Developing Algorithms for Discourse Segmentation

Diane J. Litman
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
diane@research.att.com

Rebecca J. Passonneau
Department of Computer Science
Columbia University
New York, NY 10027
becky@cs.columbia.edu

Abstract

The structuring of discourse into multi-utterance segments has been claimed to correlate with linguistic phenomena such as reference, prosody, and the distribution of pauses and cue words. We discuss two methods for developing segmentation algorithms that take advantage of such correlations, by analyzing a coded corpus of spoken narratives. The coding includes a linear segmentation derived from an empirical study we conducted previously. Hand tuning based on analysis of errors guides the development of input features. We use machine learning techniques to automatically derive algorithms from the same input. Relative performance of the hand-tuned and automatically derived algorithms depends in part on how segment boundaries are defined. Both methods come much closer to human performance than our initial, untuned algorithms.

Introduction

Although it has been taken for granted that discourse has a global structure, and that structural devices like lexical choices, prosodic cues, and referential devices are partly conditioned by and reflect this structure, few natural language processing systems make use of this presumed interdependence. This partly reflects limitations in both understanding and generation technology, but also reflects a dearth of verifiable data. In our view, a major drawback to exploiting the relation between global structure and linguistic devices is that there is far too little data about how they mutually constrain one another, and how the constraints might vary across genres, modalities, and individual speakers (or writers). We have been engaged in a study addressing this gap, and report here on our methodology for hypothesis testing and development. In previous work (Passonneau & Litman 1993), we reported on a method for empirically validating global discourse units, and on our evaluation of three algorithms to identify these units, each based on a single type of linguistic feature. Our focus here is on developing more predictive segmentation algorithms by exploiting multiple features simultaneously.

In this paper, we discuss two methods for developing segmentation algorithms using multiple types of linguistic cues. In section 2, we give a brief overview of related work, followed by a summary of our previous results in section 3. In section 4, we discuss our analysis of the two error types in our earlier work, mis-identification of a non-boundary, and mis-identification of a boundary. Our analysis leads us to an enhancement of the input features, and to a particular combination of knowledge sources. In section 5, we discuss our use of machine learning tools to automatically derive segmentation algorithms. The input to the machine learning program includes the original set of input features, new features that can be automatically extracted from the data, and the enhanced features discussed in section 4. Both the hand tuned and automatically derived algorithms improve considerably over previous performance, performing similarly to one another and much more comparably to humans.

Related Work

Segmentation has played a significant role in much work on discourse comprehension and generation. Grosz and Sidner (Grosz & Sidner 1986) propose a tri-partite discourse structure of three isomorphic levels: linguistic structure (roughly equivalent to what we refer to as segmental structure), intentional structure, and attentional state. The structural relations in their model are *dominates* and *precedes*. In other work (e.g., (Hobbs 1979)(Polanyi 1988)), segmental structure is an artifact of coherence relations among utterances, and few if any specific claims are made regarding segmental structure per se. Another tradition of defining relations among utterances arises out of Rhetorical Structure Theory (Mann & Thompson 1988), an approach which informs much work in generation (e.g., (Moore & Paris 1993)). In addition, recent work (Moore & Paris 1993) (Moore & Pollack 1992) has addressed the integration of intentions and rhetorical relations. Although all of these approaches have involved detailed analyses of individual discourses or representative corpora, we believe that there is a need for many more rigorous empirical studies of discourse corpora of various modali-

ties (e.g., spoken, written), genres (e.g., task-oriented, narrative), registers (e.g., conversational, formal), etc.

