Building and Refining Rhetorical-Semantic Relation Models

Sasha Blair-Goldensohn and
Google, Inc.
76 Ninth Avenue
New York, NY
sasha@google.com

Kathleen R. McKeown† and **Owen C. Rambow**‡
† Department of Computer Science
‡ Center for Computational Learning Systems
Columbia University
{kathy,rambow}@cs.columbia.edu

Abstract

We report results of experiments which build and refine models of rhetorical-semantic relations such as Cause and Contrast. We adopt the approach of Marcu and Echihabi (2002), using a small set of patterns to build relation models, and extend their work by refining the training and classification process using parameter optimization, topic segmentation and syntactic parsing. Using human-annotated and automatically-extracted test sets, we find that each of these techniques results in improved relation classification accuracy.

1 Introduction

Relations such as Cause and Contrast, which we call rhetorical-semantic relations (RSRs), may be signaled in text by cue phrases like *because* or *however* which join clauses or sentences and explicitly express the relation of constituents which they connect (Example 1). In other cases the relation may be implicitly expressed (2).[1]

Example 1 *Because of the recent accounting scandals, there have been a spate of executive resignations.*

Example 2 *The administration was once again beset by scandal. After several key resignations ...*

[1]The authors would like to thank the four anonymous reviewers for helpful comments. This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

The first author performed most of the research reported in this paper while at Columbia University.

In this paper, we examine the problem of detecting such relations when they are not explicitly signaled. We draw on and extend the work of Marcu and Echihabi (2002). Our baseline model directly implements Marcu and Echihabi's approach, optimizing a set of basic parameters such as smoothing weights, vocabulary size and stoplisting. We then focus on improving the quality of the automatically-mined training examples, using topic segmentation and syntactic heuristics to filter out training instances which may be wholly or partially invalid. We find that the parameter optimization and segmentation-based filtering techniques achieve significant improvements in classification performance.

2 Related Work

Rhetorical and discourse theory has a long tradition in computational linguistics (Moore and Wiemer-Hastings, 2003). While there are a number of different relation taxonomies (Hobbs, 1979; McKeown, 1985; Mann and Thompson, 1988; Martin, 1992; Knott and Sanders, 1998), many researchers have found that, despite small differences, these theories have wide agreement in terms of the core phenomena for which they account (Hovy and Maier, 1993; Moser and Moore, 1996).

Work on automatic detection of rhetorical and discourse relations falls into two categories. Marcu and Echihabi (2002) use a pattern-based approach in mining instances of RSRs such as Contrast and Elaboration from large, unannotated corpora. We discuss this work in detail in Section 3. Other work uses human-annotated corpora, such as the RST Bank (Carlson et al., 2001), used by Soricut and Marcu (2003), the GraphBank (Wolf and Gibson, 2005), used by Wellner et al. (2006), or *ad-hoc* annotations, used by (Girju, 2003; Baldridge and Lascarides, 2005). In the past year, the ini-

Proceedings of NAACL HLT 2007, pages 428–435,
Rochester, NY, April 2007.

tial public release of the Penn Discourse TreeBank (PDTB) (Prasad et al., 2006) has significantly expanded the discourse-annotated corpora available to researchers, using a comprehensive scheme for both implicit and explicit relations.

Some work in RSR detection has enlisted syntactic analysis as a tool. Marcu and Echihabi (2002) filter training instances based on Part-of-Speech (POS) tags, and Soricut and Marcu (2003) use syntactic features to identify sentence-internal RST structure. Lapata and Lascarides (2004) focus their work syntactically, analyzing temporal links between main and subordinate clauses. Sporleder and Lascarides (2005) extend Marcu and Echihabi's approach with the addition of a number of features, including syntactic features based on POS and argument structure, as well as lexical and other surface features. They report that, when working with sparse training data, this richer feature set, combined with a boosting-based algorithm, achieves more accurate classification than Marcu and Echihabi's simpler, word-pair based approach (we describe the latter in the next section).

3 The M&E Framework

We model two RSRs, Cause and Contrast, adopting the definitions of Marcu and Echihabi (2002) (henceforth M&E) for their Cause-Explanation-Evidence and Contrast relations, respectively. In particular, we follow their intuition that in building an automated model it is best to adopt a higher-level view of relations (cf. (Hovy and Maier, 1993)), collapsing the finer-grained distinctions that hold within and across relation taxonomies.

