EVALUATING HIERARCHICAL DISCOURSE SEGMENTATION OF EXPOSITORY SPEECH

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

in

Computational Linguistics

---

by

Lucien Carroll

Aug 2010

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Lucien Carroll:

Evaluating Hierarchical Discourse Segmentation of Expository Speech

---

Robert Malouf, Chair
Department of Linguistics and Oriental Languages

Eniko Csomay
Department of Linguistics and Oriental Languages

Joseph Lewis
Department of Computer Science

Approval Date

© Copyright 2010
by
Lucien Carroll

ABSTRACT OF THE THESIS

Evaluating Hierarchical Discourse Segmentation of Expository Speech
by
Lucien Carroll
Master of Arts in Computational Linguistics
San Diego State University, 2010

There is a large body of literature describing work in linear discourse segmentation, especially of news data, and some work describing algorithms for hierarchical discourse segmentation. However, little work has been done on segmenting more conversational genres, and even less on evaluating hierarchical segmentation. I describe a method for compiling a gold standard for tree segmentation of expository monolog, and I propose an error metric. I then evaluate two hierarchical segmentation algorithms with that metric. The segmentation algorithms both perform quite poorly on this language variety, but one of the two is shown to be significantly better than baseline segmentations.

TABLE OF CONTENTS

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

ACKNOWLEDGEMENTS

This work has been enriched by the questions and advice of many people, especially Eniko Csomay and Rob Malouf, but also Lara Taylor, Rebecca Colavin, Andy Kehler, Emily Medina, Paul Kalmar and Erin O'Connor. This thesis would also not have been completed if it were not for the encouragement of Angie Jin. I am also grateful to Freddy Choi for making his code and data available.

CHAPTER 1

Introduction

Discourse segmentation is the task of identifying coherent clusters of sentences and the transitions between them. Because of its close relationship to discourse coherence, discourse segmentation can be viewed as shallow discourse parsing. In discourse segmentation, the segments and the relations between them are left unlabeled, focusing instead on the boundaries between the segments (i.e., the bracketing). Much of the work to date has focused on linear discourse segmentation, in which segments are non-overlapping and sequential. In contrast, the work in discourse structure and discourse parsing is explicitly hierarchical. I focus here on hierarchical discourse segmentation, in which larger segments subsume sets of subsegments. Hierarchical segmentation thus differs from linear segmentation in that it includes segment information at multiple layers of granularity, and it differs from discoursing parsing in that it focuses on large scale structure rather than the discourse structure between individual sentences and within sentences. Hierarchical segmentation information complements what is captured in discourse parsing, and it is potentially more informative and more faithful to linguistic theory than linear segmentation is, but it poses a more challenging evaluation problem.

1.1 Benefits of hierarchical segmentation

Discourse structure has variously been defined in terms of communicative intention, attention, topic/subtopic structure, coherence relations, and cohesive devices (Grosz, Joshi, & Weinstein, 1995; Marcu, 2000; Kehler, 2002; Asher & Lascarides, 2003; Webber, 2004; Polanyi, Culy, Berg, Thione, & Ahn, 2004). It is also not clear whether discourse is best modeled as a sequence (Hearst, 1994), a tree (Marcu, 2000), a directed acyclic graph (Danlos, 2004), or a general graph (Wolf & Gibson, 2004), and different genres and different definitions of discourse structure are likely different in this regard. Newswire data like that used in the Topic Detection and Tracking (TDT) tasks (Allan, 2002) lends itself to a sequence model. Narrative is more complex while still largely sequential **[cite]**, while expository text is often associated with tree-like outlines **[cite]**, and conversation and computer-mediated chat suggest more complex structures **[cite]**. Hierarchical segmentation may not be able to accurately represent the more complex graph structures, but the hierarchical segmentation can at least represent a greater portion of these structures than the sequence model of linear segmentation can.

Sufficiently robust discourse parsers would render obsolete the utility of discourse segmentation algorithms, since a segmentation could be easily derived from such parses just as shallow syntactic parses can be derived from full parses. However, automatic discourse parsing has so far focused on small texts, on local coherence scales where cue phrases, co-reference and verbal morphology are most helpful. On the other hand, hierarchical segmentation indicates global coherence but not local coherence. The segmentation systems could serve well as guides to parsing systems, or they could be combined as complementary subsystems in a larger system. In any case, automatic discourse parsing does not yet provide the large-scale discourse structure that discourse segmentation targets.

Discourse segmentation is thought to facilitate automatic summarization (Angheluta, De Busser, & Moens, 2002; Boguraev & Neff, 2000), information retrieval (Kaszkiel & Zobel, 1997), co-reference and anaphora resolution (Walker, 1997; Cristea, Ide, Marcu, & Tablan, 2000) and question answering (Chai & Jin, 2004). Automatic discourse segmentation, as shallow annotation of discourse structure, also provides a testing grounds for linguistic theories of discourse (Passonneau & Litman, 1997) and provides a principled partitioning of texts in linguistic corpora (Biber, Csomay, Jones, & Keck, 2004).

1.2 HIERARCHICAL SEGMENTATION IS DIFFICULT AND POORLY EXPLORED

Studies of linear segmentation have revealed that discourse boundaries are inherently fuzzy. Human annotators demonstrate frequent disagreement about where the transitions between segments occur, while still demonstrating statistically significant agreement (Passonneau & Litman, 1997). Because of this, 'near misses' should be treated much the same as complete matches, but conventional precision and recall measures penalize them. The crossing-bracket measure (Carbone, Gal, Shieber, & Grosz, 2004) is more forgiving, but still over-penalizes near misses and furthermore favors sparse bracketing. An error measure P_k proposed by Beeferman, Berger, and Lafferty (1999) and modified by Pevzner and Hearst (2001) compensates for the variation in boundary locations, but no comparable measure has been proposed for hierarchical segmentation.

Four studies have described hierarchical discourse segmentation algorithms, but none of them rigorously evaluated the segmentation in its hierarchical form. Yaari used a hierarchical clustering algorithm for hierarchical discourse segmentation, and to evaluate it, he linearized the tree (taking all boundaries equally) and compared the resulting precision and recall to contemporary linear segmentation algorithms. Slaney and Ponceleon used scale-space segmentation (an image segmentation algorithm) on the discourse's trajectory in a Latent Semantic Indexing (LSI) space (Landauer, Foltz, & Laham, 1998). They evaluated the algorithm by visual comparison with the heading-subheading structure of the text. Angheluta

et al. applied a linear discourse segmentation algorithm recursively, segmenting each major segment into a sequence of subsegments. They used the result in a summarization system, and they evaluated the summarization system but not the segmentation itself. Eisenstein used a Bayesian latent topic model to find a hierarchical segmentation, and he comes the closest to quantitative evaluation of the whole segmentation. He evaluated it against three recursive segmentation algorithms on a corpus that had just two levels of segment depth and considers these two levels as separate and equally important. While each of these studies offers some insight into the validity of the hierarchical segmentation, none of these evaluation methods directly and quantitatively assesses the hierarchical segmentation as a whole.

A few state-of-the-art linear segmentation algorithms also use hierarchical clustering or matrix partitioning, making them applicable to hierarchical segmentation with only trivial modification. For example, the C99 algorithm (Choi, 2000) applies contrast enhancement and divisive clustering to a matrix of lexical vector cosine similarities. The CWM algorithm (Choi, Wiemer-Hastings, & Moore, 2001) applies the same procedure to a similarity matrix of LSI vectors. Using these algorithms for hierarchical segmentation simply requires keeping record of the order of the cluster splits, but until now they have only been used for linear segmentation.

1.3 EXPOSITORY SPEECH

To date, segmentation research has also focused overwhelmingly on news, and after that on expository and narrative texts. For newswire in particular, the sequence model of linear segmentation may be a good approximation of the discourse structure. In contrast, the linguistic theories have focused on expository text, spoken narrative, and conversation. This study focuses on monologic academic lectures—expository speech. We would expect it, as expository discourse, to share the characteristic hierarchical structure that has often been attributed to expository text. As unscripted spoken language, however, it should share some of the computationally difficult and cognitively revealing features demonstrated by less planned and more informal discourse such as conversation and chat. Using monologic lectures thus extends segmentation research toward more complicated discourse structures without addressing all the issues at once.

One outstanding issue in using spoken language data is the absence of explicit discourse boundaries. Newswire data has headlines and expository text often has headings, but spoken data has no such indicators of discourse boundaries. One approach is to have trained experts annotate discourse structures, such as in the case of the Boston Directions Corpus (BDC) (Hirschberg & Nakatani, 1996). Even expert annotators disagree significantly, but these annotations can be combined via annotator discussion or algorithmically (Carbone et al., 2004). Passonneau and Litman (1996) provide a method for getting a linear segmentation

gold standard for spoken narrative, using agreement among untrained annotators. The method requires more readers but is (arguably) more theory-neutral. Moreover, the manner in which the statistical significance of inter-annotator agreement is used suggests hierarchical structure, but it has some issues that become more problematic in hierarchical segmentation.

1.4 THIS STUDY

Building on work in evaluating linear segmentation, this study considers the evaluation of tree segmentations. I propose a method, similar to Passonneau and Litman's, for obtaining a gold standard tree segmentation from human annotation, and I propose an error measure, derived from the Beeferman measure, for evaluating the alignment of a tree segmentation to a reference segmentation. I then evaluate three hierarchical segmentation algorithms against the gold standard derived from human annotation.