For example, despite the observation that segmentation is rather subjective (Mann, Matthiessen, & Thompson 1992) (Johnson 1985), several researchers have begun to investigate the ability of humans to agree with one another on segmentation, and to propose methodologies for quantifying their findings. Grosz and Hirschberg (Grosz & Hirschberg 1992) (Hirschberg & Grosz 1992) asked subjects to locally and globally structure AP news stories, according to the model of Grosz and Sidner (Grosz & Sidner 1986). Hearst (Hearst 1993) (Hearst 1994) asked subjects to place boundaries between paragraphs of expository texts, to indicate topic changes. As will be discussed below, we asked subjects to segment transcripts of oral narratives using an informal notion of speaker intention. To quantify these findings, all of these studies used notions of agreement (Gale, Church, & Yarowsky 1992) and/or reliability (Passonneau & Litman 1993).

By asking subjects to segment discourse using a non-linguistic criterion, the correlation of linguistic devices with independently derived segments can then be investigated. Grosz and Hirschberg (Grosz & Hirschberg 1992) (Hirschberg & Grosz 1992) derived discourse structures by incorporating the structural features agreed upon by all of their subjects, then used statistical measures to characterize these discourse structures in terms of acoustic-prosodic features. Morris and Hirst (Morris & Hirst 1991) structured a set of magazine texts using the theory of (Grosz & Sidner 1986), developed a thesaurus-based lexical cohesion algorithm to segment text, then qualitatively compared their segmentations with the results. Hearst (Hearst 1993) derived discourse structures by incorporating boundaries agreed upon by the majority of her subjects, developed a lexical segmentation algorithm based on information retrieval measurements, then qualitatively compared the results with the structures derived from both her subjects and from those of Morris and Hirst. In addition, Hearst (Hearst 1994) presented two implemented segmentation algorithms based on term repetition, and compared the boundaries produced to the boundaries marked by at least 3 of 7 subjects, using information retrieval metrics. Kozima (Kozima 1993) had 16 subjects segment a simplified short story, developed an algorithm based on lexical cohesion, and qualitatively compared the results. Reynar (Reynar 1994) proposed an algorithm based on lexical cohesion in conjunction with a graphical technique, and used information retrieval metrics to evaluate the algorithm's performance in locating boundaries between concatenated news articles (rather than in segmenting a single document). We (Passonneau & Litman 1993) derived segmentations based on the statistical significance of the agreement among our subjects, developed three segmentation algorithms based on results in the discourse literature, then used measures from information retrieval to quantify and evaluate the correlation.

Finally, machine learning has begun to be used as a way of automatically inducing and evaluating segmentation algorithms (in the form of decisions trees or if-then rules) directly from coded data. Grosz and Hirschberg (Grosz & Hirschberg 1992) used the system CART (Brieman *et al.* 1984) to construct decision trees for classifying aspects of discourse structure from intonational feature values. Machine learning has also been used in other areas of discourse (Siegel in press) (Siegel & McKeown 1994) (Litman 1994) (Soderland & Lehnert 1994), to both automate and improve upon the results of previous work.

Our Previous Results

We have been investigating narratives from a corpus of monologues originally collected by Chafe (Chafe 1980), known as the Pear stories. The first phase of our study, reported in (Passonneau & Litman 1993), posed two empirical questions. First, we investigated whether units of global structure consisting of sequences of utterances can be reliably identified by naive subjects. Our primary motivation here involved an important methodological consideration. We are ultimately interested in the interdependence between global discourse structure and lexico-grammatical and prosodic features, but this can be objectively investigated only if global structure can be reliably identified on independent grounds. Having found statistically significant agreement on identification of contiguous segments ($.114\ x\ 10^6 < p < .6\ x\ 10^9$), we could address our second question.

We looked at the predictive power of three types of linguistic cues in identifying the segment boundaries agreed upon by a significant number of subjects (at least 4 out of 7). We used three distinct algorithms based on the distribution of referential noun phrases (NPs), cue words, and pauses. The pause and cue algorithms directly incorporated some of the existing observations in the literature (e.g., (Hirschberg & Litman 1993)) and the NP algorithm was developed by the second author from existing and new hypotheses. The algorithms were designed to replicate the segmentation task presented to the human subjects. This task was to break up a narrative into contiguous segments, with segment breaks falling between the prosodic phrases that had been identified by Chafe and his students (Chafe 1980). The algorithms differed in the amount of pre-processing of the data that was required, and in the amount of knowledge exploited. The NP algorithm used four features, all requiring hand coding. The pause and cue word algorithms required no pre-processing, and each made use of one feature.

