Combining Multiple Knowledge Sources for Discourse Segmentation

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

Rebecca J. Passonneau*
Bellcore
445 South Street
Morristown, NJ 07960
beck@bellcore.com

Abstract
We predict discourse segment boundaries from linguistic features of utterances, using a corpus of spoken narratives as data. We present two methods for developing segmentation algorithms from training data: hand tuning and machine learning. When multiple types of features are used, results approach human performance on an independent test set (both methods), and using cross-validation (machine learning).

1 Introduction
Many have argued that discourse has a global structure above the level of individual utterances, and that linguistic phenomena like prosody, cue phrases, and nominal reference are partly conditioned by and reflect this structure (cf. (Grosz and Hirschberg, 1992; Grosz and Sidner, 1986; Hirschberg and Grosz, 1992; Hirschberg and Litman, 1993; Hirschberg and Pierrehumbert, 1986; Hobbs, 1979; Lascarides and Oberlander, 1992; Linde, 1979; Mann and Thompson, 1988; Polanyi, 1988; Reichman, 1985; Webber, 1991)). However, an obstacle to exploiting the relation between global structure and linguistic devices in natural language systems is that there is too little data about how they constrain one another. We have been engaged in a study addressing this gap. In previous work (Passonneau and Litman, 1993), we reported on a method for empirically validating global discourse units, and on our evaluation of algorithms to identify these units. We found significant agreement among naive subjects on a discourse segmentation task, which suggests that global discourse units have some objective reality. However, we also found poor correlation of three untuned algorithms (based on features of referential noun phrases, cue words, and pauses, respectively) with the subjects’ segmentations.

In this paper, we discuss two methods for developing segmentation algorithms using multiple knowledge sources. In section 2, we give a brief overview of related work and summarize our previous results. In section 3, we discuss how linguistic features are coded and describe our evaluation. In section 4, we present our analysis of the errors made by the best performing untuned algorithm, and a new algorithm that relies on enriched input features and multiple knowledge sources. In section 5, we discuss our use of machine learning tools to automatically construct decision trees for segmentation from a large set of input features. Both the hand tuned and automatically derived algorithms improve over our previous algorithms. The primary benefit of the hand tuning is to identify new input features for improving performance. Machine learning tools make it convenient to perform numerous experiments, to use large feature sets, and to evaluate results using cross-validation. We discuss the significance of our results and briefly compare the two methods in section 6.

2 Discourse Segmentation
2.1 Related Work
Segmentation has played a significant role in much work on discourse. The linguistic structure of Grosz and Sidner’s (1986) tri-partite discourse model consists of multi-utterance segments whose hierarchical relations are isomorphic with intentional structure. 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. Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) is another tradition of defining relations among utterances, and informs much work in generation. In addition, recent work (Moore and Paris, 1993; Moore and 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 there is a need for more rigorous empirical studies.

Researchers have begun to investigate the ability of humans to agree with one another on segmen-
tation, and to propose methodologies for quantifying their findings. Several studies have used expert coders to locally and globally structure spoken discourse according to the model of Grosz and Sidner (1986), including (Grosz and Hirschberg, 1992; Hirschberg and Grosz, 1992; Nakatani et al., 1995; Stiefelmann, 1995). Hearst (1994) asked subjects to place boundaries between paragraphs of expository texts, to indicate topic changes. Moser and Moore (1995) had an expert coder assign segments and various segment features and relations based on RST. To quantify their findings, these studies use notions of agreement (Gale et al., 1992; Moser and Moore, 1995) and/or reliability (Passonneau and Litman, 1993; Passonneau and Litman, to appear; Isard and Carletta, 1995).

