Learning Cross-document Structural Relationships using Boosting

Zhu Zhang
School of Information and
Department of EECS
University of Michigan
Ann Arbor, MI 48109
zhuzhang@umich.edu

Jahna Otterbacher
School of Information
University of Michigan
Ann Arbor, MI 48109
jahna@umich.edu

Dragomir Radev
School of Information and
Department of EECS
University of Michigan
Ann Arbor, MI 48109
radev@umich.edu

ABSTRACT

Multi-document discoure analysis has emerged with the potential of improving various information retrieval applications. Based on the newly proposed Cross-document Structure Theory (CST), this paper describes an empirical study that uses boosting to classify CST relationships between sentence pairs extracted from topically related documents. We show that the binary classifier for determining existence of structural relationships significantly outperforms the baseline. We also achieve promising results on the multi-class case in which the full taxonomy of relationships are considered.

Categories and Subject Descriptors

H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Miscellaneous; I.2.7 [**ARTIFICIAL INTELLIGENCE**]: Natural Language Processing—*discourse*; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning—*concept learning, induction*

General Terms

Algorithms, Experimentation

Keywords

discourse analysis, cross-document structure, classification, boosting

1. INTRODUCTION

Computational models for natural language discourse structure have been widely studied since the 1970s. They have gained increasing attention with the proliferation of web-based information resources and related applications such as text summarization and question answering. Recently, the study of multi-document discourse has emerged, which is different from traditional discourse analysis in that multiple related documents may be written in different styles and use different vocabulary.

Inspired by Rhetorical Structure Theory (RST) [11], the notion of Cross-document Structure Theory (CST) was proposed by [16]. The central idea is to posit a set of rhetorical relationships that hold between sentences cross topically-related documents. It has been shown that the availability of such information can help multi-document text summarization [21]. It is also conceivable that other IR-related applications, such as semantic entity and relation extraction (where semantic relations may instantiate cross document boundaries), and non-factoid question answering (where answer generation may demand "fusing" information from multiple documents), can potentially benefit from multi-document discourse structure.

However, as far as we are aware of, no work has been done to show whether the relationships posited in the CST framework can be automatically identified from free text. In this paper, we explore the possibility of classifying CST relationships by using a machine learning approach, specifically Boosting [20].

2. RELATED WORK

A similar effort has been recently attempted in the arena of RST. Marcu [12] proposed a first-order formalization of the high-level rhetorical structure of text, and provided a theoretical analysis and an empirical comparison of four algorithms for automatic derivation of text structures. A set of empirically motivated algorithms were designed for rhetorical parsing, i.e., determining the elementary textual units of a text, hypothesizing rhetorical relations that hold among these units, and eventually deriving the discourse structure of the text. Marcu's approach is basically knowledge-based. He relied on "cue phrases" in implementing algorithms to discover the valid RST trees for a single document. This approach is reasonable because of the conventions of writing and the valid assumption that authors tend to write documents using certain rhetorical techniques. However, in the case of multiple documents and cross-document relationships (links), we cannot expect to encounter a reliable analog to the cue phrase. This is because separate documents, even when they are related to a common topic, are generally not written with an overarching structure in mind. Particularly in the case of news, we are most often looking at articles which are written by different authors working from par-

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.

tially overlapping information as it becomes available. So, except in cases of explicit citation or repetition, we cannot expect to find a static phrase in one document which reliably indicates a particular relationship to some phrase in another document. Therefore, it may pay off to look for deeper-level cues and to pursue statistical approaches instead.

More recently, [13] presented a machine learning approach to classifying RST relationships. Only lexical features are used in the naive Bayes classifier, and part-of-speech information is only used for feature selection for comparison purposes. Worth noting is that the authors take advantage of the available linguistic knowledge and exploit various cues in obtaining training data. Doing so is, again, much harder in the cross-document context.

In SimFinder [5], the authors introduced a machine-learned similarity measure at the paragraph level, but they look at a slightly different problem, i.e., whether two paragraphs contain "common information". This doesn't directly fit into the CST framework and it is only a binary classification problem. What is inspiring is that they use various syntactic and semantic features in their logistic regression model.