To evaluate how well an algorithm predicts segmental structure, we used the information retrieval (IR) metrics defined in Fig. 1. Recall is defined as the ratio of correctly hypothesized boundaries to target boundaries (a/(a+c)). Recall quantifies how many "true"

Algorithm	Subjects	
	Boundary	Non-Boundary
Boundary	a	b
Non-Boundary	c	d

Recall = a/(a+c) **Fallout** = b/(b+d)
Precision = a/(a+b) **Error** = (b+c)/(a+b+c+d)

Figure 1: Information Retrieval Metrics

	Recall	Precision	Fallout	Error	Sum
Ideal	1.00	1.00	.00	.00	.00
Avg. Human	.74	.55	.09	.11	.91
NP	.50	.30	.15	.19	1.54
Cue	.72	.15	.53	.50	2.16
Pause	.92	.18	.54	.49	1.93
NP/pause	.47	.42	.08	.13	1.32

Table 1: Previous Training Results

segment boundaries are identified by the algorithm. Precision is defined as the ratio of hypothesized boundaries that are correct to the total hypothesized boundaries (a/(a+b)). Precision identifies what proportion of boundaries proposed by the algorithm are "true" boundaries. (Cf. Fig. 1 for definitions of fallout and error rate). Ideal algorithm performance would be to identify all and only the target boundaries.

Table 1 compares average algorithm performance reported in (Passonneau & Litman 1993) (Passonneau & Litman To appear) to ideal performance and average human performance in identifying target boundaries that at least 4 subjects agreed upon. We use a sum of deviations (column 6) to intuitively rate overall performance, computed for each metric by summing the differences between the observed and ideal values (ideal means that the values for b and c in Fig. 1 both equal 0, i.e., that there are no errors). Perfect performance thus has a summed deviation of 0. Recall and precision values for the worst possible performance equal 0 instead of 1, and fallout and error equal 1 instead of 0. Thus the maximum summed deviation equals 4.

As shown, human performance is far from ideal (summed deviation=.91), with precision having the greatest deviation from ideal (.45). The NP algorithm performed better than the unimodal algorithms, and a combination of the NP and pause algorithms performed best. In particular, the NP algorithm is much worse than human performance (1.54 versus .91), but performs better than cue or pause (2.16, 1.93). Initially, we tested a simple additive method for combining algorithms in which a boundary is proposed if each separate algorithm proposes a boundary. As reported in (Passonneau & Litman 1993), we evaluated all pairwise combinations of the NP, pause and cue algorithms, and the combination of all 3. The NP/pause combination had the best performance, showing some improvement over the NP alone (1.32 versus 1.54). However, performance still falls far short of human performance.

Thus, our initial results indicated that no algorithm or simple combination of algorithms performed as well as humans. We felt that significant improvements could be gained by refining the algorithms and by combining them in more complex ways.

Our current phase of research is mainly aimed at improving the predictive power of the algorithms. Here we report testing the tuned algorithms on a portion of a reserved test corpus. Furthermore, we also discuss modifications to the evaluation metrics. These metrics yield a conservative evaluation because all errors are penalized equally. That is, there is no difference between an error in which the algorithm proposes a boundary that is close to a target boundary (e.g., one utterance away) and an error in which the proposed boundary is distant from the target boundary (e.g., many utterances away).