By asking subjects to segment discourse using a non-linguistic criterion, the correlation of linguistic devices with independently derived segments can then be investigated in a way that avoids circularity. Together, (Grosz and Hirschberg, 1992; Hirschberg and Grosz, 1992; Nakatani et al., 1995) comprise an ongoing study using three corpora: professionally read AP news stories, spontaneous narrative, and read and spontaneous versions of task-oriented monologues. Discourse structures are derived from subjects’ segmentations, then statistical measures are used to characterize these structures in terms of acoustic-prosodic features. Grosz and Hirschberg’s work also used the classification and regression tree system CART (Breiman et al., 1984) to automatically construct and evaluate decision trees for classifying aspects of discourse structure from intonation feature values. Morris and Hirst (1991) structured a set of magazine texts using the theory of (Grosz and Sidner, 1986), developed a thesaurus-based lexical cohesion algorithm to segment text, then qualitatively compared their segmentations with the results. 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 (1993) had 16 subjects segment a simplified short story, developed an algorithm based on lexical cohesion, and qualitatively compared the results. 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.

2.2 Our Previous Results

We have been investigating a corpus of monologues collected and transcribed by Chafe (1980), known as the Pear stories. As reported in (Passonneau and Litman, 1993), we first investigated whether units of global structure consisting of sequences of utterances could be reliably identified by naive subjects. We analyzed linear segmentations of 20 narratives performed by naive subjects (7 new subjects per narrative), where speaker intention was the segment criterion. Subjects were given transcripts, asked to place a new segment boundary between lines (prosodic phrases)1 wherever the speaker had a new communicative goal, and to briefly describe the completed segment. Subjects were free to assign any number of boundaries. The qualitative results were that segments varied in size from 1 to 49 phrases in length (Avg.=5.9), and the rate at which subjects assigned boundaries ranged from 5.5% to 41.3%. Despite this variation, we found statistically significant agreement among subjects across all narratives on location of segment boundaries (.114 x 10^-6 < p < .6 x 10^-6).

We then looked at the predictive power of linguistic cues for identifying the segment boundaries agreed upon by a significant number of subjects. We used three distinct algorithms based on the distribution of referential noun phrases, cue words, and pauses, respectively. Each algorithm (NP-A, CUE-A, PAUSE-A) was designed to replicate the subjects’ segmentation task (break up a narrative into contiguous segments, with segment breaks falling between prosodic phrases). NP-A used three features, while CUE-A and PAUSE-A each made use of a single feature. The features are a subset of those described in section 3.

To evaluate how well an algorithm predicted segmental structure, we used the information retrieval (IR) metrics described in section 3. As reported in (Passonneau and Litman, to appear), we also evaluated a simple additive method for combining algorithms in which a boundary is proposed if each separate algorithm proposes a boundary. We tested all pairwise combinations, and the combination of all three algorithms. No algorithm or combination of algorithms performed as well as humans. NP-A performed better than the other unimodal algorithms, and a combination of NP-A and PAUSE-A performed best. We felt that significant improvements could be gained by combining the input features in more complex ways rather than by simply combining the outputs of independent algorithms.

3 Methodology

3.1 Boundary Classification

We represent each narrative in our corpus as a sequence of potential boundary sites, which occur between prosodic phrases. We classify a potential boundary site as boundary if it was identified as such by at least 3 of the 7 subjects in our earlier study. Otherwise it is classified as non-boundary. Agreement among subjects on boundaries was significant at below the .02% level for values of j ≥ 3, where j is

1 We used Chafe’s (1980) prosodic analysis.
...Because he's looking at the girl.

Figure 1: Excerpt from narr. 6, with boundaries.

the number of subjects (1 to 7), on all 20 narratives.²

Fig. 1 shows a typical segmentation of one of the narratives in our corpus. Each line corresponds to a prosodic phrase, and each space between the lines corresponds to a potential boundary site. The bracketed numbers will be explained below. The boxes in the figure show the subjects' responses at each potential boundary site, and the resulting boundary classification. Only 2 of the 7 possible boundary sites are classified as boundary.

3.2 Coding of Linguistic Features

Given a narrative of n prosodic phrases, the n-1 potential boundary sites are between each pair of prosodic phrases $P_i$ and $P_{i+1}$, i from 1 to n-1. Each potential boundary site in our corpus is coded using the set of linguistic features shown in Fig. 2.

