International Conference on Intelligent Text Processing and Computational Linguistics

CICLing 2011: Computational Linguistics and Intelligent Text Processing pp 328-339 | Cite as

Comparing Approaches to Tag Discourse Relations

Authors | Authors and affiliations

Shamima Mithun, Leila Kosseim

Part of the Lecture Notes in Computer Science book series (LNCS, volume 6608)

Abstract

It is widely accepted that in a text, sentences and clauses cannot be understood in isolation but ...ation with each other through discourse relations that may or may not be explicitly

Comparing Approaches to Tag Discourse Relations

Shamima Mithun and Leila Kosseim

Concordia University
Department of Computer Science and Software Engineering
Montreal, Quebec, Canada
{s_mithun, kosseim}@encs.concordia.ca

Abstract. It is widely accepted that in a text, sentences and clauses cannot be understood in isolation but in relation with each other through discourse relations that may or may not be explicitly marked. Discourse relations have been found useful in many applications such as machine translation, text summarization, and question answering; however, they are often not considered in computational language applications because domain and genre independent robust discourse parsers are very few. In this paper, we analyze existing approaches to identify five discourse relations automatically (namely, *comparison, contingency, illustration, attribution*, and *topic-opinion*), and propose a new approach to identify *attributive* relations. We evaluate the accuracy of each approach with respect to the discourse relations it can identify and compare it to a human gold standard. The evaluation results show that the state of the art systems are rather effective at identifying most of the relations considered, but other relations such as *attribution* are still not identified with high accuracy.

1 Introduction

It is widely accepted that sentences and clauses in a text cannot be understood in isolation but in relation with each other. A text is not a linear combination of clauses but a hierarchial organized group of clauses placed together based on informational and interactional relations to one another. For example, in the sentence *"If you want the full Vista experience, you'll want a heavy system and graphics hardware, and lots of memory"*, the first and second clauses do not bear much meaning independently; they become more meaningful when we realize that they are related through the discourse relation *condition*.

In a discourse, different kinds of relations such as *contrast*, *causality*, *elaboration* may be expressed. The use of such discourse structures modelled by rhetorical predicates (described in section 2) have been found useful in many applications such as document summarization and question answering ([9, 7]). For example, [9] showed that rhetorical predicates can be used to select the content and generate coherent text in question answering with the help of schemata. Recently, [10] has demonstrated that rhetorical predicates can be useful in blog summarization. Rhetorical predicates have also been found useful for anaphora resolution [9] and machine translation [11].

Though rhetorical predicates are useful in many applications, their automatic identification remains a challenging task. Existing rhetorical predicate identification approaches (e.g. [9, 11]) are often domain or genre dependent. For example, in [9], predicates are identified based on the hierarchical structures and prestored relations in a knowledge base. In certain sub-languages, predicates are often identified by means of key words and other linguistic clues (e.g. *because*, *if*, *then*) or through verb frameworks [11]. With verb frameworks, characteristics of a verb are defined for the specified sub-language and each verb is associated with possible rhetorical predicates. [11] also used domain knowledge with verb frameworks to identify predicates.

In this paper, we focus on genre and domain independent intra-sentential rhetorical predicates identification approaches which can tag individual rhetorical predicates as opposed to performing a more complete discourse parse. Only intra-sentence predicates are considered because in many applications such as extractive summarization, question answering, and information retrieval, individual sentences are extracted from different documents or from different positions of a document to build a candidate sentence list. As a result, there is very little chance that inter-sentential relations will exist among candidate sentences. On the other hand, intra-sentential relations have already been found useful to organize texts and select content by utilizing schema in summarization and question answering [9, 10, 1]. Intra-sentential relations may enable a system to answer non-factoid questions such as “Why do people like Picasa?” by selecting clauses related through a causality; and [1] showed that 95% of the time, causality occurred within sentences in the corpus T (a gigaword newswire corpus of 4.7 million newswire documents[1]).