M&E use a three-stage approach common in corpus linguistics: collect a large set of class instances (*instance mining*), analyze them to create a model of differentiating features (*model building*), and use this model as input to a *classification* step which determines the most probable class of unknown instances.

The intuition of the M&E model is to apply a set of RSR-associated cue phrase patterns over a large text corpus to compile a training set without the cost of human annotation. For instance, Example 1 will match the Cause-associated pattern "Because of W_1 , W_2 .", where W_1 and W_2 stand for non-empty strings containing word tokens. In the aggregate, such instances increase the prior belief that, e.g., a text span containing the word *scandals* and one containing *resignations* are in a Cause relation. A critical point is that the cue words themselves (e.g., *because*) are discarded before extracting these word pairs; otherwise these cue phrases themselves would likely be the most distinguishing features learned.

More formally, M&E build up their model through the three stages mentioned above as follows: In *instance mining*, for each RSR r they compile an instance set I_r of (W_1, W_2) spans which match a set of patterns associated with r. In *model building*, features are extracted from these instances; M&E extract a single feature, namely the frequency of token pairs derived from taking the cartesian product of $W_1 = \{w_1...w_n\} \times W_2 = \{w_{n+1}...w_m\} = \{(w_1, w_{n+1})...(w_n, w_m)\}$ over each span pair instance $(W_1, W_2) \in I$; these pair frequencies are tallied for each RSR into a frequency table F_r. Then in *classification*, the most likely relation r between two unknown-relation spans W_1 and W_2 can be determined by a naïve Bayesian classifier as $\mathrm{argmax}_{r \in R} P(r|W_1, W_2)$, where the probability $P(r|W_1, W_2)$ is simplified by assuming the independence of the individual token pairs to: $\prod_{(w_i,w_j) \in W_1,W_2} P((w_i, w_j)|r)$. The frequency counts F_r are used as maximum likelihood estimators of $P((w_i, w_j)|r)$.

4 TextRels

TextRels is our implementation of the M&E framework, and serves as our platform for the experiments which follow.

For *instance mining*, we use a set of cue phrase patterns derived from published lists (e.g., (Marcu, 1997; Prasad et al., 2006)) to mine the Gigaword corpus of 4.7 million newswire documents[2] for relation instances. We mine instances of the Cause and Contrast RSRs discussed earlier, as well as a NoRel "relation". NoRel is proposed by M&E as a default model of same-topic text across which no specific RSR holds; instances are extracted by taking text span pairs which are simply sentences from the same document separated by at least three intervening sentences. Table 1 lists a sample of our ex-

[2]distributed by the Linguistic Data Consortium

Type	Sample Patterns	Instances	Instances, M&E
Cause	*BOS* Because W_1 , W_2 *EOS* *BOS* W_1 *EOS BOS* Therefore , W_2 *EOS*.	926,654	889,946
Contrast	*BOS* W_1 , but W_2 *EOS* *BOS* W_1 *EOS BOS* However , W_2 *EOS*.	3,017,662	3,881,588
NoRel	*BOS* W_1 *EOS (BOS EOS)*{3,} *BOS* W_2 *EOS*	1,887,740	1,000,000

Table 1: RSR types, sample extraction patterns, number of training instances used in TextRels, and number of training instances used by M&E. BOS and EOS are sentence beginning/end markers.

traction patterns and the total number of training instances per relation; in addition, we hold out 10,000 instances of each type, which we divide evenly into development and training sets.

For *model building*, we compile the training instances into token-pair frequencies. We implement several parameters which control the way these frequencies are computed; we discuss these parameters and their optimization in the next section.

For *classification*, we implement three binary classifiers (for Cause vs Contrast, Cause vs NoRel and Contrast vs NoRel) using the naïve Bayesian framework of the M&E approach. We implement several classification parameters, which we discuss in the next section.

5 Parameter Optimization

Our first set of experiments examine the impact of various parameter settings in TextRels, using classification accuracy on a development set as our heuristic. We find that the following parameters have strong impacts on classification:

• *Tokenizing* our training instances using stemming slightly improves accuracy and also reduces model size.

