Caroline Sporleder

Lexical Models to Identify Unmarked Discourse Relations: Does WordNet help?

Abstract

In this paper, we address the task of automatically determining which discourse relation holds between two text spans. We focus on relations that are not explicitly signalled by a discourse marker like *but*. While lexical models have been found useful for the task, they are also prone to data sparseness problems, which is a big drawback given the scarcity of discourse annotated data. We therefore investigate whether the use of lexical-semantic resources, such as WordNet, can be exploited to back-off to a more general representation of lexical information in cases were data are sparse. We compare such a semantic back-off strategy to morphological generalisations over word forms, such as stemming and lemmatising.

1 Introduction

To be able to interpret a text it is important to know how its sentences and clauses relate to each other. For example, whether the events referred to stand in a causal relation or whether one text segment provides an elaboration or a summary of another. This type of information is also crucial for many natural language processing (NLP) tasks. Question answering, for instance, frequently involves recognising cause and effect, e.g., to answer questions like "*Why did Romano Prodi resign?*" or "*What is the effect of Benzodiazepines in elderly people?*". Likewise, text summarisation systems need to know which pieces of information in a text are essential and which ones merely elaborate.

While there has been a considerable research effort dedicated to the automatic identification of discourse relations between text segments, the problem is still far from being solved, with state-of-the-art systems typically obtaining F-Scores between 40% and 70%, depending on the exact task and the number of discourse relations considered (see, e.g., Marcu (2000); Soricut and Marcu (2003); Le Thanh et al. (2004); Pardo et al. (2004); Baldridge and Lascarides (2005); Baldridge et al. (2007)). Moreover, most approaches heavily rely on surface cues, especially the presence of overt discourse markers such as *because* or *but*. Few systems have been dedicated to determine relations *in the absence* of such markers.[1] However, it has been estimated that only around half of all relations are explicitly signalled by a discourse connective (Redeker (1990); Eugenio et al. (1997); Marcu (2000)). Connectives are also often ambiguous, either between discourse usage and non-discourse usage (e.g., *for* as a synonym of *because* vs. *for* as a preposition) or be-

[1] A notable exception are Marcu and Echihabi (2002).

tween two or more discourse relations (e.g., *since* can signal a temporal or an explanation relation). Effectively, one can distinguish three, progressively more difficult, cases:

1. a relation is signalled by an **unambiguous marker**
2. a relation is signalled by an **ambiguous marker**
3. a relation is **not explicitly signalled** by any marker

Relations falling in the first set can be trivially identified provided one has a list mapping unambiguous markers to the relations they signal.[2] For the second case, discourse markers have to be disambiguated. For the third case, relations need to be identified based on other cues, such as the lexical semantics of the words in the sentences. The performance on the third task is likely to be much lower than the F-Scores of 40%-70% reported above for systems that address all types of relations. Identifying discourse relations which are not signalled by explicit discourse markers is thus one of the main bottlenecks for the automatic determination of discourse structure.

In this paper, we focus specifically on distinguishing *unmarked* discourse relations, which we define as covering both, relations which are signalled by ambiguous markers (case two above) and relations which are not signalled by any discourse markers (case three). The reason for not distinguishing between these two cases is that it is sometimes difficult to tell whether a relation is ambiguously signalled or not at all; some discourse connectives, such as *and*, are so ambiguous with respect to the relations they can signal that they supply hardly any discourse information at all.

While we do not aim at *solving* the task of recognising unmarked relations in this paper, we intend to shed some light on *lexical cues* that can or cannot help to identify such relations. Intuitively, lexical information provides useful cues for this task, as the correct discourse relation can often be guessed on the basis of the lexical semantics of the words involved. For instance, the two spans (marked by square brackets) in example (1) are related by EXPLANATION and a human may already be able to infer this from the words *late* and *missed the bus* alone. Likewise, the CONTRAST relation in example (2) (taken from Marcu and Echihabi (2002)) can be guessed from the presence of the two words *good* and *fail* which indicate a contrast. Similarly, in example (3) the SUMMARY relation might be inferable from the occurrence of *expensive* and *$7,000* in the left and right spans, respectively.