3. PROBLEM DEFINITION

3.1 Cross-document Structure Theory

Cross-document Structure Theory (CST) is a functional theory for multi-document discourse structure. It is used to describe cross-document semantic connections, such as "elaboration", "contradiction", "attribution", and "historical background", among text units of related documents . CST is related to RST but assumes no deliberateness of writing and no underlying tree representation. While the graph-like representation look like "semantic hyperlinks" [19], the relationships are all linguistically motivated. We focus on sentence-level CST relationships in this study.

The full taxonomy of CST relationships, refined from those presented in [16] and [21], can be found in Figure 1. Notice that some CST relationships, such as *identity*, are symmetric (*multinuclear*, in RST terms), while some other ones, such as *subsumption*, do have directionality, i.e., they have *nucleus* and *satellite*. It is also worth noting that all CST relationships are domain-independent.

3.2 Formulation of the classification problem

As a first approximation, we cast the CST relationship identification problem in a standard classification framework. Conceptually, given an unordered sentence pair $P(S_1, S_2)$, where sentences S_1 and S_2 are from two different but topically related documents, we are interested in determining the type(s) of cross-document relationships between them.

In this paper, we investigate the following two scenarios:

- The binary classification scenario: here we are interested in the existence of cross-document relationships regardless of type. If the two sentences are related in any types, the pair is assigned a label "1", otherwise a "0".

- The full classification scenario: in this case we do care about the type(s) of cross-document relationships between the sentence pair. Moreover, it is possible for a single pair to have multiple labels (see section 4 and 5.2 for more detail).

For the full classification scenario, the class labels are adapted from the CST taxonomy presented in Figure 1. Therefore, there are 19 possible labels in total (18 CST types plus a special type "no relationship").

4. EXPERIMENTAL SETUP AND DATA COLLECTION

Due to the lack of systematic linguistic knowledge for the problem that we are addressing, we could not obtain training data as in [13]. Instead, we had to actively collect data and have human judges annotate the CST relationships.

As our first attempt, we collected six clusters of related news articles from various sources. The clusters were chosen to be diverse with respect to their topics, the time span across the documents, the cluster size, and the news agencies from which the articles were collected. Table 1 shows the characteristics of the clusters. The cluster names reflect the source from which the cluster of documents was obtained.

Three of the clusters were collected from secondary sources while three were collected by the authors. The DUC cluster was obtained from the 2001 Document Understanding Conference (DUC) training data, the HKNews cluster was taken from the Hong Kong News Corpus (LDC2000T46), and the Novelty cluster was a cluster from this year's TREC Novelty track test data. The Milan9 and Gulfair11 clusters were collected by the authors live from the Web from several news sites: USA Today, MSNBC, CNN, FOX News, the BBC, the Washington Post and ABC News. Finally, the NIE cluster was collected automatically using NewsInEssence, a publicly available research prototype [1].

The Milan9 cluster was used strictly for corpus development and judge training purposes. It was annotated for CST relationships by two authors in developing the markup scheme and the guidelines to be used by the independent judges to be hired. The five clusters that were annotated by the judges were DUC, Gulfair11, HKNews, NIE, and Novelty.

Human annotation is not only expensive, and the results are not always ideal. Human judges often do not agree, due to the inherently ambiguous nature of natural language. The large search space makes the situation even worse. In a ten-document cluster with 20 sentences on average in each document, for example, a human judge will have to examine roughly 18,000 sentence pairs if he or she wants to exhaust all possibilities. This is an incredibly tedious job in any sense, and because of that, it is very difficult for multiple judges to reach reasonable agreement on the annotation.