• *Laplace smoothing* is as accurate as Good-Turing, but is simpler to implement. Our experiments find peak performance with 0.25 λ value, i.e. the frequency assumed for unseen pairs.

• *Vocabulary size* of 6,400 achieves peak performance; tokens which are not in the most frequent 6,400 stems (computed over Gigaword) are replaced by an UNK pseudo-token before F is computed.

• *Stoplisting* has a negative impact on accuracy; we find that even the most frequent tokens contribute useful information to the model; a stoplist size of zero achieves peak performance.

• *Minimum Frequency* cutoff is imposed to discard from F token pair counts with a frequency of < 4; results degrade slightly below this value, and discarding this long tail of rare pair counts significantly shrinks model size.

Classif. / TestSet	Pdtb Opt	Pdtb Seg	Auto Opt	Auto Seg	Auto-S Opt	Auto-S Seg	M&E
Cau/Con	59.1	61.1	69.8	69.7	70.3	70.6	87
Cau/NR	75.2	74.3	72.7	73.5	71.2	72.3	75
Con/NR	67.4	69.7	70.7	71.3	68.2	70.0	64

Table 2: Classifier accuracy across PDTB, Auto and Auto-S test sets for the parameter-optimized classifier ("Opt") and the same classifier trained on segment-constrained instances ("Seg"). Accuracy from M&E is reported for reference, but we note that they use a different test set so the comparison is not exact. Baseline in all cases is 50%.

To evaluate the performance of our three binary classifiers using these optimizations, we follow the protocol of M&E. We present the classifier for, e.g., Cause vs NoRel with an equal number of span-pair instances for each RSR (as in training, any pattern text has been removed). We then determine the accuracy of the classifier in predicting the actual RSR of each instance; in all cases we use an equal number of input pairs for each RSR so random baseline is 50 %. We carry out this evaluation over two different test sets.

The first set ("PDTB") is derived from the Penn Discourse TreeBank (Prasad et al., 2006). We extract "Implicit" relations, i.e. text spans from adjacent sentences between which annotators have inferred semantics not marked by any surface lexical item. To extract test instances for our Cause RSR, we take all PDTB Implicit relations marked with "Cause" or "Consequence" semantics (344 total instances); for our Contrast RSR, we take instances marked with "Contrast" semantics (293 to-

tal instances).[3] PDTB marks the two "Arguments" of these relationship instances, i.e. the text spans to which they apply; these are used as test (W_1, W_2) span pairs for classification. We test the performance on PDTB data using 280 randomly selected instances each from the PDTB Cause and Contrast sets, as well as 280 randomly selected instances from our test set of automatically extracted NoRel instances (while there is a NoRel relation included in PDTB, it is too sparse to use in this testing, with 53 total examples).

The second test set ("Auto") uses the 5,000 test instances of each RSR type automatically extracted in our instance mining process.

Table 2 lists the accuracy for the optimized ("Opt") classifier over the Auto and PDTB test sets[4]. (The "Seg" columns and "Auto-S" test set are explained in the next section.)

We also list for reference the accuracy reported by M&E; however, their training and test sets are not the same so this comparison is inexact, although their test set is extracted automatically in the same manner as ours. In the Cause versus Contrast case, their reported performance exceeds ours significantly; however, in a subset of their experiments which test Cause versus Contrast on instances from the human annotated RSTBank corpus (Carlson et al., 2001) where no cue phrase is present, they report only 63% accuracy over a 56% baseline (the baseline is $> 50\%$ because the number of input examples is unbalanced).

Since we also experience a drop in performance from the automatically derived test set to the human-annotated test set (the PDTB in our case), we further examined this issue. Our goal was to see if the lower accuracy on the PDTB examples is due to (1) the inherent difficulty of identifying implicit relation spans or (2) something else, such as the corpus-switching effect due to our model being trained and tested on different corpora (Gigaword and PDTB, respectively). To informally test this, we tested against explicitly cue-phrase marked examples gathered from PDTB. That is, we used the M&E-style method for mining instances, but we gathered them from the PDTB corpus. Interestingly, we found that (1) appears to be the case: for the Cause vs. Contrast (68.7%), Cause vs. NoRel (73.0%) and (Contrast vs. NoRel (71.0%) classifiers, the performance patterns with the Auto test set rather than the results from the PDTB Implicit test set. This bolsters the argument that "synthetic" implicit relations, i.e. those created by stripping of originally present cue phrases, cannot be treated as fully equivalent to "organic" ones annotated by a human judge but which are not explicitly indicated by a cue phrase. Sporleder and Lascarides (To Appear) recently investigated this issue in greater detail, and indeed found that such synthetic and organic instances appear to have important differences.