In the following chapter, I review the linguistic literature on discourse structure, discuss the issues in evaluating discourse segmentation and some proposed solutions, and review some of the segmentation algorithms proposed in the literature. In chapter 3, I describe the method I use for obtaining a gold standard tree segmentation for lecture data, and in chapter 4 I introduce the error measure and demonstrate its use on a linear segmentation corpus. In chapter 5, I evaluate two hierarchical segmentation algorithms against the gold standard tree segmentation, and the discussion and conclusion follow that.

CHAPTER 2

Discourse Structure and Segmentation

[Discourse structure and linear segmentation sections are currently hidden]

2.0.1 Hierarchical Segmentation

Four studies have described algorithms for hierarchical discourse segmentation. However, none of them rigorously evaluated the segmentation in its hierarchical form, because there has been no evaluation metric suitable to the task.

Yaari (1997), interested in document retrieval and summarization, applied hierarchical agglomerative clustering to the lexical vectors in expository texts. Open class words were stemmed and weighted according to a tf-idf scheme, and similarity was judged by the cosine metric. Yaari discusses how the hierarchical tree of the clustering reflects the hierarchical structure of the discourse. However, because the similarity tree apparently does not directly correspond to the discourse tree, the tree is converted to a linear segmentation not by a simple tree cut but by a pair of rules that balance a minimum size constraint with a constraint about the internal similarity of the resulting segments. This linear segmentation is evaluated on one expository text from Hearst's magazine article data set, using precision and recall metrics.

Slaney and Ponceleon (2001) were interested in creating tables of contents for indexing video, and as part of a system that segmented video based on audio and visual cues, they developed an algorithm that creates a hierarchical segmentation of transcribed language. The algorithm uses Gaussian smoothing to filter out high signal frequencies, smoothing the discourses' trajectory in LSI space. Maxima in the velocity of this trajectory tend to indicate topic changes. Finding these maxima on a wide range of scales, by varying the size of the Gaussian window, amounts to identifying the location of topic changes and their relative importance. They evaluate their algorithm on two texts, one a chapter scanned from a technical book, and the other a manual transcription of 30 minutes of broadcast news. They acknowledge the lack of a quantitative measure for comparing fuzzy hierarchical structures, and simply plot the outline's heading and subheadings beside the boundaries found by the algorithm.

Angheluta et al. (2002) include a hierarchical segmentation system in a document summary extraction system. They do not describe their algorithm in detail, but it makes use of lexical chains (recurring semantically related words) and gives special attention to terms in

the grammatical topic of sentences. The segmentation is heuristic, and results in nested segments, each associated with one term. They do not evaluate the segmentation itself, but report results comparing their summarization system to other groups competing in DUC 2002 (Over & Liggett, 2002), a task focused on summarizing news stories.

Eisenstein (2009) used a Bayesian latent topic model to find a hierarchical segmentation, where certain topics and their words are assumed to belong to high-level segments and other topics and their words are assumed to belong to lower-level segments. He comes the closest to quantitative evaluation of the whole segmentation. He evaluated the hierarchical Bayesian algorithm against three recursive segmentation algorithms on a corpus that had just two levels of segment depth and considers these two levels as separate and equally important.

2.1 Discourse Segmentation Evaluation

2.1.1 Annotators disagree

A recurring problem in developing algorithms, error measures and gold standards for discourse segmentation is that discourse boundaries are inherently vague. As Passonneau and Litman (1996, 1997) describe, human annotators disagree about the number of boundaries and the precise location of the boundaries. In spite of the disagreement, annotator agreement does reach very high values of statistical significance for some boundaries. Possible causes of the disagreement include differences in the understood inferential ties, difficulty of converting a more complex discourse graph to a sequence, and differences in understanding of the segmentation task itself.

In the methodology Passonneau and Litman (1996) describe, untrained readers were asked to examine transcribed narratives and mark points (restricted to prosodic boundaries) at which the narrator's purpose changed. The seven readers per narrative produced widely varying numbers of segments. The observed distribution of boundary judgments was compared, using Cochran's Q (Cochran, 1950), to a distribution created by assuming the seven readers each had a predetermined number of boundaries to mark, but that the particular boundaries were chosen randomly. In all narratives, boundaries were highly statistically significant if 4 or more readers marked a boundary. They note, however, that their methodology misses some boundaries where, for example, two annotators marked one prosodic boundary and three others marked the next prosodic boundary, at an episodic transition in the narrative. Under the exact agreement criterion, no boundary is assigned, even though 5 readers agree there is a discourse boundary somewhere close.

2.1.2 The Beeferman error measure

The fuzziness of segment boundaries also makes measuring performance difficult. Conventional precision and recall measures penalize 'near misses' when they should be treated much the same as complete matches. The crossing-bracket measure (Carbone et al., 2004) is more forgiving, but still over-penalizes near misses and favors sparse bracketing. A 'word error' measure P_k proposed by Beeferman et al. (1999) compensates for the variation in boundary locations. It considers a moving window of width k equal to half the average

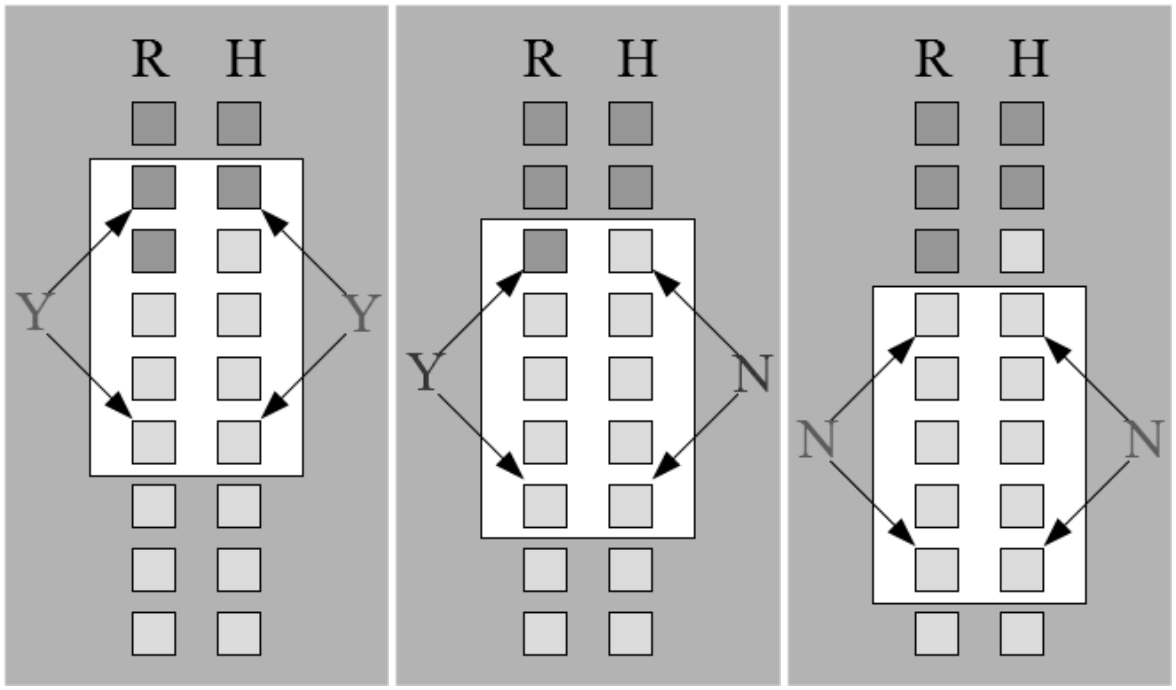


Figure 2.1. Moving window of the word error rate measures. At each step, the reference and hypothesized segmentations are compared regarding segment boundaries between the first and last words in the window. In the first image above, they agree that there is a boundary. In the second, they disagree, and in the third they agree that there is no boundary.

segment length in the reference segmentation. The error is the average disagreement, between the reference segmentation and the hypothesized segmentation, about whether the two ends of the window are in the same segment. Formally,

$$P_k = \frac{1}{N-k} \sum_{i=1}^{N-k} [1 - \delta(\delta(r_i, r_{i+k}), \delta(h_i, h_{i+k}))]$$

where N is the total number of words in the document, and k is the window width. The arguments r_i and h_i are the indices of the segments that contain word i in the reference and hypothesized segmentations, respectively, and δ is the Kronecker delta function, evaluating to 1 if its arguments are equal and to 0 otherwise.

2.1.3 Problems with the Beeferman error measure

However, several problems with the Beeferman error measure have been identified. Pevzner and Hearst (2001) pointed out that the Beeferman's P_k is more sensitive to false negatives than false positives. As a solution, they proposed WindowDiff, a modification of P_k that indicates the average disagreement about how many boundaries lie within the window, replacing the inner δ functions with the count of segment boundaries between the two words.

$$WD = \frac{1}{N-k} \sum_{i=1}^{N-k} [1 - \delta((r_{i+k} - r_i), (h_{i+k} - h_i))]$$

It is as sensitive to false positives as it is to false negatives.

