A Decision-Based Approach to Rhetorical Parsing

Daniel Marcu
Information Sciences Institute and Department of Computer Science
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6601
marcu@isi.edu

Abstract

We present a shift-reduce rhetorical parsing algorithm that learns to construct rhetorical structures of texts from a corpus of discourse-parse action sequences. The algorithm exploits robust lexical, syntactic, and semantic knowledge sources.

1 Introduction

The application of decision-based learning techniques over rich sets of linguistic features has improved significantly the coverage and performance of syntactic (and to various degrees semantic) parsers (Simmons and Yu, 1992; Magerman, 1995; Hermjakob and Mooney, 1997). In this paper, we apply a similar paradigm to developing a rhetorical parser that derives the discourse structure of unrestricted texts.

Crucial to our approach is the reliance on a corpus of 90 texts which were manually annotated with discourse trees and the adoption of a shift-reduce parsing model that is well-suited for learning. Both the corpus and the parsing model are used to generate learning cases of how texts should be partitioned into elementary discourse units and how discourse units and segments should be assembled into discourse trees.

2 The Corpus

We used a corpus of 90 rhetorical structure trees, which were built manually using rhetorical relations that were defined informally in the style of Mann and Thompson (1988): 30 trees were built for short personal news stories from the MUC7 coreference corpus (Hirschman and Chinchor, 1997); 30 trees for scientific texts from the Brown corpus; and 30 trees for editorials from the Wall Street Journal (WSJ). The average number of words for each text was 405 in the MUC corpus, 2029 in the Brown corpus, and 878 in the WSJ corpus. Each MUC text was tagged by three annotators; each Brown and WSJ text was tagged by two annotators.

The rhetorical structure assigned to each text is a (possibly non-binary) tree whose leaves correspond to *elementary discourse units (edu)*s, and whose internal nodes correspond to contiguous text spans. Each internal node is characterized by a *rhetorical relation*, such as ELABORATION and CONTRAST. Each relation holds between two non-overlapping text spans called NUCLEUS and SATELLITE. (There are a few exceptions to this rule: some relations, such as SEQUENCE and CONTRAST, are multinuclear.) The distinction between nuclei and satellites comes from the empirical observation that the nucleus expresses what is more essential to the writer's purpose than the satellite. Each node in the tree is also characterized by a promotion set that denotes the units that are important in the corresponding subtree. The promotion sets of leaf nodes are the leaves themselves. The promotion sets of internal nodes are given by the union of the promotion sets of the immediate nuclei nodes.

*Edu*s are defined functionally as clauses or clause-like units that are unequivocally the NUCLEUS or SATELLITE of a rhetorical relation that holds between two adjacent spans of text. For example, "because of the low atmospheric pressure" in text (1) is not a fully fleshed clause. However, since it is the SATELLITE of an EXPLANATION relation, we treat it as elementary.

[Only the midday sun at tropical latitudes is warm enough] [to thaw ice on occasion,] [but any liquid water formed in this way would evaporate almost instantly] [because of the low atmospheric pressure.] (1)

Some *edu*s may contain *parenthetical units*, i.e., embedded units whose deletion does not affect the understanding of the *edu* to which they belong. For example, the unit shown in italics in (2) is parenthetic.

This book, *which I have received from John,* is the best book that I have read in a while. (2)

The annotation process was carried out using a rhetorical tagging tool. The process consisted in assigning *edu* and parenthetical unit boundaries, in assembling *edu*s and spans into discourse trees, and in labeling the relations between *edu*s and spans with rhetorical relation names from a taxonomy of 71 relations. No explicit distinction was made between intentional, informational, and textual relations. In addition, we also marked two constituency relations that were ubiquitous in our corpora and that often subsumed complex rhetorical constituents. These relations were ATTRIBUTION, which was used to label the relation between a reporting and a reported clause, and APPOSITION. Marcu et al. (1999) discuss in detail the annotation tool and protocol and assess the inter-judge agreement and the reliability of the annotation.

3 The parsing model

We model the discourse parsing process as a sequence of shift-reduce operations. As front-end, the parser uses a *discourse segmenter*, i.e., an algorithm that partitions the input text into *edu*s. The discourse segmenter, which is also decision-based, is presented and evaluated in section 4.