In this paper, we first introduce the set of rhetorical predicates which we have taken into consideration. Then we present different available approaches such as the SPADE parser [13], Jindal et al.'s [5] work, and Fei et al.'s [3] work that can be used to identify these rhetorical predicates. We have also developed an approach to identify the *attributive* predicate. We then evaluate the performance of each of these approaches using precision, recall, and F-Measure. We have also developed gold standards for the identification of each predicate to evaluate the effectiveness of these approaches. The evaluation results show that the current state of the art is acceptable to identify some predicates (e.g. *illustration*) but not others (e.g. *attribution*).

2 Rhetorical Predicates

Rhetorical predicates are the means which a speaker has to describe information. Rhetorical predicates describe different predicating acts a speaker can use

[1] Distributed by the Linguistic Data Consortium, http://www.ldc.upenn.edu

and describe the structural relations between clauses in a text. Some examples are *constituency* (that provides details about sub-parts), and *attributive* (that provides details about an entity or object).

Rhetorical predicates take clauses as arguments. Clauses represent the smallest units that stand in informational or interactional relationship with other parts of texts. In this framework, clauses are classified into rhetorical predicates based on their underlying information. Rhetorical predicates classify clauses into two broad categories:

1. *A clause that contains a relation with another clause.*
2. *A clause that provides information on its own.*

In the first case, rhetorical predicates describe the relation between clauses and thus express the relationship that unite them (e.g. the *evidence* predicate creates a relation with the stated fact in order to provide support) [9]. In the second case, rhetorical predicates characterize the structural purpose of a clause (e.g. the *attributive* predicate can describe the attribute of an object). Here, a single clause can characterize a predicate. This kind of discourse structure is not considered by most of the discourse theories except rhetorical predicates.

Our work was performed within the framework of developing a query-based summarizer for blogs. Hence, we considered the predicates that were most useful for this application [10]. We considered six types of rhetorical predicates, namely *comparison*, *contingency*, *illustration*, *attribution*, *topic-opinion*, and *attributive*. The *comparison*, *contingency*, *illustration*, and *attribution* predicates are also considered by most of the work in the field of discourse such as the PDTB research group [12] and [2]. We considered two additional classes of predicates: *attributive* and *topic-opinion*.

The *attributive* predicate, also included in Grimes' predicates [4], is considered because it describes attributes or features of an object or event and is often used in query-based summarization and question answering. We introduced the *topic-opinion* predicate because by analyzing the TAC-2008 corpus[2], we have found that the discourse structures (e.g. feelings, thoughts) captured by this predicate are often used in opinionated texts. In building our predicate model, we considered all main discourse structures listed in Mann and Thompson's Rhetorical Structure Theory (RST) taxonomy [6]. These discourse structures are also considered in Grimes' and Williams' predicate lists [9]. Description of these rhetorical predicates are given below:

1. **Comparison**: Gives a comparison and contrast among different situations - e.g. *Perhaps that's why for my European taste Starbucks makes great espresso while Dunkin's stinks.* The *comparison* predicates also subsume the *contrast*, *analogy*, and *preference* predicates.

2. **Contingency**: Provides cause, condition, reason, evidence for a situation, result or claim - e.g. *The meat is good because they slice it right in front of*

[2] http://www.nist.gov/tac

you. The *contingency* predicate subsumes the *explanation*, *evidence*, *reason*, *cause*, *result*, *consequence*, *background*, *condition*, *hypothetical*, *enablement*, and *purpose* predicates.

3. **Illustration**: Is used to provide additional information or detail about a situation - e.g. *I have a special relationship with the lovely people who work in the Dunkin' Donuts in the Harvard Square T Station in Cambridge.* The *joint*, *list*, *disjoint*, and *elaboration* predicates are subclasses of the *illustration* predicate.