A second problem is that P_k favors sparse bracketing, and WD is actually worse in this respect. For a typical reference segmentation with many more non-boundaries than boundaries, the NONE baseline, which hypothesizes no boundaries, obtains a P_k error of about 45%, while the ALL baseline, which hypothesizes that all potential boundaries are discourse boundaries, obtains a P_k error of about 55%. WindowDiff assigns the same error to the NONE baseline, while giving the ALL baseline an error close to 100%.

A third problem is that both P_k and WD undercount boundaries near the beginning and end of the text. The summation in the calculation of the error conventionally runs from $i = 0$ to $i = N - k$, that is, the counting begins with the beginning of the window at the beginning of the text, and ends when the end of the window gets to the end of the text. This means that a boundary that is $j < k$ units from the beginning or end of the text has a weight j/k compared to boundaries in the middle of the text.

Finally, neither of these measures are able to deal with the hierarchical structure of discourse.

2.1.4 A sketch of a solution

The proposed solution to the evaluation problem hinges on understanding linear segmentation as a special case of hierarchical segmentation. That is, a linear segmentation represents a collection of‘ some of the most prominent boundaries in a discourse, without any indication of the relative prominences of those boundaries. This provides a path to inducing a hierarchical segmentation from conflicting linear segmentation annotations and a fairly

straightforward method for generalizing the linear segmentation error measures to hierarchical cases.

CHAPTER 3

The Reference Segmentation

Seven highly monologic lectures were selected from the Michigan Corpus of Academic Spoken English (MiCASE) (Simpson, Briggs, Ovens, & Swales, 2000), and following the design of Passonneau and Litman (1997), an average of six naive readers per discourse sample provided linear segmentations. The linear segmentations were combined into hierarchical segmentations according to the statistical significance of annotator agreement.

3.1 ANNOTATION

Word-per-turn ratios were calculated for all lectures in MiCASE, and the seven lectures with the highest word-per-turn ratios were selected. Each lecture was split into two or three sections of approximately 4000 words, to make the annotators' tasks more bearable and more comparable in size. All pauses, marked in MiCASE by punctuation, were treated as delimiting prosodic phrases. The number of splits per text was determined by the word counts and the texts were split into sections by dividing the number of prosodic phrases. Via an internet web form interface, native speakers of English, recruited from introductory linguistics courses, were asked to identify points where 'topic changes' occur. Potential boundary locations were restricted to prosodic boundaries. One page of the annotation interface is shown if Figure 3.1. (See Appendix 1 for the complete process.) The annotators were permitted to mark as many or as few boundaries as they saw fit. Because sample texts were chosen at random for the annotation task, and it was possible to reload the page with a new text, some of the texts (likely some of the more difficult ones) had too few annotators to achieve statistical significance of annotator agreement. Four texts with fewer than five annotators were not included in the evaluation.

3.2 DIFFERENCES IN BOUNDARY LOCATION

Passonneau and Litman (1996) note that their methodology misses some boundaries where, for example, two annotators marked one prosodic boundary and three others marked the next prosodic boundary, at an episodic transition in the narrative. Under the exact agreement criterion, no boundary is assigned, even though 5 readers agree there is a discourse boundary somewhere close. At a lower standard of statistical significance (Passonneau & Litman, 1997), a boundary is assigned if 3 or more readers mark a boundary. In this case it

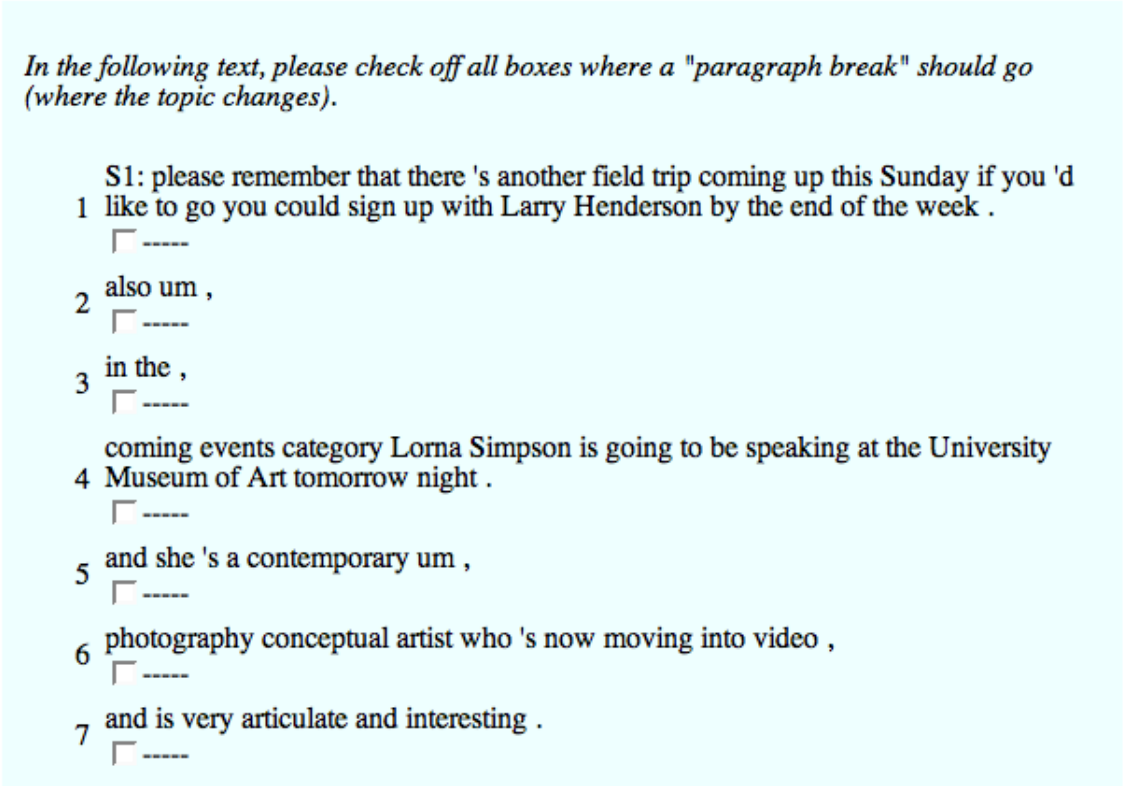


Figure 3.1. One page of the annotation interface

would be quite possible for 3 or 4 to mark one prosodic boundary and 3 others to mark the next point, in which case two boundaries would be assigned where there should really be only one.

We first consider a sliding window a few phrases wide. When the cumulative weights of the boundaries in the window reach a certain threshold, we could count that as agreement. A boundary in the reference segmentation would be placed at the middle of the window, and the contributing boundary judgments would be removed from further consideration. The problem with this is that it favors boundaries early in the discourse, and joins judgments that might not belong together. For example, consider the discourse sample in Figure 3.2, where the small boxes represent the hypothetical boundary marks of 6 different individuals. On the first pass through the text, looking for boundaries with 6 assenting annotators, no 3-boundary window reaches the required count. On the second pass, looking for 5 boundary judgments, the window spanning *a* to *d* is considered first, and a discourse boundary is assigned between *b* and *c*, even though no annotator marked that point. Furthermore, if we had passed the window through the text in the other direction, the boundary would have been placed between *d* and *e*, and the best position for a boundary is actually between *c* and *d*.

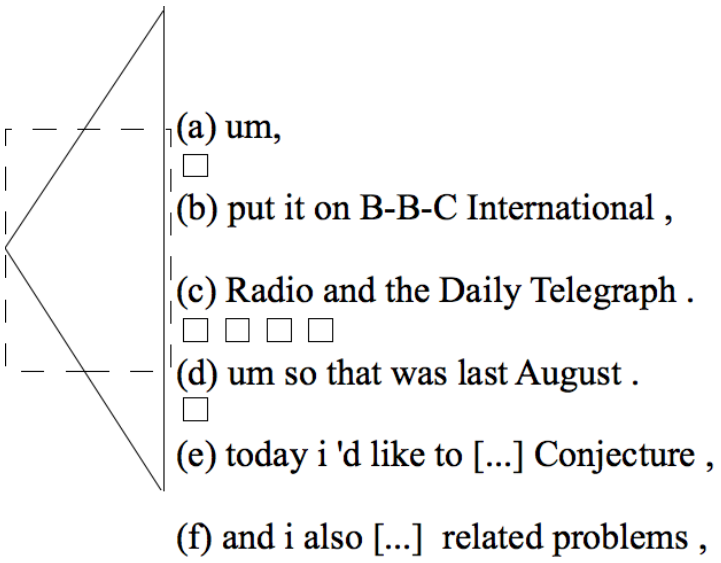


Figure 3.2. Illustration of rectangular and triangular annotation weighting windows. The squares represent the positions where 6 hypothetical annotators indicated a boundary.

These problems can be relieved somewhat by down-weighting boundaries further from the center of the window. A gaussian or triangular window could be used rather than a rectangular window. When a down-weighted boundary contributed to a reference boundary, that portion of its weight is removed from further consideration, not the whole boundary. Using a triangular window, with the central point receiving full weight, the points next to it receiving $\frac{2}{3}$ weight, and the points next further out receiving $\frac{1}{3}$ weight, the discourse in the previous paragraph would obtain maximum count of 5 when the window is centered between c and d, assigning the boundary in the correct position. In this study, I use a gaussian window with $\sigma = 1.5$. If an annotator has more than one boundary mark within the window, the mark closest to the center of the window is used, or if they are equidistant, the earlier one is used, since later marks may belong to boundaries that have not yet passed through the window, whereas the earlier boundaries have already been checked. Residual weights that fall below a threshold are also removed, to limit selection interference with larger remaining weights.