Values for the prosodic features are obtained by automatic analysis of the transcripts, whose conventions are defined in (Chafe, 1980) and illustrated in Fig. 1: "" and "" indicate sentence-final intonational contours; "" indicates phrase-final but not sentence final intonation; "[X]" indicates a pause lasting X seconds; "..." indicates a break in timing too short to be measured. The features before and after depend on the final punctuation of the phrases $P_i$ and $P_{i+1}$, respectively. The value is ‘+sentence.final.contour’ if "" or "" , -sentence.final.contour’ if "". Pause is assigned ‘true’ if $P_{i+1}$ begins with [X], ‘false’ otherwise. Duration is assigned X if pause is ‘true’, 0 otherwise.

The cue phrase features are also obtained by automatic analysis of the transcripts. Cue1 is assigned ‘true’ if the first lexical item in $P_{i+1}$ is a member of the set of cue words summarized in (Hirschberg and Litman, 1993). Word1 is assigned this lexical item if

²We previously used agreement by 4 subjects as the threshold for boundaries; for $j \geq 4$, agreement was significant at the .01% level. (Passonneau and Litman, 1993)

• Prosodic Features
  - before:+sentence.final.contour,-sentence.final.contour
  - after: +sentence.final.contour,-sentence.final.contour
  - pause: true, false.
  - duration: continuous.

• Cue Phrase Features
  - cue1: true, false.
  - word1: also, and, anyway, basically, because, but, finally, first, like, meanwhile, no, now, oh, okay, only, or, see, so, then, well, where, NA.
  - cue2: true, false.
  - word2: and, anyway, because, boy, but, now, okay, or, right, so, still, then, NA.

• Noun Phrase Features
  - coref: +coref, -coref, NA.
  - infer: +infer, -infer, NA.
  - global.pro: +global.pro, -global.pro, NA.

• Combined Feature
  - cue-prosody: complex, true, false.

Figure 2: Features and their potential values.

cue1 is true, ‘NA’ (not applicable) otherwise.³ Cue2 is assigned ‘true’ if cue1 is true and the second lexical item is also a cue word. Word2 is assigned the second lexical item if cue2 is true, ‘NA’ otherwise.

Two of the noun phrase (NP) features are hand-coded, along with functionally independent clauses (FICs), following (Passonneau, 1994). The two authors coded independently and merged their results. The third feature, global.pro, is computed from the hand coding. FICs are tensed clauses that are neither verb arguments nor restrictive relatives. If a new FIC (Cj) begins in prosodic phrase $P_{i+1}$, then NPs in Cj are compared with NPs in previous clauses and the feature values assigned as follows⁴:

1. coref = +coref’ if Cj contains an NP that co-

2. infer = +infer’ if Cj contains an NP whose refer-

3. global.pro = +global.pro’ if Cj contains a defi-

4The NP algorithm can assign multiple boundaries within one prosodic phrase if the phrase contains multiple clauses; these very rare cases are normalized (Pas-

5The cue phrases that occur in the corpus are shown as potential values in Fig. 2.

6The NP algorithm can assign multiple boundaries within one prosodic phrase if the phrase contains multiple clauses; these very rare cases are normalized (Passonneau and Litman, 1993).
...Because he's looking at the girl.  
[75] (ZERO-PRONOUN) Falls over,

| before | after | pause | duration | cue₁ | word₁ | cue₂ | word₂ | coref | infer | global.pro | cue-prosody |
|--------|-------|-------|----------|------|-------|------|-------|-------|-------|-------------|-------------|
| +s.f.c | -s.f.c | true  | .75      | false | NA    | false | NA    | +     | -     | +           | true        |

Figure 3: Example feature coding of a potential boundary site.

1. before = ‘+sentence.final.contour’
2. pause = ‘true’
3. And either:
   (a) cue₁ = ‘true’, word₁ ≠ ‘and’
   (b) cue₁ = ‘true’, word₁ = ‘and’, cue₂ = ‘true’, word₂ ≠ ‘and’

Else, cue-prosody has the same values as pause.