4. **Attribution**: Is used to convey reported speech both direct and indirect. This predicate can also be used to express feelings, thoughts, or hopes - e.g. *I said actually I think Zillow is great.*

5. **Topic-Opinion**: Can be used to express an opinion on a specific topic; an agent can express internal feeling or belief towards an object or an event - e.g. *The thing I love about their sandwiches is the bread.*

6. **Attributive**: Provides details about an entity or an event. It can be used to illustrate a particular feature about a concept - e.g. *Mary has a pink coat.*

As stated earlier, our study focused only on these predicates but other predicates would also be interesting to consider (e.g. *antithesis*).

3 Discourse Tagging

Several approaches to automatically identify the predicates described above have been proposed; the most notable ones are: the SPADE parser [13], Jindal et al.'s approach [5], and Fei et al.'s approach [3].

3.1 The SPADE Parser

The SPADE parser [13] was developed within the framework of RST. In SPADE, a large number of fine grained discourse relations are considered compared to those in RST. The SPADE parser identifies discourse relations within a sentence by first identifying elementary discourse units (EDU)s, then identifying discourse relations between two EDUs (clauses) by following the RST theory. For example, in the sentence below, the SPADE parser identifies two clauses:

a. *[Perhaps that's why for my European taste Starbucks makes great espresso]*
b. *[while Dunkin's stinks.]*

and assigns the relation *contrast* between these two clauses.

The parser consists of two components: the *discourse segmenter* and the *discourse parser*. The *discourse segmenter* divides sentences into clauses. It uses two components for this purpose namely a *statistical model*, which assigns a probability to the insertion of a discourse boundary after each word in the sentence, and a *segmenter* which uses the probabilities computed by the model for inserting discourse boundaries. Given a sentence, this model first finds the syntactic parse tree of the sentence. Then using both lexical and syntactic features of the parse tree it determines a probability of inserting a discourse boundary. Once the discourse boundaries of a sentence are determined the *discourse parser* creates a discourse tree for the sentence. The *discourse parser* also consists of two components: a *parsing model*, which assigns a probability to every potential candidate parse tree, and the *discourse parser*, which is an algorithm for finding the best discourse tree. To find the best discourse tree, it implements a bottom-up algorithm which searches through the space of all possible discourse trees using dynamic programming. In this process, between two clauses if more than one discourse relation is available then the relation with the highest probability score (that is calculated based on their syntactic and lexical information from the training corpus) is selected.

The SPADE parser can only identify discourse structures across clauses, and cannot identify those occurring within a clause. For example, in "*Dunkin Donuts' coffee tasted better than Starbucks*" a *comparison* structure is used, but would not be identified by SPADE. However, in our analysis, we found that *comparisons*, *topic-opinion*, and *attributive* do occur within a clause. To identify these kinds of structures, the taggers described in the next sections were considered.

3.2 Jindal et al.'s Approach

In order to label a clause as containing a *comparison* predicate, Jindal et al.'s approach [5] can be used. In this approach, using a set of keywords and annotated texts, the classifier first generates patterns for comparison sentence mining called sequences.

To build the sequence database, the classifier first considers the sentences which contain at least one predefined keyword. Then it creates a sequence using words which occur within a window of 3 words around the keyword. In the next step, these words are replaced with their part of speech (POS) tag and a class is associated with the sequence based on whether this sentence is a comparison or non-comparison sentence. For example, the sentence "With/IN Carmax/NNP you/PRP will/MD generally/RB always/RB pay/VB **more**/RBR than/IN from/IN going/VBG to/TO a/DT good/JJ used/VBN car/NN dealer/NN" contains the keyword "more" and the sequence will be stored in the database :

