence pairs were aligned in the training set and 281 in the testing set. The other sentence pairs which were not aligned served as negative examples, yielding a total of 4192 training instances and 3884 testing instances.[9]

As a confirmation that there is no order preservation in comparable corpora, there were up to nine order shifts in each of the annotated text pairs. Table 2 shows that a large fraction of manually aligned sentence pairs have low lexical similarity. Similarity is measured here by the number of words in common, normalized by the number of types in the shorter sentence.

Parameter Tuning. We tuned all the parameters on our training set, obtaining the following values: the skip penalty is 0.001, and the cosine threshold for selecting pairs with high lexical similarity is 0.5. BoosTexter was trained in 200 iterations. To find the optimal number of clusters for each collection, Vertical Clustering was performed with different numbers of clusters, ranging from 10 to 40; we selected the alternatives with the best performance on the training set: 20 for both collections.

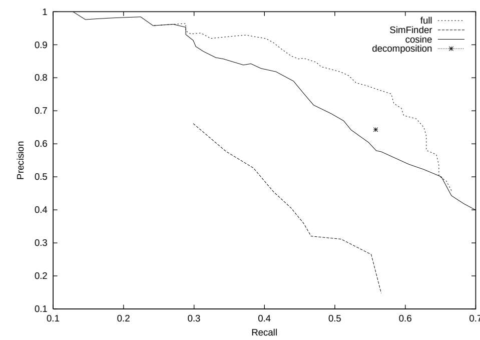

Figure 6: Precision/Recall for SimFinder, Cosine, Decomposition, and our full method.

5 Results

We first present the comparison of our method with other systems developed for the same task. Next we focus on the impact of individual components on the performance of our method.

5.1 Comparison with Other Systems

An obvious choice for a baseline in this task is the following: any two sentences are considered aligned if their cosine similarity exceeds a certain threshold. We also compare our algorithm with two state-of-the-art systems, SimFinder (Hatzivassiloglou et al., 1999) and Decomposition (Jing, 2002).[10] Figure 6 shows Precision/Recall curves for the different systems. For our full system, we obtain different values of Recall keeping constant the skip penalty and the cosine threshold and varying the value of the mismatch penalty from 0 to 0.45.[11] This setup results in recall values in a 25%–65% range. The curve for SimFinder was obtained by running SimFinder with different similarity thresholds, ranging from 0.1 to 0.95. In the case of Decomposition, there are several hard-coded parameters which are not trainable. As a result, we were able to obtain results for Decomposition only at a 55.8% Recall level. Table 3 reports Precision values at this level of Recall.

[9] Our corpus is available at `http://www.cs.columbia.edu/~noemie/alignment`.

[10] Jing (2002) reports that Decomposition outperforms the algorithm of Marcu (1999); we, therefore, did not compare our method against his system.

[11] Varying the mismatch penalty is a natural choice: varying the skip penalty produces a narrow range of Recall values, while the cosine threshold controls only a small portion of the sentence pairs that can be identified (the ones with high similarity).

System	Precision
SimFinder	24%
Cosine	57.9%
Decomposition	64.3%
Without Topic Mapping	71.6%
Without Local Alignment	73.3%
Full Method	**76.9%**
Full Method (with ideal clusters)	80.9%

Table 3: Precision at 55.8% Recall.

(*)	Gradually the German culture and language became more widespread in the city.
	Capping Prague's rebirth, it was designated a European City of Culture in 2000.
	Prague is a centuries-old city with a wealth of historic landmarks.
	The physical attractions and landmarks of Prague are many.

Figure 7: Aligned pairs, (*) denotes an incorrect alignment.

Our full method outperforms both the baseline ("*Cosine*") and the more complex systems ("*SimFinder*" and "*Decomposition*"). Interestingly, methods that use simple local similarity functions significantly outperform SimFinder (SimFinder was trained on newswire texts; we did not have access to SimFinder's training component for retraining on our corpus). This confirms our hypothesis that while it is an appealing idea, putting all one's eggs in the basket of a sophisticated local similarity measure to achieve good performance may be too hard a task. The simple cosine baseline is competitive with the Hidden Markov Model of Decomposition (Decomposition was specifically developed to identify sentence pairs with cut-and-paste transformations, not all possible paraphrase pairs). This suggests that when looking for an alignment, Cosine is a good, yet simple, starting local similarity measure. Adding on top of it an explicit search mechanism relying on the context surrounding the sentences, as in our method, results in a performance improvement of 19% at 55.8% Recall. Figure 7 shows examples of pairs identified by our method.

5.2 Analysis

Impact of Horizontal Paragraph Mapping. We hypothesize that exploiting the regularity in mapping between semantic units, such as topics, improves the alignment. We compared the performance of our full method with a variation that does not take any topical information into account. For the paragraph mapping, we replaced the learned rules by a single rule based on lexical similarity: two paragraphs are mapped if their cosine measure is above a pre-specified threshold.[12] This new mapping is a good point of comparison because it does not rely on any knowledge inferred from the other texts in the corpus. The results confirm our hypothesis: learning paragraph mapping based on topical structures improves the performance (see "*Without Topic Mapping*" vs. "*Full Method*", Table 3).