Fig. 3 illustrates how the first boundary site in Fig. 1 would be coded using the features in Fig. 2. The prosodic and cue phrase features were motivated by previous results in the literature. For example, phrases beginning discourse segments were correlated with preceding pause duration in (Grosz and Hirschberg, 1992; Hirschberg and Grosz, 1992). These and other studies (e.g., (Hirschberg and Litman, 1993)) also found it useful to distinguish between sentence and non-sentence final intonational contours. Initial phrase position was correlated with discourse signaling uses of cue words in (Hirschberg and Litman, 1993); a potential correlation between discourse signaling uses of cue words and adjacency patterns between cue words was also suggested. Finally, (Litman, 1994) found that treating cue phrases individually rather than as a class enhanced the results of (Hirschberg and Litman, 1993).

Passonneau (to appear) examined some of the few claims relating discourse anaphoric noun phrases to global discourse structure in the Pear corpus. Results included an absence of correlation of segmental structure with centering (Grosz et al., 1983; Kameyama, 1986), and poor correlation with the contrast between full noun phrases and pronouns. As noted in (Passonneau and Litman, 1993), the NP features largely reflect Passonneau’s hypotheses that adjacent utterances are more likely to contain expressions that corefer, or that are inferentially linked, if they occur within the same segment; and that a definite pronoun is more likely than a full NP to refer to an entity that was mentioned in the current segment, if not in the previous utterance.

3.3 Evaluation

The segmentation algorithms presented in the next two sections were developed by examining only a training set of narratives. The algorithms are then evaluated by examining their performance in predicting segmentation on a separate test set. We currently use 10 narratives for training and 5 narratives for testing. (The remaining 5 narratives are reserved for future research.) The 10 training narratives range in length from 51 to 162 phrases (Avg.=101.4), or from 38 to 121 clauses (Avg.=76.8). The 5 test narratives range in length from 47 to 113 phrases (Avg.=87.4), or from 37 to 101 clauses (Avg.=69.0). The ratios of test to training data measured in narratives, prosodic phrases and clauses, respectively, are 50.0%, 43.1% and 44.9%. For the machine learning algorithm we also estimate performance using cross-validation (Weiss and Kulikowski, 1991), as detailed in Section 5.

To quantify algorithm performance, we use the information retrieval metrics shown in Fig. 4. Recall is the ratio of correctly hypothesized boundaries to target boundaries. Precision is the ratio of hypothesized boundaries that are correct to the total hypothesized boundaries. (Cf. Fig. 4 for fallout and error.) Ideal behavior would be to identify all and only the target boundaries: the values for b and c in Fig. 4 would thus both equal 0, representing no errors. The ideal values for recall, precision, fallout, and error are 1, 1, 0, and 0, while the worst values are 0, 0, 1, and 1. To get an intuitive summary of overall performance, we also sum the deviation of the observed value from the ideal value for each metric: (1-recall) + (1-precision) + fallout + error. The summed deviation for perfect performance is thus 0.

Finally, to interpret our quantitative results, we use the performance of our human subjects as a target goal for the performance of our algorithms (Gale et al., 1992). Table 1 shows the average human performance for both the training and test sets of narratives. Note that human performance is basically the same for both sets of narratives. However, two
factors prevent this performance from being closer to ideal (e.g., recall and precision of 1). The first is the wide variation in the number of boundaries that subjects used, as discussed above. The second is the inherently fuzzy nature of boundary location. We discuss this second issue at length in (Passonneau and Litman, to appear), and present relaxed IR metrics that penalize near misses less heavily in (Litman and Passonneau, 1995).