The input to the parser is an empty stack and an input list that contains a sequence of elementary discourse trees, *edt*s, one *edt* for each *edu* produced by the discourse segmenter. The status and rhetorical relation associated with each *edt* is UNDEFINED, and the promotion set is given by the corresponding *edu*. At each step, the parser applies a SHIFT or a REDUCE operation. Shift operations transfer the first *edt* of the input list to the top of the stack. Reduce operations pop the two discourse trees located on the top of the stack; combine them into a new tree updating the statuses, rhetorical relation names, and promotion sets associated with the trees involved in the operation; and push the new tree on the top of the stack.

Assume, for example, that the discourse segmenter partitions a text given as input as shown in (3). (Only the *edu*s numbered from 12 to 19 are shown.) Figure 1 shows the actions taken by a shift-reduce discourse parser starting with step i. At step i, the stack contains 4 partial discourse trees, which span units [1,11], [12,15], [16,17], and [18], and the input list contains the *edt*s that correspond to units whose numbers are higher than or equal to 19.

... [Close parallels between tests and practice tests are common,[12]] [some educators and researchers say.[13]] [Test-preparation booklets, software and worksheets are a booming publishing subindustry.[14]] [But some practice products are so similar to the tests themselves that critics say they represent a form of school-sponsored cheating.[15]] (3)

["If I took these preparation booklets into my classroom,[16]] [I'd have a hard time justifying to my students and parents that it wasn't cheating,"[17]] [says John Kaminsky,[18]] [a Traverse City, Mich., teacher who has studied test coaching.[19]] ...

At step i the parser decides to perform a SHIFT operation. As a result, the *edt* corresponding to unit 19 becomes the top of the stack. At step $i+1$, the parser performs a REDUCE-APPOSITION-NS operation, that combines *edt*s 18 and 19 into a discourse tree whose nucleus is unit 18 and whose satellite is unit 19. The rhetorical relation that holds between units 18 and 19 is APPOSITION. At step i+2, the trees that span over units [16,17] and [18,19] are combined into a larger tree, using a REDUCE-ATTRIBUTION-NS operation. As a result, the status of the tree [16,17] becomes NUCLEUS and the status of the tree [18,19] becomes SATELLITE. The rhetorical relation between the two trees is ATTRIBUTION. At step $i+3$, the trees at the top of the stack are combined using a REDUCE-ELABORATION-NS operation. The effect of the operation is shown at the bottom of figure 1.

In order to enable a shift-reduce discourse parser derive any discourse tree, it is sufficient to implement one SHIFT operation and six types of REDUCE operations, whose operational semantics is shown in figure 2. For each possible pair of nuclearity assignments NUCLEUS-SATELLITE (NS), SATELLITE-NUCLEUS (SN), and NUCLEUS-NUCLEUS (NN) there are two possible ways to attach the tree located at position top in the stack to the tree located at position $top-1$. If one wants to create a binary tree whose immediate children are the trees at top and $top-1$, an operation of type REDUCE-NS, REDUCE-SN, or REDUCE-NN needs to be employed. If one wants to attach the tree at top as an extra-child of the tree at $top-1$, thus creating or modifying a non-binary tree, an operation of type REDUCE-BELOW-NS, REDUCE-BELOW-SN, or REDUCE-BELOW-NN needs to be employed. Figure 2 illustrates how the statuses and promotion sets associated with the