6 Using Topic Segmentation

In our experiments with topic segmentation, we augmented the instance mining process to take account of topic segment boundaries. The intuition here is that all sentence boundaries should not be treated equally during RSR instance mining. That is, we would like to make our patterns recognize that some sentence boundaries indicate merely an orthographic break without a switch in topic, while others can separate quite distinct topics. Sometimes the latter type are marked by paragraph boundaries, but these are unreliable markers since they may be used quite differently by different authors.

Instead, we take the approach of adding topic segment boundary markers to our corpus, which we can then integrate into our RSR extraction patterns. In the case of NoRel, our assumption in our original patterns is that the presence of at least three intervening sentences is a sufficient heuristic for finding spans which are not joined by one of the other RSRs; we add the constraint that sentences in a NoRel relation be in distinct topical segments, we can increase model quality. Conversely, for two-sentence Cause and Contrast instances, we add the constraint that there must *not* be an intervening topic segment boundary between the two sentences.

[3] Note that we are using the initial PDTB release, in which only three of 24 data sections have marked Implicit relations, so that the number of such examples will presumably grow in the next release.

[4] We do not provide pre-optimization baseline accuracy because this would be arbitrarily depend on how sub-optimally we select values select parameter values. For instance, by using a Vocabulary Size of 3,200 (rather than 6,400) and a Laplace λ value of 1, the mean accuracy of the classifiers on the Auto test set drops from 71.6 to 70.5; using a Stoplist size of 25 (rather than 0) drops this number to 67.3.

Before applying these segment-augmented patterns, we must add boundary markers to our corpus. While the concept of a topic segment can be defined at various granularities, we take a goal-oriented view and aim to identify segments with a mean length of approximately four sentences, reasoning that these will be long enough to exclude some candidate NoRel instances, yet short enough to exclude a non-trivial number of Contrasts and Cause instances. We use an automatic topic segmentation tool, LCSeg (Galley et al., 2003) setting parameters so that the derived segments are of the approximate desired length. Using these parameters, LCSeg produces topic segments with a mean length of 3.51 sentences over Gigaword, as opposed to 1.54 sentences for paragraph boundaries. Using a simple metric that assumes "correct" segment boundaries always occur at paragraph boundaries, LCSeg achieves 76% precision.

We rerun the instance mining step of TextRels over the segmented training corpus, after adding the segment-based constraints mentioned above to our pattern set. Although our constraints reduce the overall number of instances available in the corpus, we extract for training the same number of instances per RSR as listed in Table 1 (our non-segment-constrained training set does not use all instances in the corpus). Using the optimal parameter settings determined in the previous section, we build our models and classifiers based on these segment-constrained instances.

To evaluate the classifiers built on the segment-constrained instances, we can essentially follow the same protocol as in our Parameter Optimization experiments. However, we must choose whether to use a held-out test set taken from the segment-constrained instances ("Auto-S") or the same test set as used to evaluate our parameter optimization, i.e. the ("Auto") test set from unsegmented training data. We decide to test on both. On the one hand, segmentation is done automatically, so it is realistic that given a "real world" document, we can compute segment boundaries to help our classification judgments. On the other hand, testing on unsegmented input allows us to compare more directly to the numbers from our previous section. Further, for tasks which would apply RSR models outside of a single-document context (e.g., for assessing coherence of a synthesized abstract), a test on unsegmented input may be more relevant. Table 2 shows the results for the "Seg" classifiers on both Auto test sets, as well as the PDTB test set.

We observe that the performance of the classifiers is indeed impacted by training on the segment-constrained instances. On the PDTB test data, performance using the segment-trained classifiers improves in two of three cases, with a mean improvement of 1.2%. However, because of the small size of this set, this margin is not statistically significant.