One possible way to alleviate the problem is to exploit the observation that CST relationships are unlikely to exist between sentences that are *lexically* very dissimilar to each other. In other words, certain similarity measures might behave as a useful proxy for finding CST-related sentence pairs. We experimented with a few lexical-level similarity metrics, including Cosine [18], word overlap, longest common subsequence and BLEU [14], and then measured their correlation with CST-relatedness. Using the very carefully annotated MI9 as "training" data, we found that word overlap rate 0.12 is the "best" cutoff criterion for selecting sentence pairs, in the sense that it helps minimize selected number of sentence pairs without losing too many CST-related pairs (the recall is 87.5%). We then applied this measure

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

ID	Relationship	Description	Text span 1 (S1)	Text span 2 (S2)
1	Identity	The same text appears in more than one location	Tony Blair was elected for a second term today.	Tony Blair was elected for a second term today.
2	Equivalence (Paraphrase)	Two text spans have the same information content	Derek Bell is experiencing a resurgence in his career.	Derek Bell is having a ”comeback year.”
3	Translation	Same information content in different languages	Shouts of “Viva la revolucion!” echoed through the night.	The rebels could be heard shouting, “Long live the revolution”.
4	Subsumption	S1 contains all information in S2, plus additional information not in S2	With 3 wins this year, Green Bay has the best record in the NFL.	Green Bay has 3 wins this year.
5	Contradiction	Conflicting information	There were 122 people on the downed plane.	126 people were aboard the plane.
6	Historical Background	S1 gives historical context to information in S2	This was the fourth time a member of the Royal Family has gotten divorced.	The Duke of Windsor was divorced from the Duchess of Windsor yesterday.
7	Citation	S1 explicitly cites document S2	An earlier article quoted Prince Albert as saying “I never gamble.”	Prince Albert then went on to say, “I never gamble.”
8	Modality	S1 presents a qualified version of the information in S2, e.g., using “allegedly”	Sean “Puffy” Combs is reported to own several multimillion dollar estates.	Puffy owns four multimillion dollar homes in the New York area.
9	Attribution	S1 presents an attributed version of information in S2, e.g. using “According to CNN,”	According to a top Bush advisor, the President was alarmed at the news.	The President was alarmed to hear of his daughter’s low grades.
10	Summary	S1 summarizes S2.	The Mets won the Title in seven games.	After a grueling first six games, the Mets came from behind tonight to take the Title.
11	Follow-up	S1 presents additional information which has happened since S2	102 casualties have been reported in the earthquake region.	So far, no casualties from the quake have been confirmed.
12	Indirect speech	S1 indirectly quotes something which was directly quoted in S2	Mr. Cuban then gave the crowd his personal guarantee of free Chalupas.	“I’ll personally guarantee free Chalupas,” Mr. Cuban announced to the crowd.
13	Elaboration (Refinement)	S1 elaborates or provides details of some information given more generally in S2	50% of students are under 25; 20% are between 26 and 30; the rest are over 30.	Most students at the University are under 30.
14	Fulfillment	S1 asserts the occurrence of an event predicted in S2	After traveling to Austria Thursday, Mr. Green returned home to New York.	Mr. Green will go to Austria Thursday.
15	Description	S1 describes an entity mentioned in S2	Greenfield, a retired general and father of two, has declined to comment.	Mr. Greenfield appeared in court yesterday.
16	Reader Profile	S1 and S2 provide similar information written for a different audience.	The Durian, a fruit used in Asian cuisine, has a strong smell.	The dish is usually made with Durian.
17	Change of perspective	The same entity presents a differing opinion or presents a fact in a different light.	Giuliani criticized the Officer’s Union as “too demanding” in contract talks.	Giuliani praised the Officer’s Union, which provides legal aid and advice to members.
18	Overlap (partial equivalence)	S1 provides facts X and Y while S2 provides facts X and Z; X, Y, and Z should all be non-trivial.	The plane crashed into the 25th floor of the Pirelli building in downtown Milan.	A small tourist plane crashed into the tallest building in Milan.