Hand Tuning

To determine how to improve algorithm performance, we analyzed the performance errors of the NP algorithm. Fig. 1 illustrates two types of performance errors. "B" type errors occur when an algorithm identifies a non-boundary as a boundary. Reduction of "b" type errors raises precision, in addition to lowering fallout and error rate. "C" type errors occur when an algorithm identifies a boundary as a non-boundary. Reduction of "c" type errors raises recall, in addition to lowering error rate. Analysis of "b" type errors led us to enhance the input NP features, while analysis of "c" type errors led us to a particular combination of NP, cue word, pause and prosodic features.

We identified two classes of "b" type errors made by the original NP algorithm that occurred relatively frequently. The original NP algorithm relied on an encoding of the following 4 conditions:

1. Does the current prosodic phrase initiate a clause?
2. When 1 is yes, does the sequence of phrases in the current clause contain an NP that corefers with an NP in the previous clause?
3. When 1 is yes, does the sequence of phrases in the current clause contain an NP whose referent can be inferred from the previous clause?
4. When 1 is yes, does the sequence of phrases in the current clause contain a definite pronoun whose referent is mentioned in a previous prosodic phrase, up to the last boundary assigned by the algorithm?

Analysis of "b" type errors led to changes in conditions 1 and 3 above. Detailed definitions of the current NP features are given in (Passonneau 1994).

One example of a change to the coding of clauses (condition 1) involves a class of formulaic utterances that have the structure of clauses, but which function like interjections. These include the phrases *let's see, let me see, I don't know, you know* where no clausal

argument is present. In the new coding, these clausal interjections are no longer classified as distinct clauses.

One example of a change to the coding of inferential links (condition 3) involves missing arguments that can be inferred from non-NP constituents in a prior clause. Example 1 illustrates a missing argument of the verb *notice* that can be inferred to be an event referred to in the preceding clause, namely that the pears have fallen. In the original coding, there would have been no inferential link between the phrases numbered here as 3.01 and 3.02.

(1) 3.01 [1.1 [.7] A-nd] he's not really.. doesn't seem to be paying all that much attention [.55? because [.45]] you know the pears fall,
3.02 and.. he doesn't really notice,

A large proportion of "c" type errors occurred where the segmentation predictions from prosodic and lexical features conflicted with those of the NP features. Experiments led to the hypothesis that the most improvement came by assigning a boundary under the following set of conditions, where P_{i-1} and P_i are successive prosodic phrases:

5. Does P_{i-1} have 'sentence final' intonation (pitch rise or fall)?
6. Does P_i start with a pause (any duration)?
7. If so,
   (a) is the 1st lexical item in P_i a cue word other than "*and*"?
   (b) Or, if the 1st lexical item in P_i is "*and*", then is the 2nd lexical item a cue word other than "*and*"?

The original pause and cue word algorithms did not use intonation features, did not distinguish between lexical and non-lexical "words" in assigning sequential position, and did not look at combinations of "*and*" with other cue words. To summarize, the tuned algorithm relies on enhanced features of clauses, referential NPs, prosody, cue words and pauses. A boundary is assigned if condition 1 discussed earlier holds, but conditions 2-4 do not. Or a boundary is assigned if conditions 5-6 hold, and either 7a or 7b holds.

	Recall	Precision	Fallout	Error	Sum
Training	.46	.55	.06	.11	1.16
Tuned Test	.58	.39	.11	.15	1.29
Untuned Test	.44	.34	.11	.16	1.49

Table 2: Performance of Hand-Tuned Algorithm

Table 2 presents the results of the tuned algorithm on boundaries agreed on by at least 4 subjects (the next section reports results for the less conservative threshold of at least 3 subjects). Testing was performed on three randomly selected, previously unseen narratives. The average recall for the tuned algorithm is actually better on the test set than on the training set, and the precision, fallout and error rate are somewhat, but not significantly, worse. The scores for the tuned algorithm on the test data are comparable to the combined NP/pause algorithm on the training data (cf. summary scores of 1.29 versus 1.32), but much better than the untuned NP algorithm on the test data (cf. summary score of 1.29 versus 1.49).