(1) [Peter was late this morning,] [he had missed the bus.]
(EXPLANATION)

(2) [Paul is *good* in maths and sciences.] [Peter *fails* almost every class he takes.]
(CONTRAST)

[2] The set of unambiguous markers depends to some extent on the discourse theory that is used. For example *in other words* can signal either RESTATEMENT or SUMMARY in *Rhetorical Structure Theory* (RST, Mann and Thompson (1987)), whereas it unambiguously signals SUMMARY in *Segmented Discourse Representation Theory* (SDRT, Asher and Lascarides (2003)) because the latter theory does not distinguish these two relations.

(3) ["It may be very *expensive*," the spokesman warned.] ["The price cannot be less than *$7,000*."]
(SUMMARY)

Empirical evidence for the importance of lexical information for identifying discourse relations has also been provided by a number of previous studies. Virtually all data-driven approaches to discourse parsing employ some lexical information to determine discourse relations. Polanyi et al. (2004), for example, make use of information about lexeme repetition and synonym, antonym, and hypernym relationships between lexemes, in addition to other cues (syntactic and structural information) to determine discourse relations. Forbes et al. (2001) rely heavily on lexicalised tree fragments to derive discourse structure. Likewise Soricut and Marcu (2003) propose a lexicalised discourse parser. Le Thanh et al. (2004) exploit lexical and syntactic cues to build their discourse trees. The system by Pardo et al. (2004) is completely based on surface cues and does not require syntactic information, relying solely on discourse markers and cue words. Similarly, Marcu and Echihabi (2002) determine the discourse relations holding between two spans solely on the basis of the words occurring in the spans. Finally, Sporleder and Lascarides (2005) found that lexical cues were among the best performing features in their multi-feature system for determining discourse relations.

While lexical cues can contribute in identifying the correct discourse relation in some examples, lexical cues also tend to be prone to data sparseness. The reliable learning of a mapping from lexical properties to discourse relations typically requires a very large amount of annotated data for training. Unfortunately, the training sets available are normally fairly small as annotated data is expensive to create. Marcu and Echihabi (2002) proposed to address the lack of training data by automatically creating labelled data from unannotated corpora. For this, they extracted unambiguously marked examples from a corpus, labelled them with the relation signalled by the marker, then removed the marker and trained a lexical model to recognise discourse relations in the absence of any marker. However, their approach was found not to generalise very well to naturally unmarked instances (Murray et al., 2006; Blair-Goldensohn et al., 2007; Sporleder and Lascarides, 2008).

An alternative to increasing the annotated data by automatic example labelling is to look for a representation of lexical information that is less prone to sparse data problems. For NLP tasks such as prepositional phrase attachment (Clark and Weir, 2000) or compound noun analysis (Nastase et al., 2006), it has been suggested to replace individual lexical items by more general classes, such as hypernyms taken from WordNet (Miller et al., 1990), in order to overcome data sparseness. In this paper, we investigate whether class-based information is also useful for identifying discourse relations and how this strategy compares to other methods of generalising over the actual word forms, such as lemmatising or stemming.

2 Experimental Set-up

To determine which of the generalisation strategies performs best, we first created a data set of pairs of text spans which are linked by unmarked discourse relations. We then created a number of two-feature classifiers, in which one feature encoded information about the left span at a given level of generalisation and the second feature encoded the same type of information for the right span. For example, the first feature might encode the stems in the left span and the second the stems in the right span. To determine the utility of each feature type, we ran a 10-fold cross-validation experiment for each of the classifiers in isolation. We also assessed the data sparseness that resulted from a particular encoding of the spans.

The next section describes the data creation in more detail. Section 2.2 outlines the machine learning framework we employed and 2.3 lists the individual features we tested.

2.1 Data

For our experiments, we looked at five relations from *Segmented Discourse Representation Theory* (SDRT, Asher and Lascarides (2003)): CONTRAST, EXPLANATION, RESULT, SUMMARY, and CONTINUATION. SDRT relations tend to be more coarsely-grained than those used by Rhetorical Structure Theory (RST, (Mann and Thompson, 1987)) and are therefore more amenable to automatic analysis. Examples of the five relations are given below (examples 4 to 8). For a detailed definition of each of the relations see Asher and Lascarides (2003).