Figure 1: CST relationships and examples

Cluster	Topic	Articles	Time span	Ave. length (sent.)	No. sources	Clustering method
Milan9	Milan plane crash	9	2 days	30	5	manual
DUC	John Lennon biography	4	4 years	46	4	manual
Gulfair11	Bahrain plane crash	11	4 days	27	6	manual
HKNews	Air and water quality	8	2.5 years	32	1	manual
NIE	N. Korea nuclear weapons	5	18 days	14	3	automatic
Novelty	Cancer and power lines	4	4 years	21	2	manual

Table 1: Characteristics of the document clusters

on the other five clusters and selected, from huge number of possible sentence pairs, a total of 4931 potentially "interesting" ones for human judges to annotate. This way the judges' workload is significantly reduced. Eight judges were hired; each judge annotated at least one cluster; each cluster was annotated by two judges. The judges were allowed to assign multiple CST types to a single sentence pair, given the inherently ambiguous nature of the problem and the fact that the CST types are not mutually exclusive.

Notice that the lexical similarity measure is merely a heuristic to filter out a large number of uninteresting sentence pairs. It is not sufficient to be a CST relationship identification algorithm by itself, because it can only work as a binary classifier, and its precision is very low (roughly 25% on the collected data).

5. CLASSIFICATION USING BOOSTING

The main goal of our study was to identify methods that can identify the presence of CST relationships in sentence pairs.

5.1 Algorithm and Features

Among a number of state-of-the-art classification algorithms, we choose to use boosting [20] for our task. The basic idea of this algorithm is to find a "strong" hypothesis by combining many "weak" or "base" hypotheses. Moreover, BoosTexter [20], the off-the-shelf implementation of boosting, explicitly supports multi-label classification, which is very convenient for the multi-label full classification scenario in our problem.

As discussed in section 2, lexical features by themselves are probably not sufficient for identifying CST relationships between sentences. Therefore, we considered various features at three linguistic levels, details of which follow. The general idea is to quantify the similarity or distance between two sentences at each level.

For most sentence pairs, the procedures for computing all features, such as tokenization, part-of-speech (POS) extraction, and head guessing, can directly or indirectly take advantage of the parse trees produced by the Charniak parser [1]. For the very few sentences on which the parser fails, we used heuristic backoff procedures.

5.1.1 Lexical features

At this level, we are only interested in the surface tokens. No stemming or stop-word deletion is done. Three features are of interest:

- Number of tokens in sentence 1
- Number of tokens in sentence 2
- Number of tokens in common

5.1.2 Syntactic-level features

At the syntactic level, we capture the overlap between two sentences with regard to 6 parts of speech: regular noun, proper noun, verb, adjective, adverb, and (cardinal) number, which are considered to convey relatively more substantial information than others.

For each x in the 6 POS types above, we compute the following counts:

- Number of tokens having POS x in sentence 1
- Number of tokens having POS x in sentence 2
- Number of common tokens having POS x

A total of 18 features are therefore used at this level.

5.1.3 Semantic-level features

Obviously, a full comparison of what two sentences "mean" at the semantic level is still an AI-complete problem, but here we propose a heuristic approximation. The idea is to find the most prominent concepts discussed in each sentence pair (by taking advantage of the syntactic structure) and compute their lexical semantic distance by using Wordnet [3]. More specifically, this is done through the following steps:

1. Find the top level NP (noun phrase) and VP (verb phrase) in sentence 1 and sentence 2.
2. Find the head tokens of both NP and VP by using the head rules in [10] and [2].
3. Align the heads correspondingly (i.e., NP vs. NP, VP vs. VP).
4. For each head pair, compute the semantic distance described in [8], [7], [17], [9], and [6], by using the semantic distance toolkit [15].

For each sentence pair, we have two pairs of heads (heads of NPs and heads of VPs). For each head pair, we compute the five semantic distance measures above. Therefore we have a total of 10 features at the semantic level.

For example, given the following (very simple) sentence pair:

```
(S(NP(NNS Birds))(VP(VBP fly)))
(S(NP(NNS Humans))(VP(VBP think)))
```

Five distance measures will be computed for word pairs {bird, human} and {fly, think} respectively. In this case, the heads of top-level NP and VP are trivial to find, but in most cases we have to resort to the fairly sophisticated rules in step 2 above.