Machine Learning

There are now a number of publically available machine learning programs which induce algorithms from preclassified examples. We have thus begun to exploit the use of machine learning to *automate* the process of developing segmentation algorithms. Our results suggest that machine learning is indeed a useful tool, and in fact not only automates algorithm development, but in some cases can also yield superior performance as compared to algorithms developed manually.

The machine learning program that we have used in our work - C4.5 (Quinlan 1993)[1] - takes two inputs. The first input is the definitions of the classes to be learned, and of the names and values of a fixed set of coding features. In our domain, there are two classes to be learned: *boundary* and *non-boundary*. Boundaries can be defined to be those potential boundary positions identified as boundaries by a majority of our subjects, as in our previous work (Passonneau & Litman 1993) (Passonneau & Litman To appear), or using some other definition as discussed below.

The features we are investigating include the set of features used in our previous algorithms, revised as described in the previous section:

- is the first lexical item in P_i a cue word?
- does P_i start with a pause?
- condition 2 (cf. "Hand Tuning" section)
- condition 3 (cf. "Hand Tuning" section)
- condition 4 (cf. "Hand Tuning" section)

We also include other features (motivated by previous observations in the literature and/or the results of hand tuning), that can be automatically computed from our transcripts:

- the first lexical item in P_i, if a cue word
- the duration (in seconds) of the pause starting P_i (if applicable)
- the type of 'final intonation' of P_{i-1}
- the type of 'final intonation' of P_i
- are the first and second lexical items in P_i cue words?
- if so, the second lexical item in P_i

[1] We have also replicated all of our experiments using the machine learning programs C4.5rules (Quinlan 1993) and Cgrendel (Cohen 1993). These programs take the same input as C4.5, but output an ordered set of if-then rules rather than a decision tree. Our results are in general quite comparable using all three systems.

	Recall	Precision	Fallout	Error	Sum
Learning ($N \geq 4$)	.33	.44	.06	.13	1.41
Learning ($N \geq 3$)	.57	.74	.04	.12	.85
Tuned ($N \geq 3$)	.58	.64	.07	.14	.98
Human ($N \geq 3$)	.63	.72	.06	.12	.83

Table 3: Training Results using Machine Learning

	Recall	Precision	Fallout	Error	Sum
Learning	.55	.53	.09	.14	1.15
Tuned	.55	.43	.11	.17	1.27
Cross validation	.50	.65	.05	.14	1.04

Table 4: Test Results, $N \geq 3$

- do conditions 5, 6, and 7 (a) or (b) hold? (cf. "Hand Tuning" section)

Detailed definitions of the features are given in (Litman & Passonneau 1995) and (Passonneau 1994).

The second input to C4.5 is the training data, i.e., a set of examples for which the class and feature values are specified. Our training corpus provides us with 1002 examples of prosodic phrases.

The output of C4.5 is a set of classification rules, expressed as a decision tree. These rules predict the class of an input phrase given its set of feature values.

Due to the ease of inducing rules under many different conditions, we are currently conducting a large number of experiments, varying both the features that are examined, as well as the definitions used for classifying a potential boundary site as a boundary or non-boundary. Table 3 presents the results of our most promising learned algorithm to date.[2] The first line of Table 3 presents the performance of the algorithm in identifying target boundaries that at least 4 subjects agreed upon, averaged over the 10 training narratives. This result is nearly comparable to our previous best result (NP/pause from Table 1), but is not nearly as good as the tuned training result presented in Table 2. The second line of Table 3 presents the performance of a different learned algorithm, and shows considerable improvement compared to the first learned algorithm. The second learned algorithm was trained using the same set of input features as the first learned algorithm, but defines boundaries to be those potential boundary sites identified as boundaries by at least *3* of 7 subjects. The third line of Table 3 shows that the second learned algorithm outperforms the manually developed algorithm (reevaluated using the revised definition of boundary). The last line of Table 3 shows that the performance of the learned algorithm is in fact comparable to human performance (reevaluated using the revised definition of boundary).