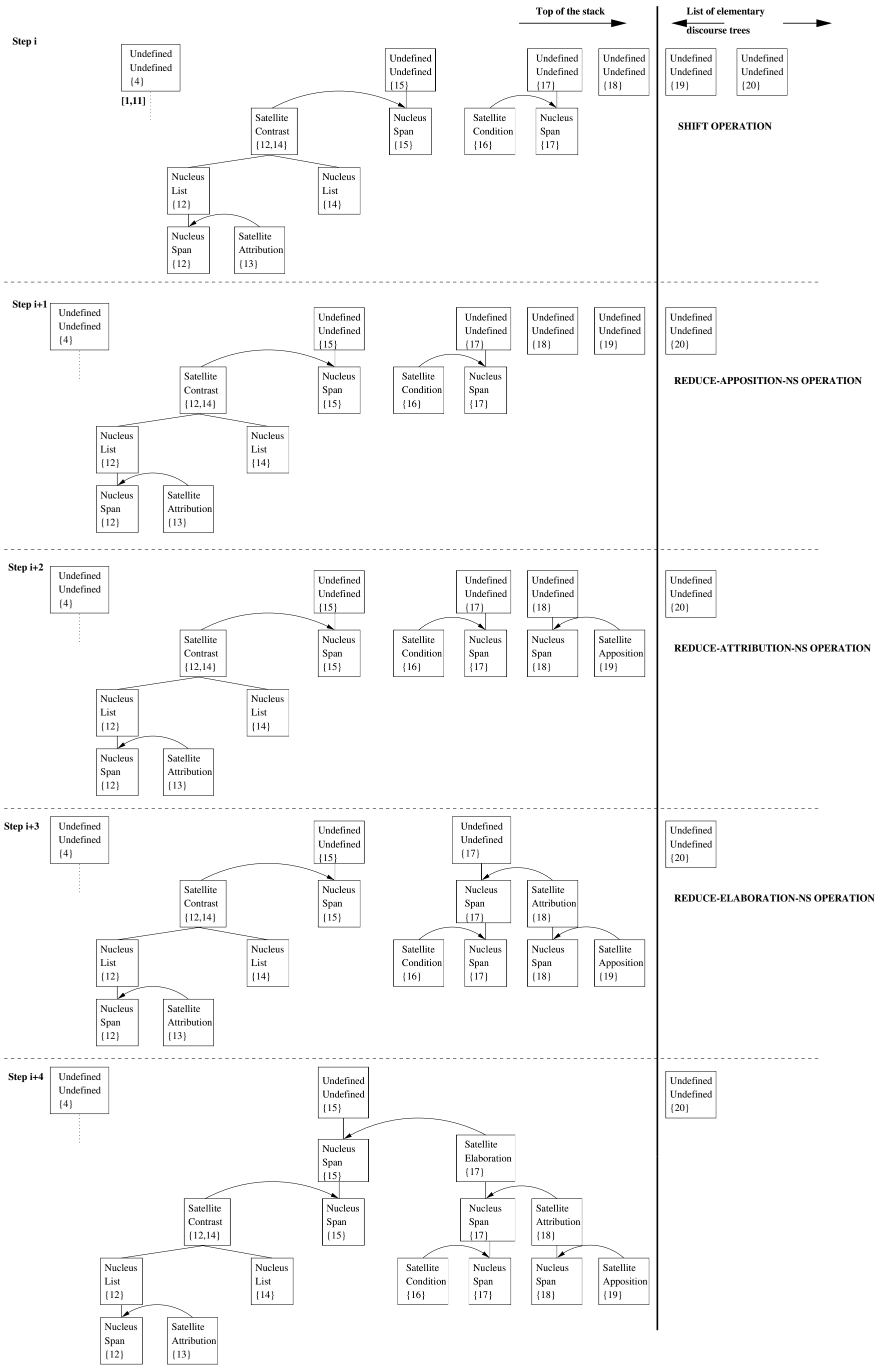

Figure 1: Example of a sequence of shift-reduce operations that concern the discourse parsing of text (3).

trees involved in the reduce operations are affected in each case.

Since the labeled data that we relied upon was sparse, we grouped the relations that shared some rhetorical meaning into clusters of rhetorical similarity. For example, the cluster named CONTRAST contained the contrast-like rhetorical relations of ANTITHESIS, CONTRAST, and CONCESSION. The cluster named EVALUATION-INTERPRETATION contained the rhetorical relations of EVALUATION and INTERPRETATION. And the cluster named OTHER contained rhetorical relations such as QUESTION-ANSWER, PROPORTION, RESTATEMENT, and COMPARISON, which were used