5.2 Data treatment

As mentioned in section 4, each document cluster is annotated by two judges, and the judges are allowed to assign multiple labels to a single pair.

The judges don't always agree. They may either disagree on whether two sentences are CST-related at all or disagree on the types of CST relationships between them. Instead of asking the judges to resolve the disagreements, as was done in [5], we decided to only include the data points on which the two judges at least agree on the existence of CST relationships (regardless of type). 3942 out of 4931 sentence pairs satisfy this condition ($kappa = 0.53$). This is an important decision based on our understanding of the underlying linguistic phenomenon, instead of technical inability of dealing with noisy data. Since the ability to determine the existence of any CST relationships is important for many potential applications, we want the model to be as clean as possible. On the other hand, once the CST-relatedness is known, it is reasonable for multiple rhetorical relationships to hold between two sentences.

Given the constraint above, labels can be assigned to data points in the binary and full classification scenarios respectively:

- In the binary classification case, a label "1" is assigned to a pair if it is unanimously believed to be CST-related, and a label of "0" if it is unanimously believed to be not related. (These are the only two cases possible due to the constraint above.)
- In the full classification case, each sentence pair is assigned the union of the labels given by the two judges if they agree that the two sentences are CST-related, or a label "0" if they agree that the two sentences are not related. (Again, these are the only two cases possible due to the constraint above.)

The whole data set is then split into a training set, a dev-test set, and a test set by uniform random sampling without replacement in the proportion of 6:2:2.

5.3 Evaluation metrics

For binary classification, besides the standard classification accuracy, we also measure precision, recall, and F-measure as defined in the information retrieval literature [18].

For the multi-class classification, we compute the following aggregate metrics suggested by [20]:

- One-accuracy (whether the top-ranked label is among the correct ones)
- Coverage (how far do we have to go down the ranked label list to find all the correct ones)
- Average precision (analogous to non-interpolated average precision frequently used to evaluate document ranking performance)

We also measure precision, recall, and F-measure for each individual class label.

6. EXPERIMENTS AND RESULTS

Now we are ready to present the experimental results from both the binary case and the multi-class case. Notice that in both cases, the baseline strategy is to assign the label "0" (i.e., no CST relationship) to all data points, which achieves an accuracy of 75.13% on the test set.

6.1 Binary classification

For the experiments in this subsection, we trained a binary classifier that hypothesizes the existence of CST relationship(s), regardless of type, between a pair of sentences.

6.1.1 *Rounds of boosting*

An important parameter in boosting is the number of weak hypotheses. We first try to find the optimal number of rounds for boosting by optimizing the classification accuracy on the dev-test set (see figure 2).

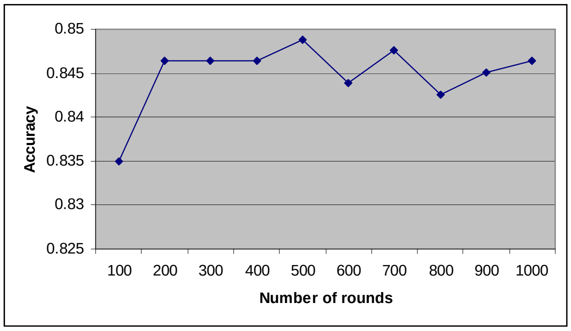

Figure 2: Dev-test performance vs. number of rounds (binary classification)

It turns out that the accuracy finds its maximum after around 500 boosting rounds. We then use this parameter to train the binary classifier and evaluate the performance of the resulting model on the test set.

6.1.2 *Effectiveness of features*

Features at different linguistic levels may play different roles in the classification problem. To test this hypothesis, we evaluated various performance metrics on the test set by training the classifier with three different sets of features. The results are summarized in Table 2.