As with the hand tuned algorithm, we evaluate the algorithm developed from the training narratives by examining its performance on a separate test set of narratives. In addition, we also *estimate* the performance of the machine learning results using the method of *cross-validation*, which often provides better performance estimates given a corpus of our size (Weiss & Kulikowski 1991). Given our 10 training narratives, 10 runs of the learning program are performed, each using 9 of the narratives for training (i.e., for learning a decision tree) and the remaining narrative for testing. Note that for each run, the training and testing sets are still disjoint. The estimated recall is obtained by averaging the actual recall for each of the 10 test narratives, and likewise for the other metrics.

Table 4 presents the test results using the more promising boundary definition of $N \geq 3$. As can be seen from the first line of the table, the results on the test set are comparable to the training results, except for a degradation in precision. The learned algorithm also outperforms the hand tuned algorithm (reevaluated using the revised definition of boundary), as can be seen from the second line. Performance of the learned algorithm is even closer to the training results when estimated using cross-validation, as can be seen from the last line.

Finally, note that our results suggest that our previous boundary threshold of 4 out of 7 was too conservative. As can be seen from Tables 1, 2, and 3, we achieve significantly better human, hand-tuned, and learned results using the lower threshold of 3 out of 7 subjects on our training data. A comparison of Tables 2 and 4 also shows an extremely slight improvement when the hand-tuned algorithm is reevaluated on the test set. Hearst (Hearst 1994) similarly found better results using at least 3 rather than at least 4 out of 7 subjects. In our earlier work (Passonneau & Litman 1993)), we evaluated partitioned values of Cochran's Q, a method for evaluating statistical significance of agreement across subjects, in order to identify an appropriate threshold. We found that probabilities for agreement on boundaries among 3 subjects exceeded a significance level of $p = .01$ in 3 out of our 20 narratives. Thus we used a threshold of 4 subjects, where the probabilities were below $p = .01$ for all 20 narratives. However, the lower threshold of 3 subjects is in fact still quite significant. The probabilities for the partioning of 3 subjects were well below $p = .01$ in the 17 other narratives, and were below $p = .02$ in all 20 narratives. Given the improvement in our results, using a significance level of $p = .01$ rather than $p = .02$ now seems too conservative.

Evaluation Issues

IR metrics provide a well-understood means of evaluation, but as noted above, they are overly conservative in our domain. All errors are treated equally. Here we address the problem of giving partial credit for a

[2](Litman & Passonneau 1995) presents the actual learned decision tree, and shows how the tree exploits a large number of the available features.

Subj.	Alg.	Phr.	
		11.01	[.35] They help him$_i$,
		11.02	.. get straightened out,
		11.03	.. put the pears back,
		11.04	[.55] in the basket,
		11.05	straighten out his bicycle,
Subj		11.06	and so forth.
	Alg	12.01	[.85] And he$_i$ goes on his merry way.
	Alg	13.01	[1.25 But then.. um [.35]] the boys realize that he$_i$'s forgotten his hat.

Figure 2: Near miss by NP algorithm

"near miss", illustrated in Fig. 2. The first column of Fig. 2 shows "Subj" if a majority of subjects assigned a boundary after the current phrase, the second column shows "Alg" if the algorithm assigned a boundary, and the third column is the prosodic phrase number. A near miss occurs if the algorithm does not place a boundary at a given boundary location (e.g., after 11.06), but does place one within one phrase of the boundary (e.g., after 12.01). Here the algorithm also assigns a boundary 2 phrases after the subjects' boundary location, which is not a near miss.