4 Hand Tuning

To improve performance, we analyzed the two types of IR errors made by the original NP algorithm (Passonneau and Litman, 1993) on the training data. Type “b” errors (cf. Fig. 4), mis-classification of non-boundaries, were reduced by changing the coding features pertaining to clauses and NPs. Most “b” errors correlated with two conditions used in the NP algorithm, identification of clauses and of inferential links. The revision led to fewer clauses (more assignments of ‘NA’ for the three NP features) and more inference relations. One example of a change to clause coding is that formulaic utterances having the structure of clauses, but which function like interjections, are no longer recognized as independent clauses. These include the phrases let’s see, let me see, I don’t know, you know when they occur with no verb phrase argument. Other changes pertained to sentence fragments, unexpected clausal arguments, and embedded speech.

Three types of inference relations linking successive clauses (C_i-1, C_i) were added (originally there were 5 types (Passonneau, 1994)). Now, a pronoun (e.g., it, that, this) in C_i referring to an action, event or fact inferrable from C_i-1 links the two clauses. So does an implicit argument, as in Fig. 5, where the missing argument of notice is inferred to be the event of the pears falling. The third case is where an NP in C_i is described as part of an event that results directly from an event mentioned in C_i-1.

“C” type errors (cf. Fig. 4), mis-classification of boundaries, often occurred where prosodic and cue features conflicted with NP features. The original NP algorithm assigned boundaries wherever the three values ‘-coref’, ‘-infer’, ‘-global.pro’ (defined in section 3) co-occurred, represented as the first conditional statement of Fig. 6. Experiments led to the hypothesis that the most improvement came by assigning a boundary if the cue-prosody feature had the value ‘complex’, even if the algorithm would not otherwise assign a boundary, as shown in Fig. 6.

\[ \text{if (coref} = \text{-coref and infer} = \text{-infer and global.pro} = \text{-global.pro}) \]
\[ \text{then boundary} \]
\[ \text{else if cue-prosody} = \text{complex then boundary} \]
\[ \text{else non-boundary} \]

Figure 6: Condition 2 algorithm.

We refer to the original NP algorithm applied to the initial coding as Condition 1, and the tuned algorithm applied to the enriched coding as Condition 2. Table 2 presents the average IR scores across the narratives in the training set for both conditions. Reduction of “b” type errors raises precision, and lowers fallout and error rate. Reduction of “c” type errors raises recall, and lowers fallout and error rate. All scores improve in Condition 2, with precision and fallout showing the greatest relative improvement. The major difference from human performance is relatively poorer precision.

The standard deviations in Table 2 are often close to 1/4 or 1/3 of the reported averages. This indicates a large amount of variability in the data, reflecting wide differences across narratives (speakers) in the training set with respect to the distinctions recognized by the algorithm. Although the high standard deviations show that the tuned algorithm is not well fitted to each narrative, it is likely that it is overspecialized to the training sample in the sense that test narratives are likely to exhibit further variation.

Table 3 shows the results of the hand tuned algorithm on the 5 randomly selected test narratives on both Conditions 1 and 2. Condition 1 results, the untuned algorithm with the initial feature set, are very similar to the training set except for worse precision. Thus, despite the high standard deviations, 10 narratives seems to have been a sufficient sample size for evaluating the initial NP algorithm. Condition 2 results are better than condition 1 in Table 3, and condition 1 in Table 2. This is strong evidence that the tuned algorithm is a better predictor of segment boundaries than the original NP algorithm. Nevertheless, the test results of condition 2 are much worse than the corresponding training results, particularly for precision (.44 versus .62). This

| Average | Recall | Prec | Fall | Error | SumDev |
|---------|--------|------|------|-------|--------|
| Condition 1 | .42    | .40  | .14  | .22   | 1.54   |
| Std. Dev. | .17    | .12  | .06  | .07   | .34    |
| Condition 2 | .58    | .62  | .08  | .14   | 1.02   |
| Std. Dev. | .14    | .10  | .04  | .05   | .18    |

Table 2: Performance on training set.

| Average | Recall | Prec | Fall | Error | SumDev |
|---------|--------|------|------|-------|--------|
| Condition 1 | .44    | .29  | .18  | .21   | 1.64   |
| Std. Dev. | .16    | .17  | .07  | .06   | .32    |
| Condition 2 | .50    | .44  | .11  | .17   | 1.34   |
| Std. Dev. | .21    | .06  | .03  | .04   | .29    |

Table 3: Performance on test set.
confirms that the tuned algorithm is over calibrated to the training set.