({RB}{RB}{VB}{more/RBR}{IN}{IN}{VBG}) *comparison*

After the database is constructed, class sequential rules (CSR) are generated. A CSR is a rule with sequences on the left and a class label on the right of the rule. The CSR rules are generated by combining sequences which are available in the sequence database. As CSR, those rules are accepted which meet the pre-specified *support* and *confidence* threshold value. The support and confidence of a rule are defined as follows:

$$Support\ of\ a\ rule = \frac{\#\ of\ instances\ containing\ this\ rule}{\#\ of\ instances\ in\ the\ sequence\ database}$$
$$Confidence\ of\ a\ rule = \frac{\#\ of\ instances\ containing\ this\ rule\ in\ this\ class}{\#\ of\ instances\ in\ the\ sequence\ database\ satisfying\ the\ rule}$$

A Naïve Bayes classifier is used using the CSR patterns as features to learn a 2-class classifier (comparison and non-comparison). To evaluate their approach, we have developed a classifier to identify the *comparison* predicate using their annotated dataset (see section 4).

3.3 Fei et al.'s Approach

The *topic-opinion* predicate indicates whether a sentence expresses an opinion towards a specific topic. Fei et al. [3] showed that the dependency relations of words defined by a dependency grammar are useful to find relations between a topic and subjective words.

Dependency relations refer the binary relations between two words where in this binary relation one word is the parent and the other word is the child. In this representation, one word can be associated with only one parent but with many children. In this way, when we create the dependency relations of a sentence it will be in a tree form (called a dependency tree). These dependency relations are useful to find relations (links) between subjective words and a topic. Different words of a sentence can be related using the dependency relations directly or based on the transitivity of these relations. For example, the dependency relation of the sentence *"Subway has bad food."* is shown in Figure 1.

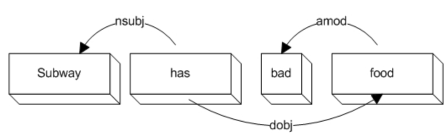

Fig. 1. Dependency Relations

The head of the arrow directs the child, the tail comes from the parent, and the tag of the arrow shows the dependency relation type. From Figure 1, we can see that both words *Subway* and *food* are children of the word *has*. The word *bad* and *food* are directly related using the dependency relation `amod`. *Subway* and *bad* are related based on the transitivity of the relations. With the help of dependency relations we can find that the topic *Subway* and the subjective word *bad* are related. Fei et al. [3] used 3 instances of dependency relations (shown in Figure 2) for opinion mining:

1. **Subjective Words that are Descendant of the Topic:** To identify whether subjective words (S-word) are descendent of the topic, subjective words should be in the modifier relation with the topic directly or based on some transitivity relations.

2. **Subjective Words and the Topic that have the Common Ancestor:** Under this category, [3] accepted instances where the same ancestor is the verb.
3. **Subjective Words that are Ancestors of the Topic:** To classify under this category according to [3], the subjective word needs to be a verb, and the topic needs to be in the subject or object of the verbs.

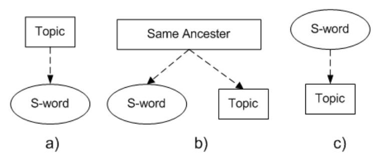

Fig. 2. Topic-opinion Dependency Relations Tree

In order to evaluate their approach, we have also built a tagger by following Fei et al.'s approach (see section 4).

3.4 Our Attributive Tagger

To our knowledge, no previous work has focused on tagging the *attributive* predicate. Hence, to identify these predicates, which typically occur within a clause, we developed our own tagger. Similarly to Fei et al.'s [3] work, we used dependency relations of words (using the Stanford parser[3]) to develop this classifier.

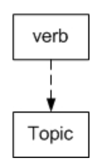

Fig. 3. Attributive Dependency Relations Tree

By analyzing TAC 2008 data, we have found that the dependency relation shown in Figure 3 can be used to identify the *attributive* predicate. From this figure, we can see that to be classified as an *attributive* predicate, the topic needs to be the descendant of a verb; however, the topic need not necessarily be directly related to the verb. From our TAC data analysis, we have also found that to classify as an *attributive* predicate, the topic needs to be in subject or object relation of the verb. We have devised a set of 5 heuristic rules by analyzing datasets containing 200 attributive sentences (sentences from TAC-2008). For example, in the sentence *"Picasa displays the zoom percentage"* there will be a dependency relation `nsubj` between the topic "*picasa*" and the verb "*displays*"(shown in Figure 4).