The tuned algorithm does not place a boundary where the majority of subjects did (after 11.06) because the referent of the pronoun subject in 12.01 is referred to earlier in the same segment, at 11.01 (conditions 1 and 4 discussed in "Hand Tuning" section), and the prosodic and cue word features do not lead to a boundary. The tuned algorithm places a boundary after 12.01 despite the presence of a pronoun coreferring with an NP in the preceding clause, because 12.01 has 'sentence final' intonation (condition 5), and 13.01 begins with a pause (condition 6) followed by a cue word other than "*and*" (condition 7a).

In our view, the completion of one semantic unit of discourse and onset of another need not coincide neatly with the end of one phrase or utterance and onset of another. For example, in Fig. 2, the event referred to in 12.01 is semantically linked to the previous utterance in that righting the boy's bicycle is a precondition for his being able to ride away; it is also semantically linked to the next utterance in that the boys can only observe that the bicyclist has left without his hat if he has indeed left. This and related phenomena (cf. discussion in (Passonneau & Litman To appear)) suggest that a looser notion of segment boundary location more directly reflects human perception.

To allow for partial credit in cases of near misses, we have developed a simple modification of how the cell values for the matrix in Fig. 1 are computed. In the canonical scoring method, if an algorithm assigns a boundary at a non-boundary location, the value of "b" (incorrect identification of a boundary) is incremented by 1 whereas the value of "a" (correct identification of a boundary) remains unchanged. Similarly, if an algorithm assigns a non-boundary at a boundary location, the value of "c" (incorrect identification of a non-boundary) is incremented by 1 and the value of "d" (correct identification of a non-boundary) remains unchanged. Our partial-credit scoring method applies only in cases of near misses, and affects the scoring of the failure to identify a boundary, as at phrase 11.06 of Fig. 2, and the misidentification of a boundary, as at phrase 12.01. At the actual boundary location (e.g., 11.06), the algorithm is treated as half correct by incrementing the values of "c" and "d" each by .5. At the near miss, the algorithm is also treated as half correct and half incorrect by incrementing the values of "a" and "b" each by .5. This scoring method may result in fractional totals for values of "a" through "d", but note that the table total will be the same as for the canonical scoring method. Values for recall, precision and so on are then computed from the fractional frequencies in the traditional way.

	Recall	Precision	Fallout	Error	Sum
Standard	.46	.55	.06	.11	1.16
Partial credit	.48	.57	.06	.11	1.12

Table 5: Performance of Hand-Tuned Algorithm

There are 16 "near misses" in our 1002 training examples. Table 5 shows that modifying our metrics to account for near misses improves the average score of the hand tuned algorithm on the training set, albeit slightly. The development of more sophisticated definitions of near misses and modified metrics thus appears to be a promising area for future research.

Conclusion

In this paper we have presented two methods for developing segmentation algorithms. Hand tuning development is based on examining the causes of errors. Machine learning automatically induces decision trees from coded discourse corpora. In both cases, we use an enriched set of input features compared to our previous work, leading to marked improvements in performance. The fact that we have obtained fairly comparable results using both methods also adds an extra degree of reliability to our work. Our results suggest that hand-tuning is useful for designing appropriate input data representations, which machine learning can then use to maximize performance. Our results also suggest that a less conservative threshold for defining boundaries, and a less conservative version of the IR metrics, can further enhance performance. We plan to continue merging the best automated and analytic results before evaluating the outcome on a new test corpus. Because we have already used cross-validation to evaluate the learned algorithm, we do not anticipate significant performance degradation on the new test set.

Acknowledgments

The second author's work was partly supported by NSF grant IRI-91-13064. We would like to thank Jason

Catlett and William Cohen for their helpful comments regarding the use of machine learning.

References

Brieman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. *Classification and Regression Trees.* Wadsworth and Brooks.

Chafe, W. L. 1980. *The Pear Stories: Cognitive, Cultural and Linguistic Aspects of Narrative Production.* Ablex.

Cohen, W. W. 1993. Efficient pruning methods for separate-and-conquer rule learning systems. In *Proc. of the 13th Intern't'l. Joint Conference on AI.*

Gale, W.; Church, K. W.; and Yarowsky, D. 1992. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proc. of the 30th Annual Meeting of the ACL*, 249–256.