5 Machine Learning

We use the machine learning program C4.5 (Quinlan, 1993) to automatically develop segmentation algorithms from our corpus of coded narratives, where each potential boundary site has been classified and represented as a set of linguistic features. The first input to C4.5 specifies the names of the classes to be learned (boundary and non-boundary), and the names and potential values of a fixed set of coding features (Fig. 2). The second input is the training data, i.e., a set of examples for which the class and feature values (as in Fig. 3) are specified. Our training set of 10 narratives provides 1004 examples of potential boundary sites. The output of C4.5 is a classification algorithm expressed as a decision tree, which predicts the class of a potential boundary given its set of feature values.

Because machine learning makes it convenient to induce decision trees under a wide variety of conditions, we have performed numerous experiments, varying the number of features used to code the training data, the definitions used for classifying a potential boundary site as boundary or non-boundary and the options available for running the C4.5 program. Fig. 7 shows one of the highest-performing learned decision trees from our experiments. This decision tree was learned under the following conditions: all of the features shown in Fig. 2 were used to code the training data, boundaries were classified as discussed in section 3, and C4.5 was run using only the default options. The decision tree predicts the class of a potential boundary site based on the features before, after, duration, cue1, word1, coref, infer, and global.pro. Note that although not all available features are used in the tree, the included features represent 3 of the 4 general types of knowledge (prosody, cue phrases and noun phrases). Each level of the tree specifies a test on a single feature, with a branch for every possible outcome of the test. A branch can either lead to the assignment of a class, or to another test. For example, the tree initially branches based on the value of the feature before. If the value is -sentence. final. contour then the first branch is taken and the potential boundary site is assigned the class non-boundary. If the value of before is +sentence. final. contour then the second branch is taken and the feature coref is tested.

The performance of this learned decision tree averaged over the 10 training narratives is shown in Table 4, on the line labeled “Learning 1”. The line labeled “Learning 2” shows the results from another machine learning experiment, in which one of the default C4.5 options used in “Learning 1” is overridden. The “Learning 2” tree (not shown due to space restrictions) is more complex than the tree of Fig. 7, but has slightly better performance. Note that “Learning 1” performance is comparable to human performance (Table 1), while “Learning 2” is slightly better than humans. The results obtained via machine learning are also somewhat better than the results obtained using hand tuning—particularly with respect to precision (“Condition 2” in Table 2), and are a great improvement over the original NP results (“Condition 1” in Table 2).

The performance of the learned decision trees averaged over the 5 test narratives is shown in Table 5. Comparison of Tables 4 and 5 shows that, as with the hand tuning results (and as expected), average performance is worse when applied to the testing rather than the training data particularly with respect to precision. However, performance is an improvement over our previous best results (“Condition 1” in Table 3), and is comparable to (“Learning 1”) or very slightly better than (“Learning 2”) the hand tuning results (“Condition 2” in Table 3).

We also use the resampling method of cross-validation (Weiss and Kulikowski, 1991) to estimate performance, which averages results over multiple partitions of a sample into test versus training data. We performed 10 runs of the learning program, each using 9 of the 10 training narratives for that run's
from scratch and thus each training and testing set
understanding how to best code the data, while ms-
suggests that hand tuning is a useful method for
features. The first method hand tunes features and
We have presented two methods for developing seg-
6 Conclusion

Because we have already used cross-validation, we
do not anticipate significant degradation on new test
narratives. An important area for future research is
to develop principled methods for identifying dis-

tinct speaker strategies pertaining to how they si-
gnal segments. Performance of individual speakers
varies widely as shown by the high standard devia-
tions in our tables. The original NP, hand tuned, and
machine learning algorithms all do relatively poorly
on narrative 16 and relatively well on 11 (both in the
test set) under all conditions. This lends sup-
port to the hypothesis that there may be consistent
differences among speakers regarding strategies for
signaling shifts in global discourse structure.