By looking at the performance numbers in Table 2, one can notice that the classifier using the full feature vector (lexical, syntactic, and semantic features) wins by a big margin over the one using lexical features only. This is reasonable because the syntactic- and semantic-level features are supposed to provide valuable information about the topic and meaning of sentences. However, a little disappointing is the fact that the semantic features, which were expected to be useful, don't seem to help by themselves. We try to account for this issue in the discussion section.

6.2 Full classification

In this case, we are not only interested in the binary decision regarding the CST-relatedness of a sentence pair, but also the type(s) of relationship, if any. This is a multi-class multi-label problem.

6.2.1 *Rounds of boosting*

Similar to the binary case, we first try to find the optimal rounds of boosting by optimizing one-accuracy on the dev-test set (see figure 3).

This time the curve peaks at around 400 boosting rounds. We then use this parameter to train the full classifier and

Feature set	Accuracy	Precision	Recall	F-measure
Full feature vector	0.8789	0.8278	0.6477	0.7267
Lexical & syntactic features	0.8775	0.8141	0.6580	0.7278
Lexical Features only	0.8351	0.7560	0.4974	0.6000

Table 2: Performance of classifier trained on different sets of features (binary case)

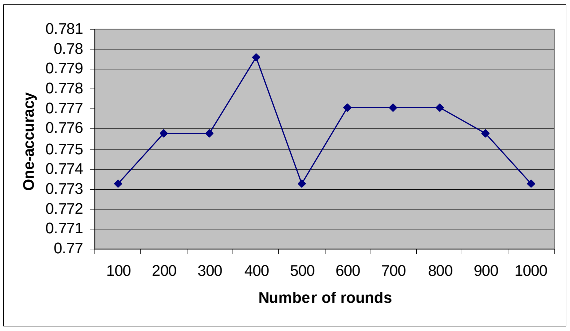

Figure 3: Dev-test performance vs. number of rounds (full classification)

CST type	Precision	Recall	F-measure
No relationship	0.8905	0.9485	0.9186
Equivalence	0.6000	0.2400	0.3429
Subsumption	0.0667	0.0417	0.0513
Follow-up	0.5088	0.3222	0.3946
Elaboration	0.4000	0.1795	0.2478
Description	0.2608	0.2143	0.2353
Overlap	0.5581	0.3529	0.4324

Table 4: Classifier performance on individual CST types

evaluate the performance of the resulting model on the test set.

6.2.2 *Effectiveness of features*

Similar to the binary case, again, we measured various performance metrics on the test set by training the classifier with three different sets of features. The result is summarized in Table 3.

A similar observation holds as in the binary case. This time the classifier without semantic features performs even slightly better than the one with full feature vector.

6.2.3 *Performance on individual relationship types*

We are also interested in the performance of the multi-class multi-label classifier on each individual CST type. Therefore we computed precision, recall, and F-measure for all CST types that occur more than 20 times in the test data (see Table 4).

As one can see, the classifier doesn't perform equally well on all CST types. It does a decent job on "equivalence", "follow-up", and "overlap"; but on the other hand, "subsumption" appears fairly hard to identify. For the types that don't appear in Table 4, the performance is inconclusive, since they don't occur frequently enough in the data.

6.3 Discussion

The performance of both classifiers is encouraging, however, the near-ineffectiveness of semantic features comes as a surprise. Our conjecture is that it can be accounted for by the following reasons:

- *Missing values*: when either word in a word pair cannot be found in Wordnet, the corresponding semantic distance features will be marked as missing values, except in the case of "hso" distance [6], where a value of 0 is given. In our problem, roughly half of the data points are subject to this issue. Although most machine learning algorithms, including boosting, can elegantly deal with missing values in general, features with too many missing values may be considered useless or even harmful.
- *Head guessing*: the lexical heads of top-level NP and VP may not sufficiently correlate with the key concepts. It might be helpful to investigate deeper representations such as the Prague Dependency Treebank structure [4].
- *Scope of semantic comparison*: comparing only heads of top-level NP and VP may not be sufficient. Sometimes the modifiers make a big difference in meaning even when the heads are the same. If we can come up with a way to align all content words (such as those having the 6 POS's discussed before) across two sentences, a wider scope of semantic differences will be computed.