3.3 Differences in segment count

Passonneau and Litman were creating linear reference segmentations, but here we are interested in creating tree reference segmentations. Readers that mark fewer boundaries are likely marking the more major transitions, representing branchings higher in the tree. A boundary identified by 3 conservative annotators should be marked more highly prominent than one identified by 3 more liberal annotators. In this study, the annotators' liberality is

taken into consideration by giving each annotator an equal total weight, so that the most conservative annotator, who might have marked 20 boundaries, would have 10000 'votes' distributed evenly over those 20 boundaries, and the most liberal annotator, who might have 100 boundaries marked, would have 10000 'votes' distributed evenly over those 100 boundaries. As a result, the first annotator's marks are weighted more heavily than the second's. The procedure for this weighted combination of annotations is made more explicit in Algorithm 3.1.

Algorithm 3.1 Combine annotations

$data \Leftarrow$ boundary numbers list for each annotator
/* * Weight annotations by how many boundaries that annotator marked * */
$VoteTotal \Leftarrow 10000$
$table \Leftarrow$ an empty list
for all $numberList \in data$ **do**
  $vote \Leftarrow VoteTotal/length(numberList)$
  append $dict(\{(num, vote)$ for $num \in numberList\})$ to $table$
end for
$votes \Leftarrow \{VoteTotal/length(numberList)$ for $numberList \in data\}$
$minSingleVote \Leftarrow median(votes)$
/* * Find and group high-vote regions by vote level * */
$gold \Leftarrow$ an empty list
$maxAgreement \Leftarrow sum(votes) + 1$
$agreementStepSize \Leftarrow (agreementStepFactor * minSingleVote + 1)$
$threshold \Leftarrow maxAgreement$
$sumMax \Leftarrow 0$
while $threshold > minSingleVote$ **do**
  for all $index \in range(maxLineNumber + 1)$ **do**
    $tempSum \Leftarrow$ sum votes in $table$ near $index$ using $windowFunction$
    if $tempSum \geq threshold$ **then**
      append $(threshold, index)$ to $gold$
      remove votes from $table$ near $index$ using $windowFunction$
    else if $tempSum > maxSum$ **then**
      $maxSum \Leftarrow tempSum$
    end if
    $threshold \Leftarrow maxSum - agreementStepSize$
    $maxSum \Leftarrow 0$
  end for
end while
print $gold$

3.4 STATISTICAL SIGNIFICANCE OF REFERENCE BOUNDARIES

Since the boundaries with greater consensus likely represent branchings higher up in a discourse tree, Iidentify boundaries in order of the statistical significance of agreement within the Gaussian window, and rank them accordingly. To help ensure that all gold standard boundaries are statistically significant, the distributions of annotator agreement are calculated empirically from randomly generated segmentations. For each discourse sample, the random segmentations are produced by using the same number of annotators with the same ratios of boundaries to non-boundaries, but assuming that each annotator's marked boundaries are a random sample of the possible boundaries. A boundary had to have a summed weight at least as great as the median single-annotator weight to be included in the distribution. A boundary in the reference segmentation was then deemed statistically significant if its summed weight was higher than 95% of the summed weights in the random distribution. The pseudocode is provided as Algorithm 3.2.

Algorithm 3.2 Find and filter boundary numbers of reference segmentation

```
data ⇐ boundary numbers list for each annotator
counts ⇐ an empty dictionary
for all index ∈ range(10) do /*Create random ensemble of segmentations*/
    lineNums ⇐ sequence of all possible line numbers
    randNums ⇐ {sample(lineNums, length(numList)) for numList ∈ data}
    randSegNums ⇐ combine randNums (Alg. 3.1)
    for all weight, lineNum ∈ randSegNums do
        increment counts[weight]
    end for
end for
limit ⇐ .05 * sum(counts.values())
votes ⇐ counts.keys() sorted by descending weight
sum {counts[weight] for weight ∈ votes} until reach limit
minSignificantVotes ⇐ last used weight
goldSegNums ⇐ combine data (Alg. 3.1)
remove element from goldSegNums if weight is less than minSignificantVotes
print goldSegNums
```

3.5 RESULTS

Features of the discourse samples are profiled in Table 3.1. The annotators varied greatly on the number of segments they indicated in the texts, ranging from 6 to 203, with document averages ranging from 22 to 61. Of the 17 original text samples, 4 had less than 5

annotators, so just 13 text samples are included in the gold standard. The resulting gold standard has considerably fewer segments than what the average human annotator found. The segment counts in the gold standard are comparable to that of the most conservative annotators. The number of segments found ranged from 5 to 17, with an average of 10.

				Segment Counts		
Files	Words	Phrases	Readers	Range	Mean	Gold
col385mu054a	4261	450	5	19–73	39	12
col385mu054b	4095	451	2	38–84	61	
lel175su106a	4212	475	5	17–79	44	11
lel175su106b	4153	475	6	21–76	39	14
lel175su106c	4356	478	7	9–35	22	8
lel200ju105a	3618	450	6	24–72	39	17
lel200ju105b	4061	448	7	15–75	31	15
lel215su150a	4714	530	5	9–180	53	6
lel215su150b	5025	530	7	10–65	29	9
lel215su150c	4687	534	7	16–203	56	7
lel300su020a	3853	495	5	24–54	37	13
lel300su020b	3635	495	2	44–45	45	
lel300su020c	4047	488	3	13–57	30	
lel320ju143a	4233	327	7	10–71	32	9
lel320ju143b	4684	327	5	22–49	31	9
lel485ju097a	4579	582	4	21–47	35	
lel485ju097b	4157	582	5	6–93	53	5

Table 3.1. Profiles of the sample texts. The readers marked a high number of segments, on average, and varied greatly, while the resulting gold standard had many fewer segments.

CHAPTER 4

The Error Measure

4.1 A HIERARCHICAL MEASURE

The proposed hierarchical error measure is the mean of Beeferman metric scores calculated over a series of linear segmentation cuts of the segmentation tree. In the first step, only the boundaries of the highest prominence are used, and in each following step, one more level of boundaries are included

Assume a set R of reference (gold standard) boundaries and a set H of hypothesized boundaries each in rank order (prominent boundaries precede less prominent ones). For the moment, assume that no two boundaries have the same rank. The hierarchical atom error rate, based on P_k, is calculated as

$$Hier_{P_k} = \frac{1}{|R|} \sum_{i=1}^{|R|} P_k(R_i, H_i)$$

where

$$R_i = \{b_j | b_j \in R \wedge j \leq i\}$$

$$H_i = \{b_j | b_j \in H \wedge j \leq i\}.$$

Since all i boundaries have approximately equal influence in each $P_k(R_i, H_i)$ term, the influence of boundary b_j on the total $Hier_{P_k}$ can be calculated as

$$\frac{1}{|R|} \sum_{i=j}^{|R|} \frac{1}{i}.$$

In the limit of large $|R|$ (i.e. the reference segmentation is richly informative), the sum reduces to an integral, and we have

$$\int_j^{|R|} \frac{1}{i}\, di = [\ln i]_j^{|R|} = \ln |R| - \ln j.$$

In this logarithmic limit, the error measure more clearly has behavior matching intuitions about the boundaries' relative importance:

- The most important boundaries have many times more influence than the least important
- Differences in ranking at high ranks have greater impact than differences in ranking at low ranks

- Adding low ranked boundaries to the set or removing some from the set has little consequence, since their individual influence on the error is small.

A similar hierarchical atom error rate based on WindowDiff ($Hier_{WD}$) can be calculated in a parallel manner. To distinguish between the two variants of the hierarchical error measure, I will refer to $Hier_{P_k}$ as the hierarchical Beeferman measure, and $Hier_{WD}$ as the hierarchical WindowDiff measure.

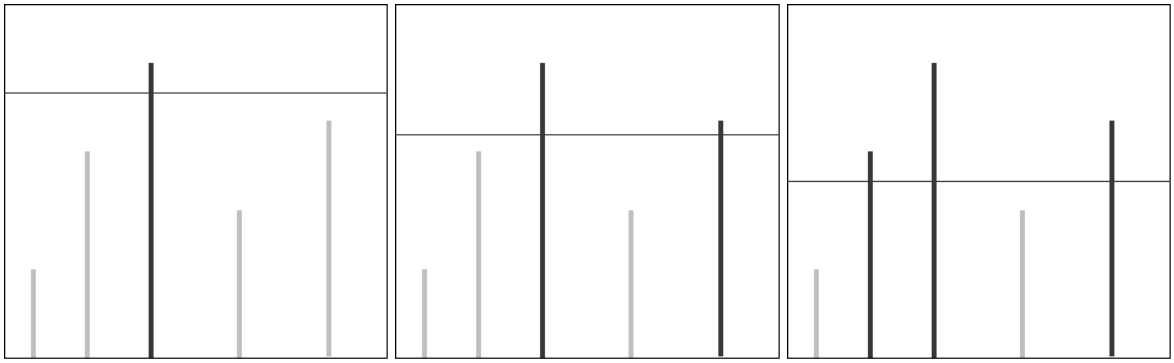


Figure 4.1. Sequential linearizations in computing hierarchical word error rate. In the first step, only the highest boundary is used, producing just two segments. Each following step includes one more boundary.