On the automatically-extracted test data, the segment-trained classifier is the best performer in all three cases when using the segmented test data; while the margin is not statistically significant for a single classifier, the overall accurate-inaccurate improvement is significant ($p < .05$) using a Chi-squared test. On the unsegmented test data, the segment-trained classifiers are best in two of three cases, but the overall accurate-inaccurate improvement does not achieve statistical significance. We conclude tentatively that a classifier trained on examples gleaned with topic-segment-augmented patterns performs more accurately than our baseline classifier.

7 Using Syntax

Whether or not we use topic segmentation to constrain our training instances, our patterns rely on sentence boundaries and cue phrase anchors to demarcate the extents of the text spans which form our RSR instances. However, an instance which matches such a pattern often contains some amount of text which is not relevant to the relation in question. Consider:

Example 3 *Wall Street investors, citing* **a drop in oil prices because weakness in the automotive sector***, sold off shares in GM today.*

In this case, a syntactically informed analysis could be used to extract the constituents in the cause-effect relationship from within the boldfaced nominal clause only, i.e. as "a drop in oil prices" and "weakness in the automotive sector." However, the output of our instance mining process simply splits the string around the cue phrase "because of" and extracts the entire first and second parts of the sentence as the constituents. Of course, this may be for

the best; in this case there is an implicit Cause relationship between the NP headed by *drop* and the *sold* VP which our pattern-based rules inadvertently capture; our experiments here test whether such noise is more helpful than hurtful.

Recognizing the potential complexity of using syntactic phenomena, we reduce the dimensions of the problem. First, we focus on single-sentence instances; this means we analyze only Cause and Contrast patterns, since NoRel uses only multi-sentence patterns. Second, within the Cause and Contrast instances, we narrow our investigation to the most productive pattern of each type (in terms of training instances extracted), given that different syntactic phenomena may be in play for different patterns. The two patterns we use are "W_1 because W_2" for Cause (accounts for 54% of training instances) and "W_1 , but W_2" for Contrast (accounts for 41% of training instances). Lastly, we limit the size of our training set because of parsing time demands. We use the Collins parser (Collins, 1996) to parse 400,000 instances each of Cause and Contrast for our final results. Compared with our other models, this is approximately 43% of our total Cause instances and 13% of our total Contrast instances. For the NoRel model, we use a randomly selected subset of 400,000 instances from our training set. For all relations, we use the non-segment-constrained instance set as the source of these instances.

7.1 Analyzing and Classifying Syntactic Errors

To analyze the possible syntactic bases for the type of over-capturing behavior shown in Example 3, we create a small development set of 100 examples each from Cause and Contrast training examples which fit the criteria just mentioned. We then manually identify and categorize any instances of over-capturing, labeling the relation-relevant and irrelevant spans. We find that 75% of Cause and 58% of Contrast examples contain at least some over-capturing; we observe several common reasons for over-capturing that we characterize syntactically. For example, a matrix clause with a verb of saying should not be part of the RSR. Using automatic parses of these instances created by we then design syntactic filtering heuristics based on a manual examination of parse trees of several examples from our development set.

For Contrast, we find that using the coordinating conjunction (CC) analysis of *but*, we can use a straightforward rule which limits the extent of RSR spans captured to the conjuncts/children of the CC node, e.g. by capturing only the boldfaced clauses in the following example:

Example 4 *For the past six months, management has been* **revamping positioning and strategy***, but* **also scaling back operations**.

This heuristic successfully cuts out the irrelevant temporal relative clause, retaining the relevant VPs which are being contrasted. Note that the heuristic is not perfect; ideally the adverb *also* would be filtered here, but this is more difficult to generalize since contentful adverbials, e.g. *strategically* should not be filtered out.

For the *because* pattern, we capture the right-hand span as any text in child(ren) nodes of the *because* IN node. We extend the left-hand span only as far as the first phrasal (e.g. VP) or finite clause (e.g. SBAR) node above the *because* node. Analyzing Example 3, the heuristic correctly captures the right-hand span; however, to the left of *because*, the heuristic cuts too much, and misses the key noun *drop*.