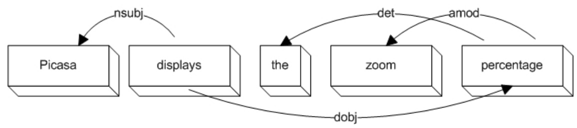

Fig. 4. Attributive Dependency Relations Example

The 5 heuristics rules are:
1. The *verb* is directly associated with the *topic* using the dependency relation `nsubj`.
2. The *verb* is associated with a *noun* using the `nsubj` relation and that *noun* is linked with the *topic* using the dependency relation `nn`.
3. The *verb* is associated with a *noun* using the `nsubj` relation and that *noun* is linked with the *topic* using the dependency relation `prep`.
4. The *verb* is associated with a *noun* using the `nsubj` relation and that *noun* is linked with the *topic* using the dependency relation `poss`.
5. The *verb* is directly associated with the *topic* using the dependency relation `dobj`.

4 Evaluation

This section describes the corpora and the evaluation results of the predicate taggers described above. This section also provides a comparison with a baseline and gold standard for each predicate.

4.1 Corpora

To evaluate the performance of the taggers, four different corpora have been used. The descriptions of these corpora are given below:

The SPADE Parser Corpus
To evaluate the SPADE parser, the publicly available RST Discourse Treebank 2002[4], which contains 385 Wall Street Journal articles from the Penn Treebank, was used. The dataset is divided into a training set of 347 articles (6132 sentences) and a testing set of 38 articles (991 sentences). In the corpus, for each document, a discourse tree was manually created by following Rhetorical Structure Theory (RST). In the evaluation, only discourse subtrees over individual sentences were utilized.

The Comparative Corpus
To evaluate the comparative classifier, the dataset developed by [5] was used. This corpus consists of 905 comparative and 4985 non-comparative sentences.

[3] available at nlp.stanford.edu/software/stanford-dependencies.shtml
[4] Distributed by the Linguistic Data Consortium (http://www.ldc.upenn)

Four human annotators labelled these data manually. This dataset consists of reviews, forum, and news articles from different sources.

The Topic-opinion Corpus

To evaluate the topic-opinion classifier, the corpus developed by [3] from the polarity dataset[5] was used. The polarity dataset originally includes 1000 positive and 1000 negative reviews on films. From this polarity dataset, [3] have randomly annotated 400 sentences that contain both film terms and opinionated expressions from General Inquirer terms[6]. In this corpus of 400 sentences, in 262 sentences opinions are attached to the target. To annotate this corpus, 86 popular film terms from the dataset and online film glossary[7] were collected.

The Attributive Corpus

Since no standard dataset was available for the *attributive* predicates, we have manually created a corpus of 400 sentences from the TAC 2008 opinion summarization track. This corpus consists of 200 attributive sentences and 200 non-attributive sentences.

4.2 Results

For the evaluation, each approach was evaluated with its associated dataset and the performance was evaluated using precision, recall, and F-Measure scores. The SPADE parser's performance was evaluated on 18 discourse relations identification. On the other hand, the performance evaluation of all other classifiers was binary (e.g. attributive versus non-attributive).