(4) [The executive said any buy-out would be led by the current board, whose chairman is Maurice Saatchi and whose strategic guiding force is believed to be Charles Saatchi.]
[Mr. Spielvogel isn't part of the board, nor are any of the other heads of Saatchi's big U.S.-based ad agencies.]
(CONTRAST)

(5) [The five astronauts returned to Earth about three hours early because high winds had been predicted at the landing site.]
[Fog shrouded the base before touchdown.]
(CONTINUATION)

(6) [The venture's importance for Thomson is great.]
[Thomson feels the future of its defense business depends on building cooperation with other Europeans.]
(EXPLANATION)

(7) [A broker may have to approach as many as 20 underwriters who insure the endeavors on behalf of the syndicates.]
[It could take six months for a claim to be paid.]
(RESULT)

(8) ["It will be very expensive," the spokesman warned.]
["The price cannot be less than $7,000."]
(SUMMARY)

To create the data set, we collected examples from the RST Discourse Treebank (RST-DT, Carlson et al. (2002)) and manually mapped them to SDRT relations (see Sporleder and Lascarides (2008) for details). We only extracted examples in which the relation is not signalled by an unambiguous discourse marker and in which the relation holds between the clauses of a sentence or between adjacent sentences; we did not collect relations holding between multi-sentence text spans.[3] Overall, our final data set contained 1,051 examples, with roughly equal proportions of all relations with the exception of SUMMARY for which we found only 44 examples in the RST-DT (see Table 1). The inter-annotator agreement for identifying the relations was 72% (kappa .592, Carletta (1996)). This is effectively an upper bound on the performance we can expect from automatic relation classifiers. The fact that the agreement is noticeably below 100% also shows that the task of classifying discourse relations in the absence of unambiguous markers is difficult even for humans.

Relation	number of examples
CONTRAST	213
EXPLANATION	268
CONTINUATION	260
RESULT	266
SUMMARY	44

Table 1: Examples per relation in the data set

2.2 Machine Learning Framework

We chose BoosTexter (Schapire and Singer, 2000) as our machine learner. BoosTexter was originally developed for text categorisation. It combines a boosting algorithm with simple decision rules and allows a variety of feature types, such as nominal, numerical or text-valued features. Text-valued features can, for instance, encode sequences of words or parts-of-speech. BoosTexter applies statistical models to automatically identify informative n-grams when forming classification hypotheses for these features (i.e., it tries to detect n-grams in the sequence which are good predictors for a given class label). BoosTexter's effective modelling of n-gram features makes it particularly suitable for our

[3] One reason for excluding the latter is that relations between larger text spans are distributed differently than relations between sentences or clauses, e.g., RST relations like ELABORATION, JOINT, and BACKGROUND are more frequent between larger units than between sentences and clauses whereas relations like CONTRAST, RESULT, and EXPLANATION are more frequent between smaller units. Consequently relations between larger units are often treated by different means than inter- or intra-sentential relations (see e.g. Marcu (2000)).

task as we can directly encode the words, stems, hypernyms etc. of the two text spans involved in a relation as text-valued features. In addition to supporting n-gram features, BoosTexter also allows the use of *sparse n-grams*, i.e. n-grams with variable slots. For instance, the sparse n-gram *Dow Jones * sank* would match among others the 4-grams *Dow Jones Industrials sank* and *Dow Jones index sank*. We experimented with both, normal and sparse n-grams up to $n = 3$ and $n = 4$. The next section lists the features in detail.

2.3 Lexical Features

We implemented 10 lexical feature pairs (with one feature for the left and the other for the right span), encoding tokens (with and without punctuation and stop words), stems, lemmas, content word lemmas, word sense disambiguated lemmas, and hypernyms. To extract this information, we employed a number of pre-processing tools: Tokenisation, lemmatisation, and part-of-speech tagging were done by the tools supplied with the RASP parser (Briscoe et al., 2006).[4] Stemming was performed by applying the Porter stemmer (Porter, 1980). For the hypernym back-off we needed to word sense disambiguate the data. This was done by employing the SenseRelate disambiguation package (Pedersen et al., 2005). In this approach a target word is disambiguated by computing the semantic relatedness between each of its possible senses and all possible senses of the neighbouring words, and then choosing the sense that gives rise to the highest relatedness score. Semantic relatedness between two senses is computed by looking at their gloss overlap in WordNet 2.0 (Fellbaum, 1998). Below, we discuss the features in more detail, using the span pair in example (9) for illustration, where *(LS)* and *(RS)* indicated the left and right span, respectively.