7.2 Error Analysis: Evaluating the Heuristics

The first question we ask is, how well do our heuristics work in identifying the actual correct RSR extents? We evaluate this against the Penn Discourse TreeBank (PDTB), restricting ourselves to discourse-annotated *but* and *because* sentences which match the RSR patterns which are the subject of our syntactic filtering. Since the PDTB is annotated on the same corpus as Penn TreeBank (PTB), we separately evaluate the performance of our heuristics using gold-standard PTB parses ("PDTB-Gold") versus the trees generated by Collins' parser ("PDTB-Prs"). We extract our test data from the PDTB data corresponding to section 23 of PTB, i.e. the standard testing section, so that the difference between the gold-standard and real parse trees is meaningful. Section 23 contains 60 annotated instances of *but* and 52 instances of *because* which we can use for this purpose. We define the measurement of accuracy here in terms of word-level precision/recall. That is, the set of words filtered by our heuristics are compared to the "correct"

Heuristic	PDTB-Prs	PDTB-Gold
Contrast	89.6 / 73.0 / 80.5	79.0 / 80.6 / 79.8
Cause	78.5 / 78.8 / 78.6	87.3 / 79.5 / 83.2

Table 3: Precision/Recall/F-measure of syntactic heuristics under various data sets and settings as described in Section 7.2.

words to cut, i.e. those which the annotated RSR extents exclude. The results of this analysis are shown in Table 3.

We performed an analysis of our heuristics on Section 24 of the PDTB. In that section, there are 74 relevant sentences: 20 sentences with *because*, and 54 sentences with *but*. Exactly half of all sentences (37) have no problems in the application of the heuristics (7 *because* sentences, 30 *but* sentences). Among the remaining sentences, the main source of problems is that our heuristics do not always remove matrix clauses with verbs of saying (15 cases total, 8 of which are *because* sentences). For the *but* clauses, our heuristics removed the subject in 12 cases where the PDTB did not do so. Additionally, the heuristic for *but* sentences does not correctly identify the second conjunct in five cases (choosing instead a parenthetical, for instance).

In looking at our syntactic heuristics for the Cause relationship, we see that they indeed eliminate the most frequent source of discrepancies with the PDTB, namely the false inclusion of a matrix clause of saying, resulting in 15 out of 20 perfect analyses.

We also evaluate the difference in performance between the PDTB-Gold and PDTB-Prs performance to determine to what extent using a parser (as opposed to the Gold Standard) degrades the performance of our heuristics. We find that in Section 24, 13 out of 74 sentences contain a parsing error in the relevant aspects, but the effects are typically small and result from well-known parser issues, mainly attachment errors. As we can see in Table 3, the heuristic performance using an automatic parser degrades only slightly, and as such we can expect an automatic parser to contribute to improving RSR classification (as indeed it does).

	Pdtb **U**	Test **Syn**	Set **P**	Auto **U**	Test **Syn**	Set **P**
Cau/Con	59.6	60.5	54.5	66.3	65.8	60.8
Cau/NR	72.2	74.9	52.6	70.3	70.2	57.3
Con/NR	61.6	60.2	52.2	69.4	69.8	56.8

Table 4: Classifier accuracy for the Unfiltered (U), Syntactically Filtered (Syn), and POS (P) models described in Section 7.3, over PDTB and Auto test sets. Baseline in all cases is 50%.

7.3 Classification Evaluation

We evaluate the impact of our syntactic heuristics on classification over the Auto and PDTB test sets using the same instance set of 400,000 training instances per relation. However, each applies different filters to the instances I before computing the frequencies F (all other parameters use the same values; these are set slightly differently than the optimized values discussed earlier because of the smaller training sets). In addition to an Unfiltered baseline, we evaluate Filtered models obtained with our syntactic heuristics for Cause and Contrast. To provide an additional point of comparison, we also evaluate the Part-of-Speech based filtering heuristic described by Marcu and Echihabi, which retains only nouns and verbs. Unlike the other filters, the POS-based filtering is applied to the NoRel instances as well as the Cause and Contrast instances. Table 4 summarizes the results of the classifying the PDTB and Auto test sets with these different models.

Before we examine the results, we note that the syntactic heuristic cuts a large portion of training data out. In terms of the total sum of frequencies in F_{cause}, i.e. the word pairs extracted from all cause instances, the syntactic filtering cuts out nearly half.