Table 1. Performance of Different Predicate Identification Approaches

Rhetorical Predicate	Clause Level	Classifier	Precision	Recall	F-Measure
Comparison	Inter	SPADE	58%	31%	40%
Comparison	Intra	Jindal et al.'s	77%	81%	79%
		Authors'	66%	68%	67%
Contingency	Inter	SPADE	85%	76%	80%
Illustration	Inter	SPADE	79%	93%	85%
Attribution	Inter	SPADE	52%	83%	64%
Topic-opinion	Intra	Fei et al.'s	75%	66%	70%
		Authors'	66%	68%	67%
Attributive	Intra	Authors'	77%	76%	77%

Table 1 shows the results of the evaluation. The table indicates: a) the rhetorical predicates which have been identified; b) at what level these predicates occurred (within a clause or across two clauses); c) which classifier is used to

[5] http://www.cs.cornell.edu/people/pabo/movie-review-data
[6] http://www.wjh.havard.edu/∼ inquirer
[7] http://www.filmsite.org/filmterms.html

identify the specified predicate; d) the evaluation results using precision, recall, and F-Measure.

From Table 1, we can see that to identify inter-clause *comparison*, *contingency*, *illustration*, and *attribution* predicates, the SPADE parser is used. As the evaluation of the SPADE parser was executed on 18 label relations and the performance for a specific predicate identification is not mentioned in [13], we have computed ourselves the performance of the SPADE parser for *contingency*, *comparison*, *illustration*, and *attribution* predicates using the same corpus used by [13]. The performance of the SPADE parser to identify each of these predicates is shown in Table 1. The table shows the evaluation results of Jindal et al.'s approach (as published in [5]) and our implementation (authors') of their approach to identify the *comparison* predicate which occur within a clause. Table 1 also shows the evaluation results of Fei et al.'s approach (as published in [3]) and our implementation of their approach. Table 1 also shows the evaluation results of our approach to identify the *attributive* predicate.

Table 2. Baseline and Gold Standard Performance

		Baseline			**Human Performance**		
Rhetorical Predicate	**Clause Level**	**P**	**R**	**F**	**P**	**R**	**F**
Comparison, Contingency, Illustration, Attribution	Inter	unknown	unknown	23%	unknown	unknown	77%
Comparison	Intra	94%	32%	48%	91%	86%	89%
Topic-opinion	Intra	70%	21%	32%	77%	77%	77%
Attributive	Intra	39%	67%	49%	79%	88%	83%

Table 2 shows the baseline and gold standard performance for identifying these rhetorical predicates using precision (P), recall (R), and F-Measure (F). The baseline and gold standard figures were computed as described below:

Baselines:

Inter-Comparison, Contingency, Illustration, Attribution: The SPADE baseline described in [13] was used. The baseline algorithm builds right branching discourse tree and labels with the most frequent relation learned from the training set.

***Intra-Comparison*:** The baseline algorithm considers a sentence as a comparison if it contains any of the keywords of Jindal et al. [5].

***Topic-opinion*:** Following [3], the baseline algorithm considers sentences as *topic-opinion* if they follow one of the two patterns below:

(RB)+JJ+(NN)+Target; ((RB)+JJ)+NN+Target

where, RB, JJ, and NN are part of speech (adverb, adjective, noun) and Target is the topic of the sentence.

***Attributive*:** To be considered as an *attributive* predicate, the topic of the sentence needs to be associated with the verb using the dependency relation `subj`.

Gold Standards:
Inter-Comparison, Contingency, Illustration, and Attribution: The gold standard of [13] was used. It is computed as the agreement between two human annotators who independently annotated 53 articles of the RST Discourse Treebank corpus.

Intra-Comparison, Topic-opinion, and Attributive: The gold standards are computed as the agreement between two human annotators who annotated 100 sentences of the comparative, the topic-opinion, and the attributive corpus for each rhetorical predicate.

4.3 Analysis

In general, the state of the art approaches do much better at tagging rhetorical predicates compared to the baseline and do respectably well compared to the gold standard. As Table 1 shows, currently, the state of the art systems have difficulty tagging the rhetorical predicate *topic-opinion* - achieving an F-Measure of 70%. However, the gold standard is also very low (75%), leading us to believe that this predicate is hard to identify. The reason behind this could be it may not be marked explicitly in the text, or may be marked in a variety of ways. Moreover, sentiment identification, which is a sub-task of *topic-opinion* predicate tagging, is a complex task on its own. As a result, the F-Measure scores of the *attribution* predicate tagging, which also requires sentiment analysis, is also low. On the other hand, the rhetorical predicate *intra-comparison* is tagged satisfactorily by the state of the art systems, and the gold standard is high too. We believe that this rhetorical predicate is more explicitly marked linguistically and in a more stereotypical manner.