| Average | Recall | Prec | Fall | Error | SumDev |
|---------|--------|------|------|-------|--------|
| Learning 1 | .54    | .76  | .04  | .11   | .85    |
| Std. Dev. | .18    | .12  | .02  | .04   | .28    |
| Learning 2 | .59    | .78  | .03  | .10   | .76    |
| Std. Dev. | .22    | .12  | .02  | .04   | .29    |

Table 4: Performance on training set.

| Average | Recall | Prec | Fall | Error | SumDev |
|---------|--------|------|------|-------|--------|
| Learning 1 | .43    | .48  | .08  | .16   | 1.34   |
| Std. Dev. | .21    | .13  | .03  | .05   | .26    |
| Learning 2 | .47    | .50  | .09  | .16   | 1.27   |
| Std. Dev. | .18    | .16  | .04  | .07   | .42    |

Table 5: Performance on test set.

| Average | Recall | Prec | Fall | Error | SumDev |
|---------|--------|------|------|-------|--------|
| Learning 1 | .48    | .63  | .05  | .15   | 1.14   |
| Std. Dev. | .19    | .16  | .03  | .03   | .24    |
| Learning 2 | .46    | .61  | .07  | .15   | 1.15   |
| Std. Dev. | .20    | .14  | .04  | .03   | .21    |

Table 6: Using 10-fold cross-validation.

training set (for learning the tree) and the remaining
narrative for testing. Note that for each iteration of
the cross-validation, the learning process begins
from scratch and thus each training and testing set are
still disjoint. While this method does not make
sense for humans, computers can truly ignore pre-
vious iterations. For sample sizes in the hundreds
(our 10 narratives provide 1004 examples) 10-fold
cross-validation often provides a better performance
estimate than the hold-out method (Weiss and Ku-
likowski, 1991). Results using cross-validation are
shown in Table 6, and are better than the estimates
obtained using the hold-out method (Table 5), with
the major improvement coming from precision. Be-
cause a different tree is learned on each iteration, the
cross-validation evaluates the learning method, not
a particular decision tree.

6 Conclusion

We have presented two methods for developing seg-
mentation hypotheses using multiple linguistic fea-
tures. The first method hand tunes features and
algorithms based on analysis of training errors. The
second method, machine learning, automatically in-
duces decision trees from coded corpora. Both meth-
ods rely on an enriched set of input features com-
pared to our previous work. With each method, we
have achieved marked improvements in performance
compared to our previous work and are approaching
human performance. Note that quantitatively, the
machine learning results are slightly better than the
hand tuning results. The main difference on average
performance is the higher precision of the automated
algorithm. Furthermore, note that the machine learn-
ing algorithm used the changes to the coding fea-
tures that resulted from the tuning methods. This
suggests that hand tuning is a useful method for
understanding how to best code the data, while ma-
chine learning provides an effective (and automatic)
way to produce an algorithm given a good feature representation.

Our results lend further support to the hypothesis
that linguistic devices correlate with discourse structure (cf. section 2.1), which itself has practical im-
port. Understanding systems could infer segments
as a step towards producing summaries, while ge-
generation systems could signal segments to increase
comprehensibility.7 Our results also suggest that to
best identify or convey segment boundaries, systems
will need to exploit multiple signals simultaneously.

We plan to continue our experiments by further
merging the automated and analytic techniques, and
evaluating new algorithms on our final test corpus.

References

Leo Breiman, Jerome Friedman, Richard Olshen,
and C. Stone. 1984. Classification and Regression
Trees. Wadsworth and Brooks, Monterey, CA.

Wallace L. Chafe. 1980. The Pear Stories. Abex
Publishing Corporation, Norwood, CA.

William Gale, Ken W. Church, and David Yarow-
sky. 1992. Estimating upper and lower bounds
on the performance of word-sense disambiguation
programs. In Proc. of the 30th ACL, pages 249-
256.