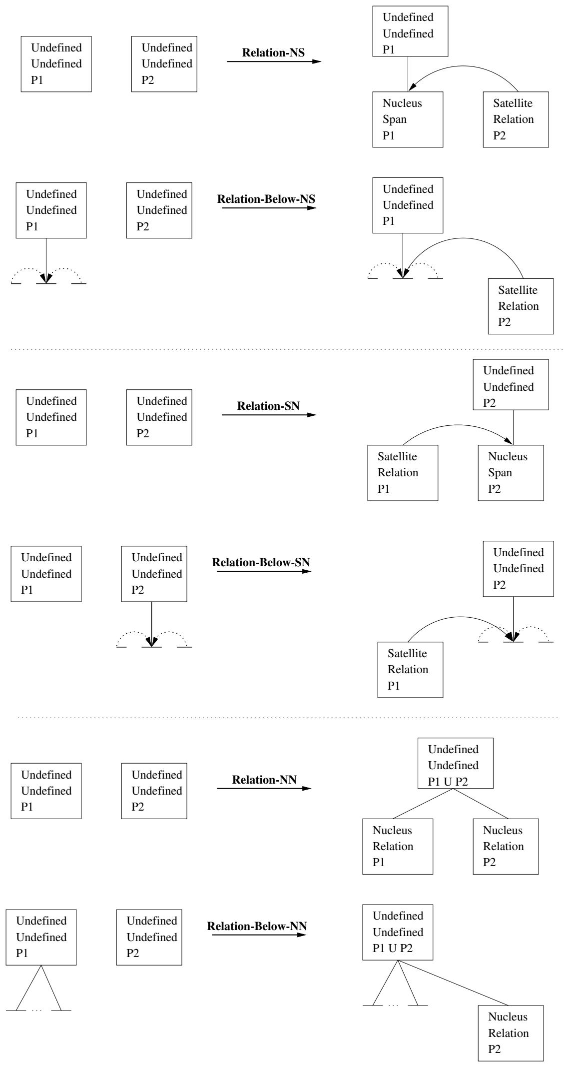

Figure 2: The reduce operations supported by our parsing model.

very seldom in the corpus. The grouping process yielded 17 clusters, each characterized by a generalized rhetorical relation name. These names were: APPOSITION-PARENTHETICAL, ATTRIBUTION, CONTRAST, BACKGROUND-CIRCUMSTANCE, CAUSE-REASON-EXPLANATION, CONDITION, ELABORATION, EVALUATION-INTERPRETATION, EVIDENCE, EXAMPLE, MANNER-MEANS, ALTERNATIVE, PURPOSE, TEMPORAL, LIST, TEXTUAL, and OTHER.

In the work described in this paper, we attempted to automatically derive rhetorical structures trees that were labeled with relations names that corresponded to the 17 clusters of rhetorical similarity. Since there are 6 types of reduce operations and since each discourse tree in our study uses relation names that correspond to the 17 clusters of rhetorical similarity, it follows that our discourse parser needs to learn what operation to choose from a set of $6 \times 17 + 1 = 103$ operations (the 1 corresponds to the SHIFT operation).

4 The discourse segmenter

4.1 Generation of learning examples

The discourse segmenter we implemented processes an input text one lexeme (word or punctuation mark) at a time and recognizes sentence and *edu* boundaries and beginnings and ends of parenthetical units. We used the leaves of the discourse trees that were built manually in order to derive the learning cases. To each lexeme in a text, we associated one learning case, using the features described in section 4.2. The classes to be learned, which are associated with each lexeme, are *sentence-break, edu-break, start-paren, end-paren,* and *none*.

4.2 Features used for learning

To partition a text into *edu*s and to detect parenthetical unit boundaries, we relied on features that model both the local and global contexts.

The local context consists of a window of size 5 that enumerates the Part-Of-Speech (POS) tags of the lexeme under scrutiny and the two lexemes found immediately before and after it. The POS tags are determined automatically, using the Brill tagger (1995). Since discourse markers, such as *because* and *and*, have been shown to play a major role in rhetorical parsing (Marcu, 1997), we also consider a list of features that specify whether a lexeme found within the local contextual window is a potential discourse marker. The local context also contains features that estimate whether the lexemes within the window are potential abbreviations.

The global context reflects features that pertain to the boundary identification process. These features specify whether a discourse marker that introduces expectations (Cristea and Webber, 1997) (such as *although*) was used in the sentence under consideration, whether there are any commas or dashes before the estimated end of the sentence, and whether there are any verbs in the unit under consideration.

A binary representation of the features that characterize both the local and global contexts yields learning examples with 2417 features/example.