The above definition assumes a unique rank order, with no two boundaries having the same prominence, in either the reference or hypothesized segmentations. In the case of reference boundaries sharing ranks, one P_k term is calculated for each rank level in the reference segmentation, and weighted according to how many boundaries were at that level. In the degenerate case of linear segmentation, all segments have the same rank, and this average reduces to the original P_k. In the case of hypothesized boundaries sharing ranks, each affected term in the summation should be the average over all combinations (n boundaries at the next rank *Choose r* boundaries to complete H_i). When the number of combinations is large, the computational complexity of the calculation can be reduced without sacrificing much accuracy by using a representative sampling of the combinations, as this closely approximates the average.

With these complications, we redefine the hierarchical atom error rate as

$$Hier_{P_k} = \frac{1}{|R|} \sum_{i=1}^{|R|} \frac{c_i}{\binom{n}{|R_i|-m}} \sum_h P_k(R_i, H_{ih})$$

where c_i is the number of boundaries at level i in R. R_i remains the same as before, but H_{ih} denotes the sets of boundaries that satisfy the criteria that $|H_{ih}| = |R_i|$ and no boundary in $H \setminus H_{ih}$ is of higher prominence than any boundary in H_{ih}. The number of combinations that fulfill these criteria are determined by n, the number of boundaries in H at the critical prominence level, and m, the number of boundaries prior to that prominence level.

We make two further adjustments to the calculation of P_k or WD. First, when the set of hypothesized boundaries is smaller than the set of reference boundaries, it is not simple to keep H_{ih} the same size as R_i. We could permit H_{ih} to be smaller than R_i when we run out of boundaries in H, but that unnecessarily penalizes the hypothesized segmentation. The set of possible boundaries (word or sentence boundaries) which were not marked as segment boundaries can be understood to be segment boundaries of a baseline low ranking. Adding these unmarked boundaries to H prevents incurring an undeserved penalty for false negatives. Second, to address the problem of undercounting boundaries near the beginning and end of the text, we allow the window to wrap around the end of the text. Once the window reaches the end of the text, the window of width k stretches from atom i to $j = (i+k) \pmod N$, redefining P_k:

$$P_k = \frac{1}{N}\sum_{i=1}^{N}[1 - \delta(\delta(r_i, r_j), \delta(h_i, h_j))]$$

4.2 Replication of Choi et al.

As a preliminary test of the error measure, I evaluated the algorithms on the standard segmentation data set that Choi (2000) compiled. Each file in that data is composed of 10 random portions of texts from the Brown Corpus. The following results are based on the T_{3-11} set, in which text segment lengths are uniformly distributed between 3 and 11 sentences. Since each file is composed of a sequence of text portions, the reference segmentation is linear, not hierarchical. Nevertheless, I here evaluate hierarchical segmentation algorithms with the hierarchical measure, to demonstrate the relationship between the linear and hierarchical measures and between the linear and hierarchical segmentation algorithms.

4.2.1 Segmentation algorithms

The C99 (Choi, 2000) and CWM (Choi et al., 2001) algorithms were evaluated. While these were designed and originally evaluated as linear segmentation algorithms, the hierarchical clustering they use makes hierarchical segmentation a trivial matter of retaining the order of the cluster splits. I refer to the hierarchical versions of these algorithms as HC99 and HCWM. The HC99 implementation used here is built directly from the C99 code which Choi released for educational use, and the HCWM implementation is based off that. The

implementation uses a document-based LSI space built with Infomap-NLP[1] from the British National Corpus (Aston & Burnard, 1998), whereas the original CWM used sentence-based and paragraph-based LSI spaces derived from the Brown Corpus. Because of these differences, the implementation of HCWM reported here differs somewhat from the implementation of CWM reported by Choi et al. (2001).

The C99 and CWM algorithms include a criterion for optional automatic determination of the number of segments, but the hierarchical error measure does not penalize a segmentation for having more segments (defined by lower ranking boundaries) than the reference segmentation, since additional lower ranked boundaries are ignored. With the hierarchical error measure, there is no advantage in this aspect of the algorithm, and for this reason, I just used a constant number of segments, equal to the reference segmentation, for the results reported here.

One baseline (BIN) was constructed by a recursive bisection of segments, and another baseline (NONE) consisted of only the implicit boundaries at the beginning and end of the discourse, and all the possible intermediate boundaries (sentence breaks) are implicitly at one unmarked lower rank.

4.2.2 Results

The calculated $Hier_{P_k}$ error rates are displayed in Fig. 4.2.[2] The error for HC99 in Fig. 4.2a (12.5%) matches what Choi et al. (2001) reported (12%), while the error for HCWM (12.1%) is higher than that reported for the version with a paragraph-based 500-dimension LSI space (9%) but appears comparable to their sentence-based 400-dimension LSI space. (They do not report results for the sentence-based spaces on this T_{3-11} data set, but based on the results they report for a larger data set, it would appear to be about 12% for the T_{3-11} set.) The result for BIN (43.9%) is slightly lower than what Choi et al. (2001) reported for their equal-size segment baseline (45%). Since BIN would be an equal-segment baseline if there were only 8 segments per text, BIN should be similar to Choi et al's equal-size baseline. And the result for NONE (46.1%) agrees with Choi et al. (2001)'s results for their NONE (46%) baseline.

Comparison of graphs (a) and (b) in Fig. 4.2 shows that continuing the sum to wrap the window around to the beginning of the text generally lowers the measured error, to the greatest extent for BIN and least for HCWM. The average segment length in the reference segmentation is 7 sentences, so the window size k is usually 3 or 4 sentences, comparable to

[1] Software available at http://infomap-nlp.sourceforge.net

[2] The error rates in this section are calculated using the word-error rate for comparison with Choi's results, but since the candidate boundaries are actually the line breaks, the line-error rate would be more appropriate. Line error rates are 1% to 2% higher.

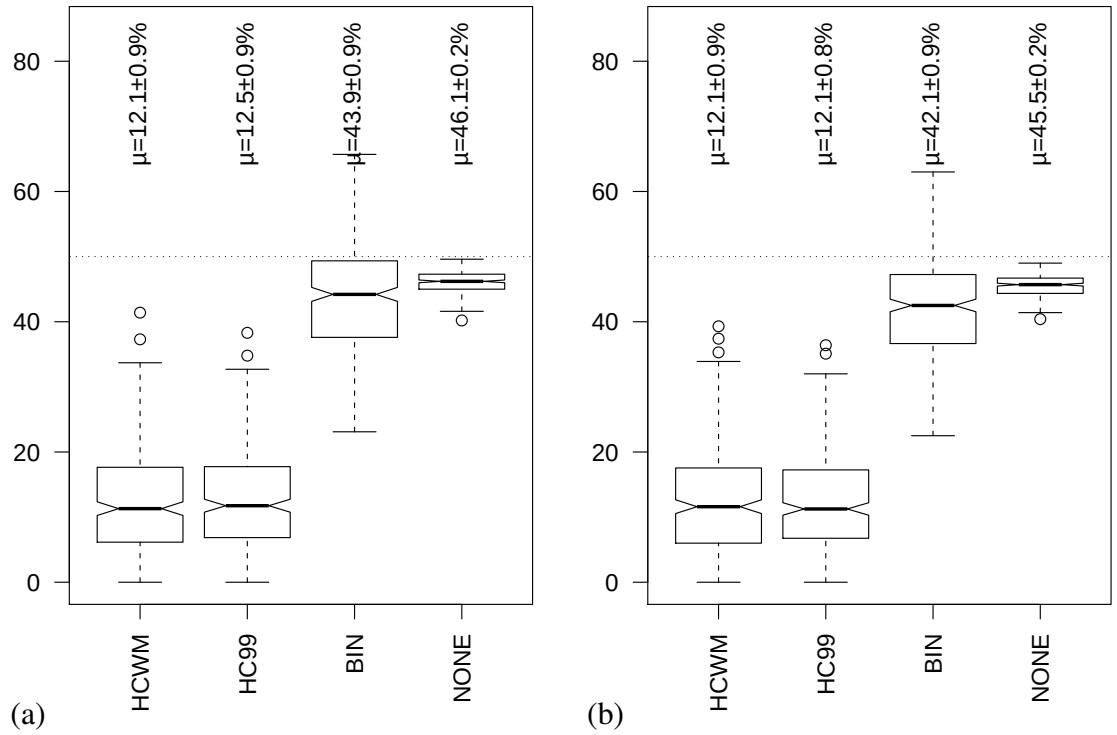


Figure 4.2. Distributions of $Hier_{P_k}$ for each of the hypothesized and baseline segmentation algorithms. The data in graph (a) is calculated with sums that stop at $N-k$ (when the window reaches the end of the text), whereas (b) is calculated with sums that run to N (wrapping the window back to the beginning). The boxes indicate the quartiles, and the means with 95% confidence intervals are written above.

the minimum segment length (3). As a result, a boundary very rarely falls within *k* sentences of the text ends, and fully including these sentences in the sum leads to a lower error for segmentations like BIN that don't hypothesize boundaries near the text ends.