We believe that the semantic-level features are very useful and merit further investigation, although they didn't show immediate success in the experiments above.

7. CONCLUSION AND FUTURE WORK

This paper describes an empirical study that uses boosting to classify cross-document structural relationships between sentence pairs extracted from various documents. We show that the binary classifier for determining existence of structural relationships significantly outperforms baseline, and promising results are also achieved on the multi-class case in which the full taxonomy of relationships are considered.

Looking into the future, on the one hand, there is plenty of room for improving the classifier performance, such as designing more sophisticated feature vectors (as discussed above) and experimenting with different machine learning algorithms. On the other hand, although it is somewhat reasonable to attribute the unbalanced performance of the full classifier on individual CST types to the inherent difference among the underlying linguistic phenomena, it may also suggest that the label system, i.e., the CST taxonomy,

Feature set	One-Accuracy	Coverage	Average precision
Full feature vector	0.8093	1.1070	0.8729
Lexical & syntactic features	0.8196	1.0722	0.8793
Lexical Features only	0.7809	1.1830	0.8618

Table 3: Performance of classifier trained on different sets of features (multi-class case)

needs more standardization. If humans have difficulty differentiating them, there is probably little hope for a machine learned classifier to work strikingly well.

The classification problem studied in this paper is still a somewhat simplified version of the full CST relationship identification problem. Some relationships have directionality, e.g., sentence 1 following-up sentence 2 is different from sentence 2 following-up sentence 1. To be able to address issues like this, more intelligence needs to be built into the CST identifier.

Another caveat is that the classifiers in the current experiments only look at "local" information within each sentence pair. In some cases, the "global" context plays an important role in determining the CST relationship(s) between two sentences.

In this paper, we have made the first attempt to show that automatic identification of CST relationships is feasible. Various IR-related applications may expect to see improvements by exploiting cross-document structure.

8. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0082884.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

[1] E. Charniak. A maximum-entropy-inspired parser. Technical Report CS-99-12, Computer Scicence Department, Brown University, 1999.

[2] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania, 1999.

[3] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 1998.

[4] J. Hajič. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In E. Hajičová, editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*, pages 12–19. Prague Karolinum, Charles University Press, 1998.

[5] V. Hatzivassiloglou, J. L. Klavans, M. L. Holcombe, R. Barzilay, M.-Y. Kan, and K. R. McKeown. Simfinder: A flexible clustering tool for summarization. In *NAACL Workshop on Text Summarization*, 2001.

[6] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet: An electronic lexical database*, pages 305–332. Cambridge, MA: The MIT Press, 1998.

[7] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the Int'l Conf. on Research on Computational Linguistics*, Taiwan, 1997.

[8] C. Leacock and M. Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. In C. Fellbaum, editor, *WordNet: An electronic lexical database*, pages 265–283. Cambridge, MA: The MIT Press, 1998.

[9] D. Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[10] D. Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 276–283, 1995.

[11] W. C. Mann and S. A. Thompson. Rhetorical Structure Theory: towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[12] D. Marcu. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts.* PhD thesis, Department of Computer Science, University of Toronto, December 1997.

[13] D. Marcu and A. Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 368–375, 2002.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, 2002.

[15] S. Patwardhan and T. Pedersen. distance.pl: Perl program that measures the semantic relatedness of words (version 0.11). http://www.d.umn.edu/ tpederse/distance.html, 2002.

[16] D. Radev. A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *Proceedings, 1st ACL SIGDIAL Workshop on Discourse and Dialogue*, Hong Kong, October 2000.

[17] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453, 1995.

[18] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM (JACM)*, 15(1):8–36, 1968.

[19] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *Information Processing & Management*, 33:193–207, 1997.

[20] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[21] Z. Zhang, S. Blair-Goldensohn, and D. Radev. Towards CST-enhanced summarization. In *Proceedings of the 18th National Conference on Artificial Intelligence*, Edmonton, Alberta, August 2002.