(9) (LS) A broker may have to approach as many as 20 underwriters who insure the endeavors on behalf of the syndicates.
(RS) It could take six months for a claim to be paid.
(RESULT)

Words: encodes the spans as they occur in the text after tokenisation and normalising capitalisation:

(10) (LS) a broker may have to approach as many as 20 underwriters who insure the endeavors on behalf of the syndicates .
(RS) it could take six months for a claim to be paid .

We also encoded variants of this feature pair in which punctuation characters and/or stop words were removed.

Lemmas: encodes the original strings with all words lemmatised:

[4] Note that we did not employ full parsing or indeed any syntactic information, such as chunking.

(11) (LS) a broker may have to approach as many as 20 underwriter who insure the endeavor on behalf of the syndicate .
(RS) it can take six month for a claim to be pay .

Stems: encodes the original strings with all words stemmed:

(12) (LS) a broker mai have to approach as mani as 20 underwrit who insur the endeavor on behalf of the syndic .
(RS) it could take six month for a claim to be paid .

Content word lemmas: encodes only content word lemmas in the two spans. Named entities and numbers are replaced by placeholders (NE and NUM, respectively). We identified named entities and numbers from the part-of-speech tagged spans.

(13) (LS) broker approach NUM underwriter insure endeavor syndicate
(RS) take NUM month claim pay

Word sense disambiguated lemmas: encodes all lemmas in the original spans but lemmas are disambiguated where possible (i.e., if the lemma can be found in WordNet):

(14) (LS) a broker#n#1 may#v have#v#13 to approach#v#5 as many as 20 underwriter#n#1 who insure#v#1 the endeavor#n#1 on behalf#n#1 of the syndicate#n#2
(RS) it could#v take#v#10 six month#n#2 for a claim#n#1 to be#v#1 pay#v#8

Hypernym back-off for all word sense disambiguated lemmas: all word sense disambiguated lemmas are replaced by their direct hypernyms in WordNet (example (15)).[5] We also implemented two variants in which we back-off to hypernyms that are two and three levels higher up the hierarchy (see example (16) for a three level back-off).

(15) (LS) a businessperson#n#1 may#v have#v#13 to address#v#9 as many as 20 agent#n#4 who verify#v#1 the undertaking#n#1 on stead#n#1 of the association#n#1
(RS) it could#v decide#v#1 six time_unit#n#1 for a assertion#n#1 to be#v#1 be#v#1

(16) (LS) a person#n#1 may#v have#v#13 to travel#v#1 as many as 20 capitalist#n#2 who confirm#v#1 the activity#n#1 on duty#n#1 of the social_group#n#1
(RS) it could#v decide#v#1 six abstraction#n#6 for a statement#n#1 to be#v#1 be#v#1

[5] The repeated occurrence *be#v#1* at the end of the second span in example (15) can be explained as follows. The first occurrence comes from *be* in *to be paid* for which there are no hypernyms for the assigned sense *be#v#1*. The second occurrence of *be#v#1* comes from the word *pay* which is wrongly disambiguated and assigned the sense used in *it pays to go through trouble*. The direct hypernym of this sense in WordNet 2.0 is also *be#v#1*. Hence the repeated occurrence of this sense at the end of the second span.

Hypernym back-off for infrequent lemmas: for this feature, lemmas are backed-off to their hypernyms if they occur only once in the data set (*hapax legomena*).[6] For example, the lemmas *endeavour* and *syndicate* are replaced by their hypernyms *undertaking* and *association*, respectively.