The $Hier_{WD}$ hierarchical error rates (calculated according to WindowDiff) are consistently higher (Fig. 4.2c, d) than the corresponding $Hier_{P_k}$. WindowDiff scores are never lower than P_k scores, because in order to count as in agreement, the two segmentations must agree about the number of boundaries within the window rather than just about whether there are boundaries within the window. But these scores are not much higher than $Hier_{P_k}$ either, even though the original linear WindowDiff measure sometimes assigns much higher scores. Under the original WindowDiff measure, with reference and hypothesized boundary sets of unequal size, the NONE baseline scores 43.8% (cf. P_k=43.5% for sum to *N*), while an ALL baseline scores 99.2% (cf. P_k=51.1% for sum to *N*). WindowDiff was designed to penalize false positives even when two boundaries are close together, a condition that P_k underpenalizes. When a hypothesized segmentation has more segments than the reference

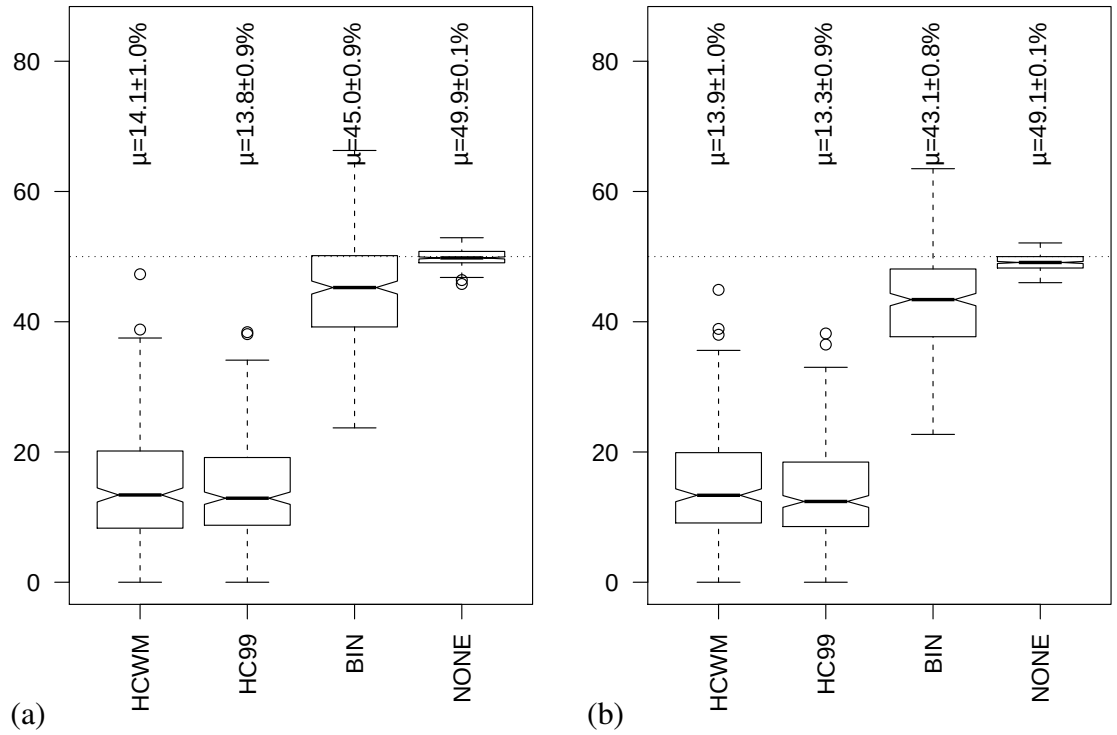


Figure 4.3. Distributions of $Hier_{WD}$ for each of the hypothesized and baseline segmentation algorithms. The data in graph (a) is calculated with sums that stop at $N-k$ (when the window reaches the end of the text), whereas (b) is calculated with sums that run to N (wrapping the window back to the beginning). The boxes indicate the quartiles, and the means with 95% confidence intervals are written above.

segmentation, the extra boundaries incur false positive penalties without corresponding false negative penalties, and WindowDiff assigns an error rate that is higher than the P_k error rate and sometimes even higher than the NONE baseline. But with the hierarchical $Hier_{WD}$ error, extra boundaries are sampled or ignored, and so every false positive has a corresponding false negative, which limits the divergence between $Hier_{WD}$ and $Hier_{P_k}$ and keeps the $Hier_{WD}$ of informed segmentations below baseline errors. As with $Hier_{P_k}$, wrapping $Hier_{WD}$ around the end (Fig. 4.2d), has only a slight effect on the error, but the effect is most pronounced on BIN, reflecting the fact that BIN, like the reference segmentation, systematically does not place boundaries near the text ends.

4.3 SUMMARY

We have seen here that treating linear segmentations as a special case of hierarchical segmentations, having just one rank of marked boundaries but having implicit higher ranking boundaries at the text ends and implicit lower ranking boundaries at all ‘non-boundaries’,

resolves the outstanding issues of unequal sensitivity that P_k and WindowDiff have. Furthermore, in sampling hypothesized boundaries to match the number of reference boundaries, the hierarchical conception of the error metric smoothly adapts to segmentations that overestimate or underestimate the number of segments. A segmentation can not do much worse than 50% (at chance) just by hypothesizing fewer or more segments than the reference segmentation 'knows' about. The major remaining strength of WindowDiff over the P_k metric is that P_k still undercounts errors when there are segments much smaller than the average size.

CHAPTER 5

Evaluation

5.1 Overview

In this chapter, I compare the two algorithms HC99 and HCWM against three kinds of baselines. I use the reference segmentation described in Chapter 3 and the hierarchical error measures developed in the previous chapter. I also evaluate the segmentations while ignoring the reference segmentation's boundary prominence information, to factor out the performance on segment position versus the performance on segment ranking.

5.2 Algorithms

The two hierarchical algorithms HC99 and HCWM were used to produce hypothesized segmentations for the MiCASE lecture data. In Choi's reference data, built from written data from the Brown Corpus (Francis & Kucera, 1979), sentences are distinct entities and they are similar in length. In contrast, the MiCASE data is composed of turns and 'prosodic phrases', but these both vary widely in length. In order to maintain an approximately constant quantity of semantic information on each line, the texts were word-wrapped at a specific number of characters. This introduces a free parameter that we will explore in the next section.

For comparison with the hierarchical segmentation algorithms, the BIN and NONE algorithms were applied to the lecture data. In addition, a baseline random segmentation (RAND) was constructed for each discourse sample in the same manner as the segmentations used to construct the random distribution of agreement scores (Alg. 3.2).

5.3 The line-length parameter

The discourse segmentations are composed by the algorithms as groupings of lines. If the line lengths are too short, the semantic representation of sequential lines varies wildly. If a line is composed entirely of stop words and rare words, which were not included in the LSI space, there may even be no semantic content represented for a line. This adds noise to the similarity measurements. On the other hand, if line lengths are too long, the algorithm may be prevented from choosing a boundary near the ideal point, or if the line lengths in the input to the algorithm are comparable to the reference segmentation's average segment length, the algorithm is forced to use the equal-size segmentation composed of those lines, and the only thing left for the algorithm to do is calculate a ranking for those segment boundaries.

In order to determine the best line length, hypothesized segmentations were produced that had line lengths from the exponential series $\{2^{\frac{n}{2}} : 10 < n < 20\}$. These were evaluated with and without the reference segmentation's boundary prominences, and quadratic regression lines were fitted to the sequence of mean error rates. The error rate for HCWM (Figure 5.2) depends heavily on line length, while HC99 (Figure 5.1) is less influenced by it. The minimum of the two HCWM regression curves and of the hierarchical error for HC99 are reached between 150 and 270. For the following evaluation, the line length is set to 200, because it lies in the middle of this range and is not too near any of the particular values just used to calculate the regression line.

5.4 RESULTS

The WindowDiff error rates for each of the hypothesized segmentations are presented in Figure 5.3. The standard error is larger than for the Choi data in the previous chapter, because that data has 300 texts whereas this data has just 13. According to one-sample two-sided t-tests on the evaluation that ignores reference boundary prominences, HC99 ($p < .05$) and HCWM ($p < .01$) are statistically significantly better than theoretical chance (50%), and according to paired t-tests, better than the baselines BIN ($p < .05$), NONE ($p < .01$) and RAND ($p < .01$). Oddly, NONE is also measured as worse than chance ($p < .01$). The other differences do not reach statistical significance. Among the fully hierarchical error rates, HCWM ($p < .01$) and NONE ($p < .05$) score below theoretical chance, and HCWM is statistically significantly better than HC99 and BIN at the $p < .05$ level, and better than NONE and RAND at the $p < .01$ level. The other differences are not statistically significant. The difference between including and ignoring reference boundary prominences is statistically significantly different for NONE ($p < .01$) according to a paired t-test, but the differences are non-significant for all of the other segmentations.

5.5 ANALYSIS

The 95% confidence intervals on the means in Figure 5.3 (corresponding to the standard error lines in Figures 5.1 & 5.2) is somewhat larger than suggested by the actual spread of means around the regression lines, but still the measurement error is noticeable for HC99, HCWM and BIN. For example, the scores shown for HC99 and HCWM in Figure 5.3a are 2% lower than predicted by the regression lines in Figures 5.1a & 5.2a. The boxplots show that the range of scores produced by HC99, HCWM and BIN are much larger than those produced by NONE or RAND. This is because HC99, HCWM and BIN make informed hypotheses about the distribution of segments, whereas NONE and RAND assume, in different ways, that potential segment boundaries are all about equally likely to be actual boundaries.