With this in mind, we see that while the syntactic filtering achieves slightly lower mean accuracy as compared to the Unfiltered baseline on the Auto test set, the pairs it does keep appear to be used more efficiently (the differences are significant). Even with this reduced training set, the syntactic heuristic improves performance in two out of three cases on the PDTB test set, including a 2.7 percent improvement for the Cause vs NoRel classifier. However, due to the small size of the PDTB test set, none of these differences is statistically significant.

We posit that bias in the Auto set may cause this

difference in performance across training sets; spans in the Auto set are not true arguments of the relation in the PDTB sense, but nonetheless occur regularly with the cue phrases used in instance mining and thus are more likely to be present in the test set.

Lastly, we observe that the POS-based filtering described by M&E performs uniformly poorly. We have no explanation for this at present, given that M&E's results with this filter appear promising.

8 Conclusion

In this paper, we analyzed the problem of learning a model of rhetorical-semantic relations. Building on the work of Marcu and Echihabi, we first optimized several parameters of their model, which we found to have significant impact on classification accuracy. We then focused on the quality of the automatically-mined training examples, analyzing two techniques for data filtering. The first technique, based on automatic topic segmentation, added additional constraints on the instance mining patterns; the second used syntactic heuristics to cut out irrelevant portions of extracted training examples. While the topic-segmentation filtering approach achieves significant improvement and the best results overall, our analysis of the syntactic filtering approach indicates that refined heuristics and a larger set of parsed data can further improve those results. We would also like to experiment with combining the two approaches, i.e. by applying the syntactic heuristics to an instance set extracted using topic segmentation constraints. We conclude that our experiments show that these techniques can successfully refine RSR models and improve our ability to classify unknown relations.

References

Jason Baldridge and Alex Lascarides. 2005. Probabilistic head-driven parsing for discourse structure. In *CoNLL 2005*.

L. Carlson, D. Marcu, and M.E. Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Eurospeech 2001 Workshops*.

M. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *ACL 1996*.

M. Galley, K.R. McKeown, E. Fosler-Lussier, and H. Jing. 2003. Discourse segmentation of multi-party conversation. In *ACL 2003*.

R. Girju. 2003. Automatic detection of causal relations for question answering. In *ACL 2003 Workshops*.

Jerry R. Hobbs. 1979. Coherence and coreference. *Cognitive Science*, 3(1):67–90.

E. Hovy and E. Maier. 1993. Parsimonious or profligate: How many and which discourse structure relations? Unpublished Manuscript.

A. Knott and T. Sanders. 1998. The classification of coherence relations and their linguistic markers: An exploration of two languages. *Journal of Pragmatics*, 30(2):135–175.

M. Lapata and A. Lascarides. 2004. Inferring sentence-internal temporal relations. In *HLT 2004*.

W.C. Mann and S.A. Thompson. 1988. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281.

D. Marcu and A. Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *ACL 2002*.

D. Marcu. 1997. *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*. Ph.D. thesis, University of Toronto, Department of Computer Science.

J. Martin. 1992. *English Text: System and Structure*. John Benjamins.

K.R. McKeown. 1985. *Text generation: Using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press.

J.D. Moore and P. Wiemer-Hastings. 2003. Discourse in computational linguistics and artificial intelligence. In M.A. Gernbacher A.G. Graesser and S.R. Goldman, editors, *Handbook of Discourse Processes*, pages 439–487. Lawrence Erlbaum Associates.

M.G. Moser and J.D. Moore. 1996. Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–420.

R. Prasad, E. Miltsakaki, N. Dinesh, A. Lee, A. Joshi, and B. Webber. 2006. The penn discourse treebank 1.0. annotation manual. Technical Report IRCS-06-01, University of Pennsylvania.

R. Soricut and D. Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *HLT-NAACL 2003*.

C. Sporleder and A. Lascarides. 2005. Exploiting linguistic cues to classify rhetorical relations. In *RANLP 2005*.

C. Sporleder and A. Lascarides. To Appear. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*.

B. Wellner, J. Pustejovsky, C. Havasi, R. Sauri, and A. Rumshisky. 2006. Classification of discourse coherence relations: An exploratory study using multiple knowledge sources. In *SIGDial 2006*.

F. Wolf and E. Gibson. 2005. Representing discourse coherence: A corpus-based analysis. *Computational Linguistics*, 31(2):249–287.