(17) (LS) a broker#n#1 may#v have#v#13 to approach#v#5 as many as 20 underwriter#n#1 who insure#v#1 the **undertaking#n#1** on behalf#n#1 of the **association#n#1**
(RS) it could#v take#v#10 six month#n#2 for a claim#n#1 to be#v#1 pay#v#8

Placeholder back-off for infrequent lemmas: this feature is a variant of the previous one, hapaxes are replaced by a placeholder (*INFR*) rather than backed-off to the next hypernym level.

(18) (LS) a broker#n#1 may#v have#v#13 to approach#v#5 as many as 20 underwriter#n#1 who insure#v#1 the **INFR** on behalf#n#1 of the **INFR**
(RS) it could#v take#v#10 six month#n#2 for a claim#n#1 to be#v#1 pay#v#8

3 Data sparseness and classification accuracy for different lexical representations

Each of the features described in the previous section is effectively a different representation of the lexical items in the two text strings involved in a discourse relation. To determine the utility of the different representations, we determined their effect on (i) data sparseness and (ii) the accuracy of the relation classifier.

3.1 Data Sparseness

We estimated the data sparseness by computing the type-to-token ratio for different representations of lexical items (words, lemmas, stems, hypernyms, etc.) and the number of hapax legomena, both in absolute terms and in relation to the number of tokens. Table 2 shows the results.

It can be seen that the highest type-to-token ratio is achieved for word strings without punctuation and stop words. The lowest number of hapaxes is predictably achieved for *hapax back-off to placeholder* which eliminates all hapaxes. Note that *hapax back-off to hypernyms* does not eliminate all hapaxes, as we only back-off one level and the hypernym may itself only occur once in the data. Encoding only content lemmas leads to the second lowest number of singular items. Word-sense disambiguation, predictably, leads to an increase in hapaxes, which is then reduced by general hypernym back-off.

3.2 Classification Accuracy

For each lexical representation we trained a two-feature classifier, where the two features corresponded to the right and left spans of the instances in the data set. We then ran four

[6] We also experimented with other frequency thresholds, without much effect on the results.

	type-token ratio	num. of hapaxes	hapax-token ratio
words	15.90% (6241/39240)	3185	8.12%
words no punct.	19.34% (6304/32591)	3185	10.06%
words no punct. no stop	33.33% (6046/18139)	3185	17.92%
stems	13.18% (4994/37888)	2453	6.47%
lemmas	14.68% (5562/37888)	2883	7.61%
content lemmas	18.19% (3001/16500)	1447	8.77%
wsd lemmas	21.80% (7122/32672)	4039	12.36%
hypernyms all, level 1	16.85% (5506/32672)	2967	9.08%
hypernyms all, level 2	14.55% (4755/32672)	2608	7.98%
hypernyms all, level 3	13.23% (4321/32672)	2423	7.42%
hapax back-off hypernyms	19.35% (6323/32672)	3044	9.32%
hapax back-off placeholder	9.44% (3084/32672)	0	0.00%

Table 2: Data sparseness for different lexical features

10-fold cross-validation experiments for each of the 12 two-feature classifiers, using four parameter settings, i.e., n-grams and sparse n-grams up to n=3 and n=4. The average classification accuracies for each run are shown in Table 3.

While the results are all relatively close together and many of the differences are not statistically significant, some trends can be observed. With respect to the different feature types it can be seen that representing only content word lemmas generally leads to the worst results with classification accuracies between 29.29% and 30.68%. Since the classifiers that are based on an encoding that represents *all* lemmas in the spans seem to perform best (with classification accuracies between 43.38% and 45.71%), it can be concluded that non-content word lemmas (e.g. function words) are quite important for the classification task. This conclusion is further corroborated by the fact that the word-based classifier which excludes stop words performs around 10% lower than the one that includes this information. Lemmatising and stemming tend to lead to a higher performance than encoding the words in the spans directly. Word-sense disambiguation leads to a drop in accuracy compared to using the non-disambiguated lemmas but this decrease is quite small for n-grams. The lower accuracies can probably be attributed to the increased data sparseness and also the introduction of noise due to wrongly disambiguated lemmas. Indiscriminant back-off to the next hypernym level leads to a further drop in performance. Only backing-off hapaxes to their hypernyms seems to be a better strategy, though the classifiers that use these features still performed worse than those that employ the disambiguated lemmas without any back-off. Hypernym back-off