The mean scores for the BIN baseline are very close to 50% on the lecture data. In contrast, the mean score for BIN on the Choi standard data (Figure 4.3) was 45% for the linear measure and 43% for the hierarchical measure, just below the lower confidence interval for BIN's mean on this lecture data. Why did BIN do so poorly here when it performed well above chance on the Choi data? The difference is likely in the distributions of segment lengths. As seen in Figure 5.4, the Choi data segment lengths are well-defined by their mean, even when counted by word, because they were constructed with uniform distributions of segment length (counted by sentence). On the other hand, the distribution of segment lengths in the lecture data is more skewed, with many quite short segments and a few quite long segments. Some of the variance is due to the fact that the lecture texts had different numbers of segments (Table 3.1) and more variable line lengths as presented to the annotators. But the distribution is strongly skewed even when comparing segment lengths to the text average, and skewed distributions are found in individual texts. The text that has the longest segment (2009 words, almost four times as long as the average segment for that text) also has segments that are 20 and 32 words long. The text that has the next longest segment has segments that are 50 and 61 words long, and the text that has the smallest segment (5 words) also has a segment that is 1690 words, one third of the whole text and three times the average for that text.

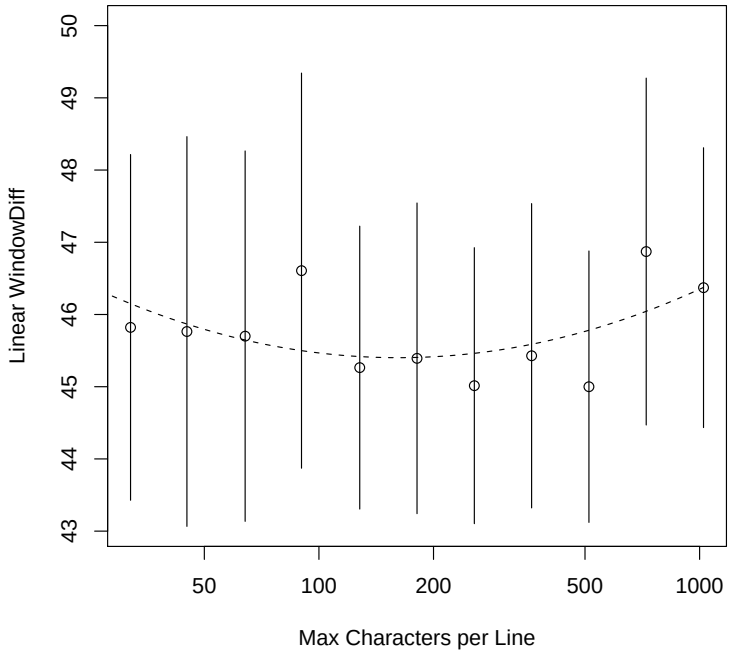
Strangely, NONE performed better when evaluated against the fully hierarchical reference segmentation than when evaluated with the reference boundary prominences ignored, and in both cases NONE performed significantly different than theoretical chance (50%). This is apparently a weakness in the measure or in how the measure was calculated here, since no algorithm should have an average performance worse than chance, and since NONE does not have any information corresponding to the actual distribution of segments, it should not perform better than chance either.

The error rates for both HC99 and HCWM are much higher on the lecture data than they are on the Choi data. Choi's evaluation corpus was specifically designed to have obvious boundaries, whereas the boundaries in these discourse samples are much less dramatic. As discussed by Kauchak and Chen (2005), algorithms developed for news and expository text do not perform well on narrative text, and the same applies to expository speech. It may be that expository speech is less tree-like than expository text, less abrupt in its transitions, or less revealed by the lexical cohesion that these two algorithms exploit.

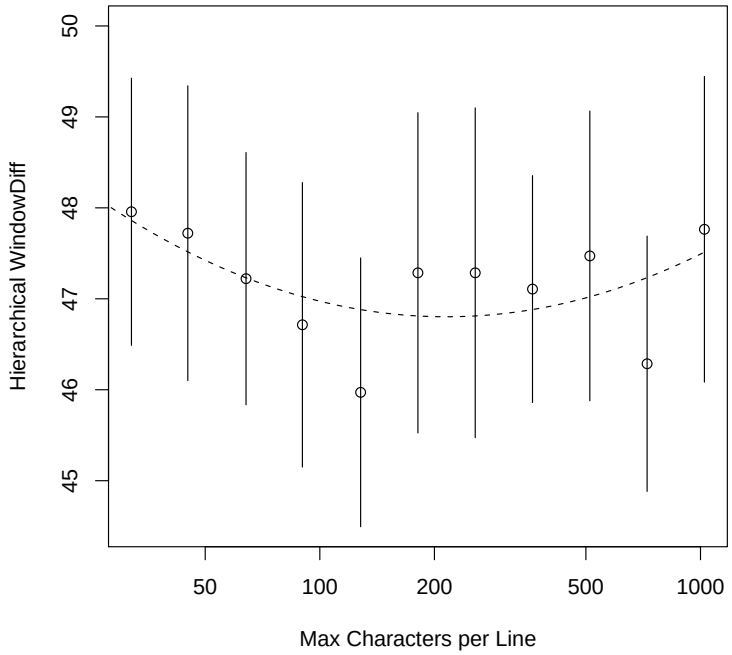
But it may also be that the high error rates in the lecture data reflect problems with the reference segmentation. To evaluate this hypothesis, I constructed an outline for one of the lectures, lel200ju105, and converted the outline into a segmentation. Comparable whole-lecture segmentations were also constructed from the multi-annotator reference segmentation (GOLD) and the RAND baseline by concatenating the evaluations texts

lel200ju105a and lel200ju105b, which were originally created by partitioning that lecture. The segment boundaries at the joint of the concatenation were removed, but the segmentations were otherwise unchanged. Segmentations by HCWM, HC99, BIN and NONE were produced by running the algorithms on the whole lecture.

The resulting error rates are displayed in Figure 5.5. All the automatic algorithms score near 50% while GOLD is much lower, below 40%. A single text is not a reliable indicator of general trends, but since HC99 and HCWM did not perform above baseline and GOLD did on this text, it suggests that the method used to construct the GOLD reference segmentation is comparable to what a careful expert would produce, and that the task is much more difficult than segmentation of the Choi data. This does not remove the possibility that the reference segmentation might be somewhat unreliable, but it suggests that the primary cause of the high error rates is the difficulty of the task itself, and not problems in the reference segmentation.

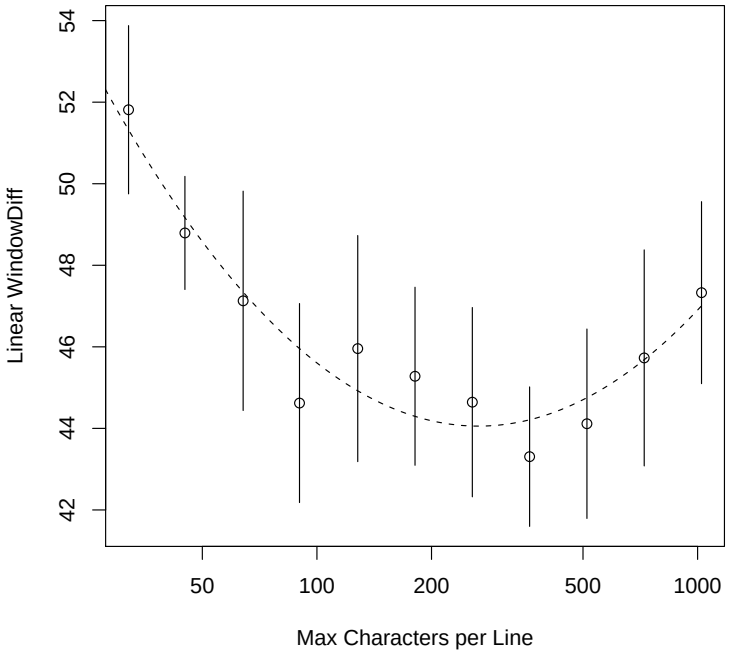


(a)

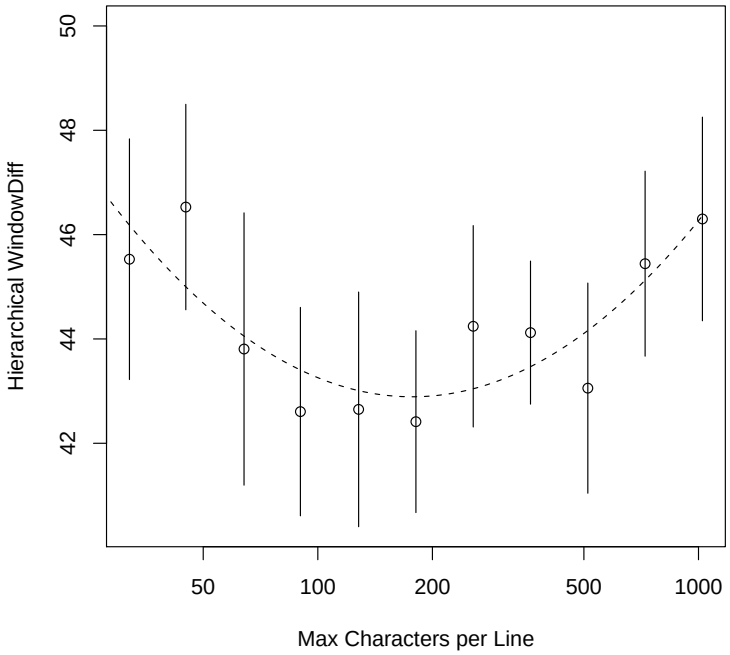


(b)

Figure 5.1. Hierarchical WindowDiff error rates for HC99 varying by line length, evaluated with (a) the reference boundary prominences ignored and (b) the reference boundary prominences included. The vertical lines represent the standard error on the mean.