Barbara Grosz and Julia Hirschberg. 1992. Some
intonational characteristics of discourse structure.
In Proc. of the International Conference on Spok-
en Language Processing.

Barbara Grosz and Candace Sidner. 1986. Atten-
tion, intentions and the structure of discourse.
Computational Linguistics, 12:175-204.

Barbara J. Grosz, Aaravind K. Joshi, and Scott
Weinstein. 1983. Providing a unified account of
definite noun phrases in discourse. In Proc. of the
21st ACL, pages 44-50.

7 Cf. (Hirschberg and Pierrehumbert, 1986) who argue
that comprehensibility improves if units are prosodically
signaled.
Julia Hirschberg and Barbara Grosz. 1992. Intonation.

Johanna D. Moore and Martha E. Pollack. 1992. Experimental studies on the disambiguation of cue phrases. Computational Linguistics, 19(3):501–530.

Julia Hirschberg and Janet Pierrehumbert. 1986. The intonational structuring of discourse. In Proc. of the 24th ACL.

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

Amy Isard and Jean Carletta. 1995. Replicability of transaction and action coding in the map task corpus. In AAAI 1995 Spring Symposium Series: Empirical Methods in Discourse Interpretation and Generation, pages 60–66.

Megumi Kameyama. 1986. A property-sharing constraint in centering. In Proc. of the 24th ACL, pages 200–206.

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

Alex Lascarides and Jon Oberlander. 1992. Temporal coherence and defeasible knowledge. Theoretical Linguistics.

Charlotte Linde. 1979. Focus of attention and the choice of pronouns in discourse. In Talmy Givon, editor, Syntax and Semantics: Discourse and Syntax, pages 337–354. Academic Press, New York.

Diane J. Litman and Rebecca J. Passonneau. 1995. Developing algorithms for discourse segmentation. In AAAI 1995 Spring Symposium Series: Empirical Methods in Discourse Interpretation and Generation, pages 85–91.

Diane J. Litman. 1994. Classifying cue phrases in text and speech using machine learning. In Proc. of the 12th AAAI, pages 806–813.

William C. Mann and Sandra Thompson. 1988. Rhetorical structure theory. TEXT, pages 243–281.

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

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

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

Megan Moser and Julia D. Moore. 1995. Using discourse analysis and automatic text generation to study discourse cue usage. In AAAI 1995 Spring Symposium Series: Empirical Methods in Discourse Interpretation and Generation, pages 92–98.

Christine H. Nakatani, Julia Hirschberg, and Barbara J. Grosz. 1995. Discourse structure in spoken language: Studies on speech corpora. In AAAI 1995 Spring Symposium Series: Empirical Methods in Discourse Interpretation and Generation, pages 106–112.

Rebecca J. Passonneau and Diane J. Litman. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. In Proc. of the 31st ACL, pages 148–155.

Rebecca J. Passonneau and D. Litman. to appear. Empirical analysis of three dimensions of spoken discourse. In E. Hovy and D. Scott, editors, Interdisciplinary Perspectives on Discourse. Springer Verlag, Berlin.

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

Rebecca J. Passonneau. to appear. Interaction of the segmental structure of discourse with explicitness of discourse anaphora. In E. Prince, A. Joshi, and M. Walker, editors, Proc. of the Workshop on Centering Theory in Naturally Occurring Discourse. Oxford University Press.

Livya Polanyi. 1988. A formal model of discourse structure. Journal of Pragmatics, pages 601–638.

John R. Quinlan. 1993. C4.5 : Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, Calif.

Rachel Reichman. 1985. Getting Computers to Talk Like You and Me: Discourse Context, Focus, and Semantics. Bradford. MIT, Cambridge.

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

Lisa J. Stiffelman. 1995. A discourse analysis approach to structured speech. In AAAI 1995 Spring Symposium Series: Empirical Methods in Discourse Interpretation and Generation, pages 162–167.

Bonnie L. Webber. 1991. Structure and ostension in the interpretation of discourse deixis. Language and Cognitive Processes, pages 107–135.

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