	avg. accuracy (%)			
	n-grams		sparse n-grams	
	$n \leq 3$	$n \leq 4$	$n \leq 3$	$n \leq 4$
words	41.87	41.47	43.06	44.07
words, no punct.	43.63	43.63	42.42	42.69
words, no punct. no stop	31.64	31.64	32.18	31.84
stems	43.16	43.75	43.40	43.84
lemmas	43.38	43.77	45.71	45.00
content lemmas	29.29	29.29	30.68	29.51
wsd lemmas	43.15	43.15	41.85	41.32
hypernyms all, level 1	40.35	40.29	40.59	40.01
hypernyms all, level 2	41.39	39.77	39.48	41.39
hypernyms all, level 3	38.33	39.48	38.38	39.40
hapax back-off hypernyms	42.83	41.52	39.99	42.57
hapax back-off placeholder	40.39	40.31	40.49	40.92

Table 3: Classification Accuracies, averaged over ten 10-fold cross-validation runs

tends to perform better than back-off to a simple placeholder. With respect to n-grams versus sparse n-grams, it seems that the latter generally lead to a higher accuracy but this is not true for all features.

On the whole, our results suggest that alleviating data-sparseness by morphological processing, such as stemming or lemmatising, is a more successful strategy than using semantic generalisation strategies, e.g., backing-off to hypernyms. One reason for this is probably that word sense disambiguation is by no means a solved NLP task and state-of-the art disambiguation systems still have a relatively high error rate.[7] Word sense disambiguation thus inevitably introduces noise, and this may outweigh any gains that could potentially be made by semantic back-off strategies. A second reason for the relatively low performance of the WordNet-based features may be that we used a relatively crude back-off strategy. Ideally one would want to automatically determine the right back-off level, i.e., backing-off to a concept that is general enough to reduce sparseness but specific enough to allow the classifier to discriminate between different discourse relations. Sophisticated semantic back-off strategies exist for a number of

[7] The exact proportion of errors depends on several factors, for example on how finely-grained the sense inventory is. One way to verify whether the relatively bad performance of hypernym back-off is indeed due to word sense disambiguation errors would be to re-run the experiments on data with manually disambiguated senses. Unfortunately, manual word sense disambiguation is a very time-consuming task and disambiguating the complete data set was beyond the scope of this paper. However, we manually checked a small sample of the automatically disambiguated data and found a significant proportion of errors (30-40%).

NLP tasks, such as parse disambiguation, (Clark and Weir, 2000, 2002; Li and Abe, 1998; Resnik, 1998). However, these require labelled training data and are therefore difficult to transfer to the task of determining discourse relations for which the amount of labelled training data is very small.

4 Conclusion and Outlook

In this paper we have presented an initial study on the benefit of different lexical representations for the task of classifying unmarked discourse relations. Since lexical models suffer from sparse data we investigated different methods of generalising over the actual word forms in the spans and backing-off to less sparse lexical items. We looked in particular at semantic back-off to hypernyms. Our results suggest that semantic generalisations are considerably less effective than morphological ones, such as lemmatising or stemming. Lemmatisation was found to be the best strategy. We also found that non-content word lemmas play a fairly important role in the classification task and should not be disregarded. The relatively low performance of semantic back-off models is probably largely due to errors in the word-sense disambiguation and possibly also to the difficulty of finding a suitable back-off level automatically.

While the current study focused only on lexical features, future work on the classification of discourse relations in unmarked examples should also take other sources of information into account. The main challenge for this task is to find a good representation of the *meaning* (or the most important aspects thereof) of the two spans involved in a relation. This representation should be general enough so that it minimises data sparseness and specific enough that a machine learning system can learn to discriminate between different relations. The task thus bears similarities to other complex semantic task such as recognising textual entailment (RTE) or finding paraphrases. Though, because the latter tasks aim at estimating semantic *similarity* at some level, some mileage can be gained by relatively simple methods such as word overlap. Most discourse relations, however, cannot be modelled by such simple statistical methods. The most successful RTE systems currently exploit a whole number of external resources, e.g., WordNet, logical inference, anaphora resolution, and large corpora of entailment examples (Hickl and Bensley, 2007; Giampiccolo et al., 2007). It is likely that such a multi-resource strategy is also necessary to successfully distinguish unmarked discourse relations.