5 Conclusion and Future Work

In our work, we have identified a set of intra-sentential rhetorical predicates which can be expressed in factoid or opinionated texts and have analyzed domain and genre independent automatic approaches to identify these rhetorical predicates. We tried to use off-the-shelf approaches which have been developed for discourse analysis or for other purposes to identify intra-sentential discourse structures. In addition, we have introduced an automatic approach to identify the *attributive* predicate based on dependency relations. As a gold standard to evaluate the tagging of each predicate was not available, we have developed one and have used it to compare the performance of various approaches. The evaluation shows that these approaches are effective to identify some discourse structures (e.g. *illustration*) compared to others (e.g. *attribution*).

As the performance of our comparative and topic-opinion classifier is not very satisfactory, in the future we plan to conduct a manual analysis to find out why. To analyze our topic-opinion classifier's performance, we plan to evaluate its accuracy in sentiment analysis. In the future, we also plan to evaluate the usability of rhetorical predicate tagging for summarization.

6 Acknowledgement

The authors would like to thank the anonymous reviewers for their comments and Soricut et al. [13], Jindal et al. [5], and Fei et al.[3] for making their resources available.

This work was financially supported by NSERC.

References

[1] Blair-Goldensohn, Sasha J.: Long-answer Question Answering and Rhetorical-semantic Relations. Department of Computer Science, Columbia University, 2007, Columbia, USA.

[2] Carlson, L., Marcu, D.: Discourse Tagging Reference Manual. University of Southern California Information Sciences Institute, ISI-TR-545, 2001.

[3] Fei, Z., Huang, X., Wu, L.: Mining the Relation between Sentiment Expression and Target Using Dependency of Words. *PACLIC20: Coling 2006: Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation.* 257–264 (2006), Wuhan, China, November.

[4] Grimes, J.E.: The Thread of Discourse. Cornell University, NSF-TR-1, NSF-GS-3180, 1972, Ithaca, New York.

[5] Jindal, N., Liu, B.: Identifying Comparative Sentences in Text Documents. *In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 244–251 (2006), Seattle, USA.

[6] Mann, W.C., Thompson, S. A.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organisation. *J. Text*, 3(8):234–281, 1988.

[7] Marcu, D.: From Discourse Structures to Text Summaries. *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization.* 1997, 82–88, Madrid, Spain, July.

[8] Marcu, D., Echihabi, A.: An Unsupervised Approach to Recognizing Discourse Relations. *ACL '02: In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.* 368–375 (2002), Philadelphia, Pennsylvania, July.

[9] McKeown, K.R.: Discourse Strategies for Generating Natural-Language Text. *J. Artificial Intelligence*, 27(1):1–41, 1985.

[10] Mithun, S., Kosseim, L.: A Hybrid Approach to Utilize Rhetorical Relations for Blog Summarization. *In Proceedings of TALN 2010.* July 2010, Montreal, Canada.

[11] Mitkov, R.: How Could Rhetorical Relations be used in Machine Translation? (And at Least Two Open Questions). *Workshop on Intentionality And Structure In Discource Relations.* 86–89 (1993), Ohio State University Columbus, Ohio, June.

[12] Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., Webber, B.: The Penn Discourse Treebank 2.0. Annotation Manual. Institute for Research in Cognitive Science, University of Pennsylvania, IRCS-08-01, 2008.

[13] Soricut, R., Marcu, D.: Sentence Level Discourse Parsing using Syntactic and Lexical Information. *NAACL '03: In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology.* 149–156 (2003), Edmonton, Canada.