This experiment also shows that representing context, even simply using only the paragraphs and their lexical similarity, achieves higher performance than methods based on more complex local similarity functions. It is an important finding, because this simplified method can be used when topic structure cannot be derived (e.g., in heterogeneous collections) or when no training data is available, since it is unsupervised.

Impact of Cluster Quality. Our method uses clustering to identify the different topics of each collection. It is important to know how sensitive our overall algorithm is to the quality of the identified clusters. Fortunately, in our corpus, some of the texts contain section headings (*e.g.*, "Climate" or "Demography"). Even though our method ignores this piece of information, we used it to derive manually "ideal" clusters.[13] We obtained eight clusters for the elementary version and 11 for the comprehensive one. When feeding these ideal clusters instead of the automatically identified ones to the learning module for paragraph mapping, we achieve 4% improvement in Precision (at 55.8% Recall). We interpret this as a sign that the algorithm handles imperfect, automatically induced clusters fairly well.

Impact of Local Alignment. Our hypothesis for computing local alignments between pairs of mapped paragraphs is that our approach allows us to identify additional matching sentence pairs: if two sentences paraphrase each other but have a low cosine measure, looking at the sentence pairs around

[12] The threshold was tuned on our training data.

[13] The process was performed manually because the sections are different from one text to another, both in names and levels of detail, and because we needed to infer clusters for the paragraphs that did not have section headings.

Range	Full		Dec.		Cos.	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
0%–40%	50%	25%	34%	28%	23%	15%
40%–70%	85%	73%	82%	74%	66%	86%
70%–100%	95%	95%	93%	88%	90%	95%

Table 4: Precision and Recall for different ranges of lexical similarity for Decomposition, Cosine, and our full method.

them may increase their chances of getting selected.

We compared our full method with a version of our algorithm that does not perform local alignment ("*Without Local Alignment*"). Instead, it simply selects sentence pairs inside the mapped paragraphs based on their cosine measure. This incomplete version of the algorithm achieves 73.3% Precision (at 55.8% Recall), 3% lower than the full method, validating our hypothesis.

Impact of Lexical Similarity. We investigated how the performance of our method depends on the lexical similarity of the input sentences. Table 4 shows Precision and Recall for our method and others at three sentence-similarity ranges based on word overlap counts (at the overall Recall of 55.8%). Our method outperforms the Cosine baseline and Decomposition on all similarity ranges.

6 Conclusions

The central finding of our work is that context plays an important role in the task of sentence alignment for monolingual comparable corpora. A weak sentence similarity measure combined with contextual information outperforms methods based on sophisticated sentence similarity functions. Experiments show that a simple representation of context is helpful. Relying on a more elaborate representation, such as topical text structure, has an even stronger impact on performance.

Acknowledgments

We would like to thank Michael Elhadad, Klara Kedem, Kathy McKeown and Becky Passonneau for the useful discussions. Thanks to Mirella Lapata, Lillian Lee, Vincent Ng and Bo Pang for their helpful comments on previous drafts. We are grateful to Hongyan Jing for giving us access to her code for Decomposition. Finally we would like to thank all the annotators (they are many). This paper is based upon work supported in part by the National Science Foundation under ITR/IM grant IIS-0081334, Digital Library Initiative Phase II Grant No. IIS-98-17434, and a Sloan Research Fellowship to Lillian Lee. Any opinions, findings, and conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Sloan Foundation.

References

Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of EACL*.

Raman Chandrasekar and Srinivas Bangalore. 1997. Automatic Induction of Rules for Text Simplification. *Knowledge-Based Systems*, 10(3):183–190.

Pascale Fung and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of ACL*.

William Gale and Kenneth Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of ACL*.

Vasileios Hatzivassiloglou, Judith Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of EMNLP*.

Marti Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of ACL*.

Hongyan Jing and Kathleen McKeown. 2000. Cut and paste based summarization. In *Proceedings of NAACL*.

Hongyan Jing. 2002. Using hidden Markov modeling to decompose human-written summaries. *Computational Linguistics*, 28(4):527–543.

Richard Kittredge, Tanya Korelsky, and Owen Rambow. 1991. On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of AAAI*.

William Mann and Sandra Thompson. 1987. Rhetorical structure theory: description and construction of text structures. In G. Kempen, editor, *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*. Kluwer Academic.

Daniel Marcu. 1999. The automatic construction of large-scale corpora for summarization research. In *Proceedings of SIGIR*.

Kathleen McKeown. 1985. *Text Generation*. Cambridge University Press.

Dan Melamed. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1):107–130.

Cecile Paris. 1993. *User Modelling in Text Generation*. Frances Pinter Publishers.

Hadar Shemtov. 1993. Text alignment in a tool for translating revised documents. In *Proceedings of EACL*.

Yoram Singer and Robert Schapire. 1998. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of COLT*.