Acknowledgements

This work was funded by the German Research Foundation DFG (grant PI 154/9-3). The author would like to thank the reviewers for their comments and suggestions.

References

Asher, N. and Lascarides, A. (2003). *Logics of Conversation*. Cambridge University Press.

Baldridge, J., Asher, N., and Hunter, J. (2007). Annotation for and robust parsing of discourse structure on unrestricted text. *Zeitschrift für Sprachwissenschaft*, 26(2):213–239.

Baldridge, J. and Lascarides, A. (2005). Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning CoNLL-2005*, pages 96–103.

Blair-Goldensohn, S., McKeown, K. R., and Rambow, O. (2007). Building and refining rhetorical-semantic relation models. In *Proceedings of NAACL-HLT*.

Briscoe, T., Carroll, J., and Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*.

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

Carlson, L., Marcu, D., and Okurowski, M. E. (2002). RST Discourse Treebank. Linguistic Data Consortium.

Clark, S. and Weir, D. (2000). A class-based probabilistic approach to structural disambiguation. In *Proceedings of the 18th International Conference on Computational Linguistics (Coling-00)*, pages 194–200.

Clark, S. and Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2).

Eugenio, B. D., Moore, J. D., and Paolucci, M. (1997). Learning features that predict cue usage. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*.

Fellbaum, C., editor (1998). *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Forbes, K., Miltsakaki, E., Prasad, R., Sarkar, A., Joshi, A., and Webber, B. (2001). D-LTAG System – discourse parsing with a lexicalized tree adjoining grammar. In *Proceedings of the ESSLLI-01 Workshop on Information Structure, Discourse Structure and Discourse Semantics*.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Hickl, A. and Bensley, J. (2007). A discourse commitment-based framework for recognizing textual entailment. In *The Third PASCAL Recognizing Textual Entailment Challenge*, pages 171–176.

Le Thanh, H., Abeysinghe, G., and Huyck, C. (2004). Generating discourse structures for written text. In *Proceedings of COLING-04*, pages 329–335.

Li, H. and Abe, N. (1998). Generalizing case frames using a thesaurus and the MDL priciple. *Computational Linguistics*, 24(2):217–244.

Mann, W. C. and Thompson, S. A. (1987). Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, ISI, Los Angeles, CA.

Marcu, D. (2000). The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3):395–448.

Marcu, D. and Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In *Proceedings of ACL-02*, pages 368–375.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

Murray, G., Renals, S., and Taboada, M. (2006). Prosodic correlates of rhetorical relations. In *Proceedings of HLT/NAACL ACTS Workshop*.

Nastase, V., Sayyad-Shiarabad, J., Sokolova, M., and Szpakowich, S. (2006). Learning noun-modifier semantic relations with corpus-based and WordNet-based features. In *Proceedings of AAAI*.

Pardo, T. A. S., das Graşas Volpe Nunes, M., and Rino, L. H. M. (2004). DiZer: An automatic discourse analyzer for brazilian portuguese. In *Proceedings of the 17th Brasilian Symposium on Artificial Intelligence (SBIA)*.

Pedersen, T., Banerjee, S., and Patwardhan, S. (2005). Maximizing semantic relatedness to perform word sense disambiguation. Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute.

Polanyi, L., Culy, C., van den Berg, M., Thione, G. L., and Ahn, D. (2004). Sentential structure and discourse parsing. In *Proceedings of the ACL-04 Workshop on Discourse Annotation*.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14:130–137.

Redeker, G. (1990). Ideational and pragmatic markers of discourse structure. *Journal of Pragmatics*, 14:367–381.

Resnik, P. (1998). WordNet and class-based probabilities. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 239–263. MIT Press.

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

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Sporleder, C. and Lascarides, A. (2005). Exploiting linguistic cues to classify rhetorical relations. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-05)*.

Sporleder, C. and Lascarides, A. (2008). Using automatically labelled examples to classify rhetorical relations. *Natural Language Engineering*, 14(3):369–416.