4.3 Evaluation

We used the C4.5 program (Quinlan, 1993) in order to learn decision trees and rules that classify lex-

Corpus	# cases	B1(%)	B2(%)	Acc(%)
MUC	14362	91.28	93.1	96.24±0.06
WSJ	31309	92.39	94.6	97.14±0.10
Brown	72092	93.84	96.8	97.87±0.04

Table 1: Performance of a discourse segmenter that uses a decision-tree, non-binary classifier.

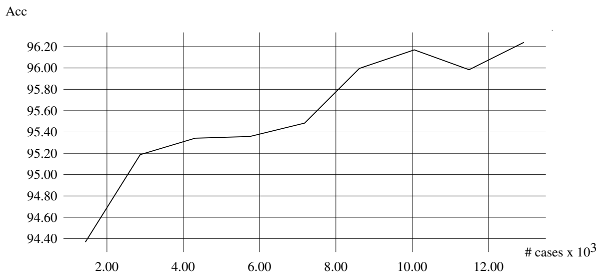

Figure 3: Learning curve for discourse segmenter (the MUC corpus).

emes as boundaries of sentences, *edu*s, or parenthetical units, or as non-boundaries. We learned both from binary (when we could) and non-binary representations of the cases.[1] In general the binary representations yielded slightly better results than the non-binary representations and the tree classifiers were slightly better than the rule-based ones. Due to space constraints, we show here (in table 1) only accuracy results that concern non-binary, decision-tree classifiers. The accuracy figures were computed using a ten-fold cross-validation procedure. In table 1, B1 corresponds to a majority-based baseline classifier that assigns *none* to all lexemes, and B2 to a baseline classifier that assigns a sentence boundary to every DOT lexeme and a non-boundary to all other lexemes.

Figure 3 shows the learning curve that corresponds to the MUC corpus. It suggests that more data can increase the accuracy of the classifier.

The confusion matrix shown in table 2 corresponds to a non-binary-based tree classifier that was trained on cases derived from 27 Brown texts and that was tested on cases derived from 3 different Brown texts, which were selected randomly. The matrix shows that the segmenter has problems mostly with identifying the beginning of parenthetical units and the intra-sentential *edu* boundaries; for example, it correctly identifies only 133 of the 220

[1]Learning from binary representations of features in the Brown corpus was too computationally expensive to terminate — the Brown data file had about 0.5GBytes.

Action		(a)	(b)	(c)	(d)	(e)
sentence-break	(a)	272				4
edu-break	(b)		133		3	84
start-paren	(c)			4		26
end-paren	(d)				20	6
none	(e)	2	38	1	4	7555

Table 2: Confusion matrix for the decision-tree, non-binary classifier (the Brown corpus).

edu boundaries. The performance is high with respect to recognizing sentence boundaries and ends of parenthetical units. The performance with respect to identifying sentence boundaries appears to be close to that of systems aimed at identifying *only* sentence boundaries (Palmer and Hearst, 1997), whose accuracy is in the range of 99%.

5 The shift-reduce action identifier

5.1 Generation of learning examples

The learning cases were generated automatically, in the style of Magerman (1995), by traversing in-order the final rhetorical structures built by annotators and by generating a sequence of discourse parse actions that used only SHIFT and REDUCE operations of the kinds discussed in section 3. When a derived sequence is applied as described in the parsing model, it produces a rhetorical tree that is a one-to-one copy of the original tree that was used to generate the sequence. For example, the tree at the bottom of figure 1 — the tree found at the top of the stack at step $i + 4$ — can be built if the following sequence of operations is performed: {SHIFT 12; SHIFT 13; REDUCE-ATTRIBUTION-NS; SHIFT 14; REDUCE-JOINT-NN; SHIFT 15; REDUCE-CONTRAST-SN; SHIFT 16; SHIFT 17; REDUCE-CONDITION-SN; SHIFT 18; SHIFT 19; REDUCE-APPOSITION-NS; REDUCE-ATTRIBUTION-NS; REDUCE-ELABORATION-NS.}

5.2 Features used for learning

To make decisions with respect to parsing actions, the shift-reduce action identifier focuses on the three top most trees in the stack and the first *edt* in the input list. We refer to these trees as the trees in focus. The identifier relies on the following classes of features.

Structural features.