Grosz, B., and Hirschberg, J. 1992. Some intonational characteristics of discourse structure. In *Proc. of the International Conference on Spoken Language Processing (ICSLP).*

Grosz, B. J., and Sidner, C. L. 1986. Attention, intentions and the structure of discourse. *Computational Linguistics* 12:175–204.

Hearst, M. A. 1993. TextTiling: A quantitative approach to discourse segmentation. Technical Report 93/24, Sequoia 2000 Tech. Report, U. of California, Berkeley.

Hearst, M. A. 1994. Multi-paragraph segmentation of expository texts. Technical Report 94/790, EECS, U. of California, Berkeley.

Hirschberg, J., and Grosz, B. 1992. Intonational features of local and global discourse structure. In *Proc. of the Darpa Workshop on Spoken Language.*

Hirschberg, J., and Litman, D. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics* 19(3):501–530.

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

Johnson, N. S. 1985. Coding and analyzing experimental protocols. In Dijk, T. A. V., ed., *Handbook of Discourse Analysis, Vol. 2: Dimensions of Discourse.* Academic Press.

Kozima, H. 1993. Text segmentation based on similarity between words. In *Proc. of the 31st Annual Meeting of the ACL (Student Session)*, 286–288.

Litman, D. J., and Passonneau, R. J. 1995. Combining multiple knowledge sources for discourse segmentation. In *Submitted.*

Litman, D. J. 1994. Classifying cue phrases in text and speech using machine learning. In *Proc. of the Twelfth National Conference on Artificial Intelligence (AAAI)*, 806–813.

Mann, W. C., and Thompson, S. 1988. Rhetorical structure theory: towards a functional theory of text organization. *TEXT* 243–281.

Mann, W. C.; Matthiessen, C. M.; and Thompson, S. A. 1992. Rhetorical structure theory and text analysis. In Mann, W. C., and Thompson, S. A., eds., *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text.* J. Benjamins.

Moore, J. D., and Paris, C. 1993. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics* 19:652–694.

Moore, J. D., and Pollack, M. E. 1992. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics* 18:537–544.

Morris, J., and Hirst, G. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17:21–48.

Passonneau, R. J., and Litman, D. J. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Proc. of the 31st Annual Meeting of the Association for Computational Linguistics*, 148–155.

Passonneau, R. J., and Litman, D. To appear. Empirical analysis of three dimensions of spoken discourse: Segmentation, coherence and linguistic devices. In Hovy, E., and Scott, D., eds., *Burning Issues in Discourse.* Springer Verlag.

Passonneau, R. J. 1994. Protocol for coding discourse referential noun phrases and their antecedents. Technical report, Columbia University.

Polanyi, L. 1988. A formal model of discourse structure. *Journal of Pragmatics* 601–638.

Quinlan, J. R. 1993. *C4.5 : Programs for Machine Learning.* Morgan Kaufmann.

Reynar, J. C. 1994. An automatic method of finding topic boundaries. In *Proc. of the 32nd Annual Meeting of the ACL (Student Session)*, 331–333.

Siegel, E. V., and McKeown, K. R. 1994. Emergent linguistic rules from inducing decision trees: Disambiguating discourse clue words. In *Proc. of the Twelfth National Conference on Artificial Intelligence (AAAI)*, 820–826.

Siegel, E. V. in press. Competitively evolving decision trees against fixed training cases for natural language processing. In *Advances in Genetic Programming.*

Soderland, S., and Lehnert, W. 1994. Corpus-driven knowledge acquisition for discourse analysis. In *Proc. of the Twelfth National Conference on Artificial Intelligence (AAAI)*, 827–832.

Weiss, S. M., and Kulikowski, C. 1991. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems.* Morgan Kaufmann.