- Features that reflect the number of trees in the stack and the number of *edt*s in the input list.
- Features that describe the structure of the trees in focus in terms of the type of textual units that they subsume (sentences, paragraphs, titles); the number

of immediate children of the root nodes; the rhetorical relations that link the immediate children of the root nodes, etc.[2]

Lexical (cue-phrase-like) and syntactic features.

• Features that denote the actual words and POS tags of the first and last two lexemes of the text spans subsumed by the trees in focus.

• Features that denote whether the first and last units of the trees in focus contain potential discourse markers and the position of these markers in the corresponding textual units (beginning, middle, or end).

Operational features.

• Features that specify what the last five parsing operations performed by the parser were.[3]

Semantic-similarity-based features.

• Features that denote the semantic similarity between the textual segments subsumed by the trees in focus. This similarity is computed by applying in the style of Hearst (1997) a cosine-based metric on the morphed segments.

• Features that denote Wordnet-based measures of similarity between the bags of words in the promotion sets of the trees in focus. We use 14 Wordnet-based measures of similarity, one for each Wordnet relation (Fellbaum, 1998). Each of these similarities is computed using a metric similar to the cosine-based metric. Wordnet-based similarities reflect the degree of synonymy, antonymy, meronymy, hyponymy, etc. between the textual segments subsumed by the trees in focus. We also use $14 \times 13/2$ relative Wordnet-based measures of similarity, one for each possible pair of Wordnet-based relations. For each pair of Wordnet-based measures of similarity w_{r_1} and w_{r_2}, each relative measure (feature) takes the value $<, =$, or $>$, depending on whether the Wordnet-based similarity w_{r_1} between the bags of words in the promotion sets of the trees in focus is lower, equal, or higher that the Wordnet-based similarity w_{r_2} between the same bags of words. For example, if both the synonymy- and meronymy-based measures of similarity are 0, the relative similarity between the synonymy and meronymy of the trees in focus will have the value $=$.

[2]The identifier assumes that each sentence break that ends in a period and is followed by two '\n' characters, for example, is a paragraph break; and that a sentence break that does not end in a punctuation mark and is followed by two '\n' characters is a title.

[3]We could generate these features because, for learning, we used sequences of shift-reduce operations and not discourse trees.

Corpus	# cases	B3(%)	B4(%)	Acc(%)
MUC	1996	50.75	26.9	61.12±1.61
WSJ	4360	50.34	27.3	61.65±0.41
Brown	8242	50.18	28.1	61.81±0.48

Table 3: Performance of the tree-based, shift-reduce action classifiers.

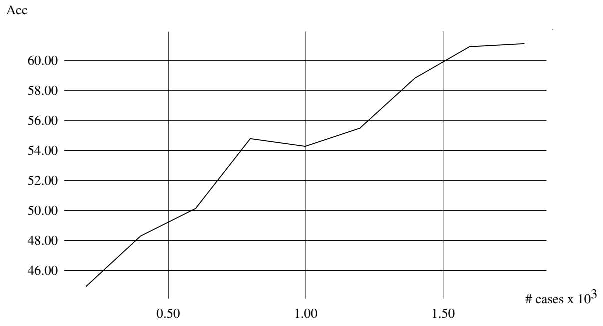

Figure 4: Learning curve for the shift-reduce action identifier (the MUC corpus).

A binary representation of these features yields learning examples with 2789 features/example.

5.3 Evaluation

The shift-reduce action identifier uses the C4.5 program in order to learn decision trees and rules that specify how discourse segments should be assembled into trees. In general, the tree-based classifiers performed slightly better than the rule-based classifiers. Due to space constraints, we present here only performance results that concern the tree classifiers. Table 3 displays the accuracy of the shift-reduce action identifiers, determined for each of the three corpora by means of a ten-fold cross-validation procedure. In table 3, the B3 column gives the accuracy of a majority-based classifier, which chooses action SHIFT in all cases. Since choosing only the action SHIFT never produces a discourse tree, in column B4, we present the accuracy of a baseline classifier that chooses shift-reduce operations randomly, with probabilities that reflect the probability distribution of the operations in each corpus.

Figure 4 shows the learning curve that corresponds to the MUC corpus. As in the case of the discourse segmenter, this learning curve also suggests that more data can increase the accuracy of the shift-reduce action identifier.

6 Evaluation of the rhetorical parser

Obviously, by applying the two classifiers sequentially, one can derive the rhetorical structure of any

Corpus	Seg-ment-er	Train-ing corpus	Elementary units				Hierarchical spans				Span nuclearity				Rhetorical relations			
			Judges		Parser		Judges		Parser		Judges		Parser		Judges		Parser	
			R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P
MUC	DT	MUC	88.0	88.0	37.1	100.0	84.4	84.4	38.2	61.0	79.1	83.5	25.5	51.5	78.6	78.6	14.9	28.7
	DT	All			75.4	96.9			70.9	72.8			58.3	68.9			38.4	45.3
	M	MUC			100.0	100.0			87.5	82.3			68.8	78.2			72.4	62.8
	M	All			100.0	100.0			84.8	73.5			71.0	69.3			66.5	53.9
WSJ	DT	WSJ	85.1	86.8	18.1	95.8	79.9	80.1	34.0	65.8	67.6	77.1	21.6	54.0	73.1	73.3	13.0	34.3
	DT	All			25.1	79.6			40.1	66.3			30.3	58.5			17.3	36.0
	M	WSJ			100.0	100.0			83.4	84.2			63.7	79.9			56.3	57.9
	M	All			100.0	100.0			83.0	85.0			69.0	82.4			59.8	63.2
Brown	DT	Brown	89.5	88.5	60.5	79.4	80.6	79.5	57.3	63.3	67.6	75.8	44.6	57.3	69.7	68.3	26.7	35.3
	DT	All			44.2	80.3			44.7	59.1			33.2	51.8			15.7	25.7
	M	Brown			100.0	100.0			81.1	73.4			60.1	67.0			59.5	45.5
	M	All			100.0	100.0			80.8	77.5			60.0	72.0			51.8	44.7

Table 4: Performance of the rhetorical parser: labeled (R)ecall and (P)recision. The segmenter is either Decision-Tree-Based (DT) or Manual (M).

text. Unfortunately, the performance results presented in sections 4 and 5 only suggest how well the discourse segmenter and the shift-reduce action identifier perform with respect to individual cases. They say nothing about the performance of a rhetorical parser that relies on these classifiers.

In order to evaluate the rhetorical parser as a whole, we partitioned randomly each corpus into two sets of texts: 27 texts were used for training and the last 3 texts were used for testing. The evaluation employs labeled recall and precision measures, which are extensively used to study the performance of syntactic parsers. Labeled recall reflects the number of correctly labeled constituents identified by the rhetorical parser with respect to the number of labeled constituents in the corresponding manually built tree. Labeled precision reflects the number of correctly labeled constituents identified by the rhetorical parser with respect to the total number of labeled constituents identified by the parser.

We computed labeled recall and precision figures with respect to the ability of our discourse parser to identify elementary units, hierarchical text spans, text span nuclei and satellites, and rhetorical relations. Table 4 displays results obtained using segmenters and shift-reduce action identifiers that were trained either on 27 texts from each corpus and tested on 3 unseen texts from the same corpus; or that were trained on 27×3 texts from all corpora and tested on 3 unseen texts from each corpus. The training and test texts were chosen randomly. Table 4 also displays results obtained using a manual discourse segmenter, which identified correctly all *edu*s. Since all texts in our corpora were manually annotated by multiple judges, we could also compute an upper-bound of the performance of the rhetorical parser by calculating for each text in the test corpus and each judge the average labeled recall and precision figures with respect to the discourse trees built by the other judges. Table 4 displays these upper-bound figures as well.

The results in table 4 primarily show that errors in the discourse segmentation stage affect significantly the quality of the trees our parser builds. When a segmenter is trained only on 27 texts (especially for the MUC and WSJ corpora, which have shorter texts than the Brown corpus), it has very low performance. Many of the intra-sentential *edu* boundaries are not identified, and as a consequence, the overall performance of the parser is low. When the segmenter is trained on 27×3 texts, its performance increases significantly with respect to the MUC and WSJ corpora, but decreases with respect to the Brown corpus. This can be explained by the significant differences in style and discourse marker usage between the three corpora. When a perfect segmenter is used, the rhetorical parser determines hierarchical constituents and assigns them a nuclearity status at levels of performance that are not far from those of humans. However, the rhetorical labeling of discourse spans is even in this case about 15-20% below human performance.

These results suggest that the features that we use are sufficient for determining the hierarchical structure of texts and the nuclearity statuses of discourse segments. However, they are insufficient for determining correctly the elementary units of discourse and the rhetorical relations that hold between discourse segments.

7 Related work

The rhetorical parser presented here is the first that employs learning methods and a thorough evaluation methodology. All previous parsers aimed at determining the rhetorical structure of unrestricted texts (Sumita et al., 1992; Kurohashi and Nagao, 1994; Marcu, 1997; Corston-Oliver, 1998) employed manually written rules. Because of the lack of discourse corpora, these parsers did not evaluate the correctness of the discourse trees they built per se, but rather their adequacy for specific purposes: experiments carried out by Miike et al. (1994) and Marcu (1999) showed only that the discourse structures built by rhetorical parsers (Sumita et al., 1992; Marcu, 1997) can be used successfully in order to improve retrieval performance and summarize text.

8 Conclusion

In this paper, we presented a shift-reduce rhetorical parsing algorithm that learns to construct rhetorical structures of texts from tagged data. The parser has two components: a discourse segmenter, which identifies the elementary discourse units in a text; and a shift-reduce action identifier, which determines how these units should be assembled into rhetorical structure trees.

Our results suggest that a high-performance discourse segmenter would need to rely on more training data and more elaborate features than the ones described in this paper — the learning curves did not converge to performance limits. If one's goal is, however, to construct discourse trees whose leaves are sentences (or units that can be identified at high levels of performance), then the segmenter described here appears to be adequate. Our results also suggest that the rich set of features that constitute the foundation of the action identifier are sufficient for constructing discourse hierarchies and for assigning to discourse segments a rhetorical status of nucleus or satellite at levels of performance that are close to those of humans. However, more research is needed in order to approach human performance in the task of assigning to segments correct rhetorical relation labels.

Acknowledgements. I am grateful to Ulf Hermjakob, Kevin Knight, and Eric Breck for comments on previous drafts of this paper.

References

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Simon H. Corston-Oliver. 1998. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis. *The AAAI Spring Symposium on Intelligent Text Summarization*, pages 9–15.

Dan Cristea and Bonnie L. Webber. 1997. Expectations in incremental discourse processing. In *Proceedings of ACL/EACL'97*, pages 88–95.

Christiane Fellbaum, editor. 1998. *Wordnet: An Electronic Lexical Database*. The MIT Press.

Marti A. Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Ulf Hermjakob and Raymond J. Mooney. 1997. Learning parse and translation decisions from examples with rich context. In *Proceedings of ACL/EACL'97*, pages 482–489.

Lynette Hirschman and Nancy Chinchor, 1997. *MUC-7 Coreference Task Definition*.

Sadao Kurohashi and Makoto Nagao. 1994. Automatic detection of discourse structure by checking surface information in sentences. In *Proceedings of COLING'94*, volume 2, pages 1123–1127.

David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proceedings of ACL'95*, pages 276–283.

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

Daniel Marcu. 1997. The rhetorical parsing of natural language texts. In *Proceedings of ACL/EACL'97*, pages 96–103.

Daniel Marcu. 1999. Discourse trees are good indicators of importance in text. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*. The MIT Press. To appear.

Daniel Marcu, Estibaliz Amorrortu, and Magdalena Romera. 1999. Experiments in constructing a corpus of discourse trees. *The ACL'99 Workshop on Standards and Tools for Discourse Tagging*.

Seiji Miike, Etsuo Itoh, Kenji Ono, and Kazuo Sumita. 1994. A full-text retrieval system with a dynamic abstract generation function. In *Proceedings of SIGIR'94*, pages 152–161.

David D. Palmer and Marti A. Hearst. 1997. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–269.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

R.F. Simmons and Yeong-Ho Yu. 1992. The acquisition and use of context-dependent grammars for English. *Computational Linguistics*, 18(4):391–418.

K. Sumita, K. Ono, T. Chino, T. Ukita, and S. Amano. 1992. A discourse structure analyzer for Japanese text. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, volume 2, pages 1133–1140.