(a)



(b)

Figure 5.2. Hierarchical WindowDiff error rates for HCWM, varying by line length, evaluated with (a) the reference boundary prominences ignored and (b) the reference boundary prominences included. The vertical lines represent the standard error on the mean.

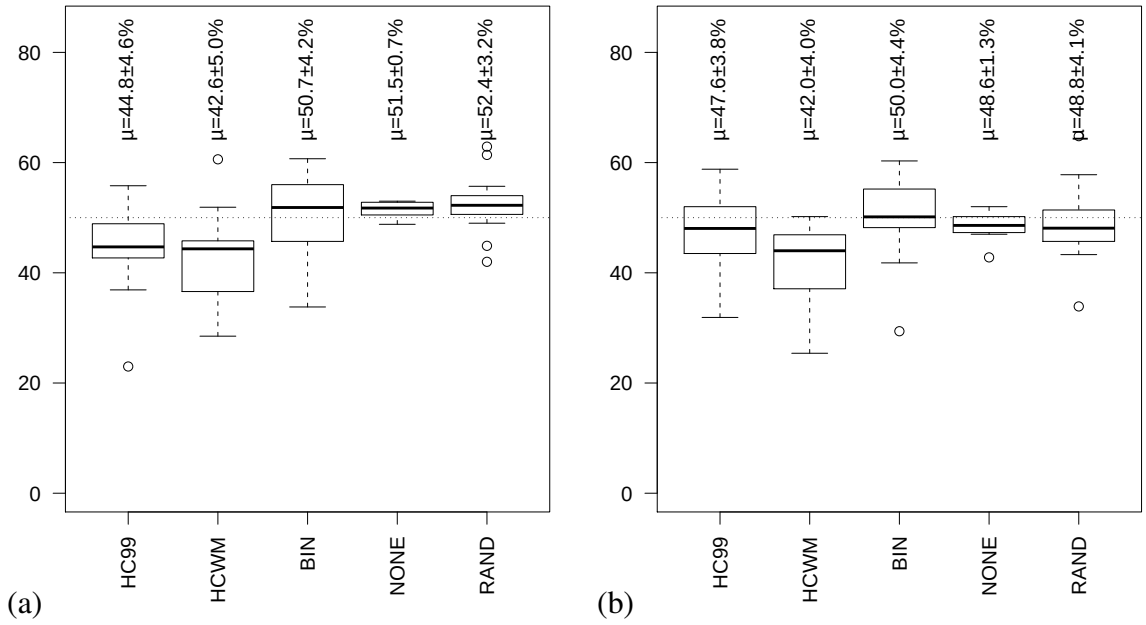


Figure 5.3. Hierarchical WindowDiff error rates for each of the segmentation algorithms. (a) Ignoring reference boundary prominences (b) Including reference boundary prominences

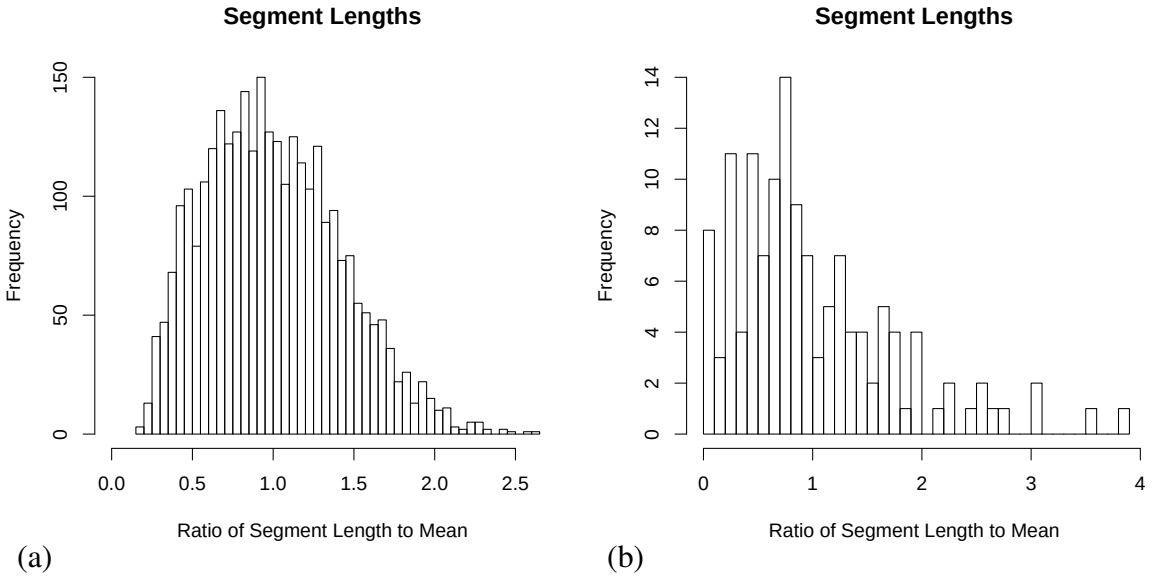
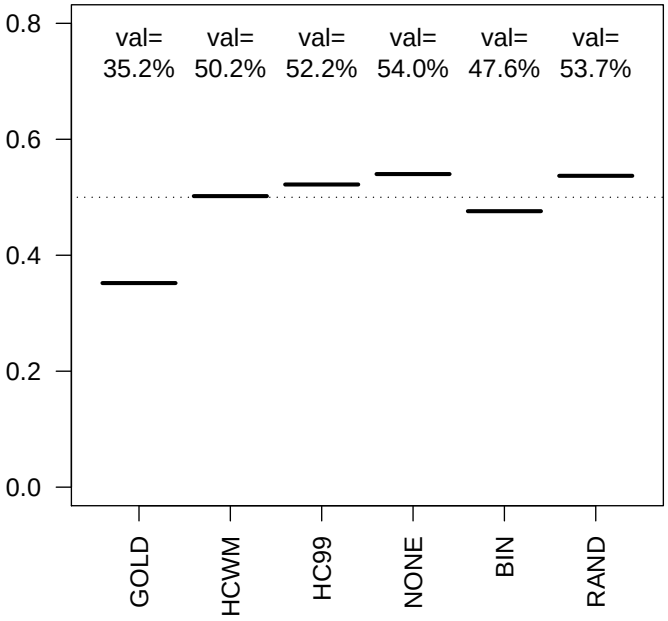
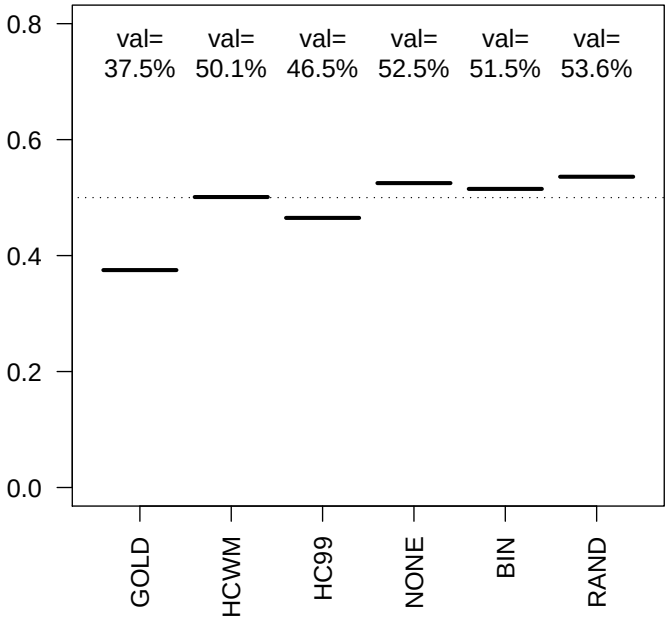


Figure 5.4. Distribution of words per segment for (a) Choi standard data (b) Lecture data



(a)



(b)

Figure 5.5. Algorithms evaluated against expert outline of one lecture, by (a) linear WindowDiff (b) hierarchical WindowDiff

CHAPTER 6

Conclusion

6.1 SUMMARY

The experiments presented in the previous chapters show that a rigorous evaluation of hierarchical segmentation is achievable, even in discourse genres like lectures that may be difficult to reliably annotate for discourse structure.

In Chapter 3, I described a modification of the annotation method of Passonneau and Litman (1997), which permits us to construct a hierarchical segmentation based on the linear segment annotations of naive readers. In the last chapter, the resulting annotation was found to agree, to a limited extent, with automatic segmentations and the hierarchical segmentation I annotated for one lecture.

In Chapter 4, I introduced a modification of the error measure developed by Beeferman et al. (1999) and Pevzner and Hearst (2001). I then showed that this modification, directed at evaluating hierarchical segmentations, also produces a more robust evaluation of linear segmentations as well. And applied to hierarchical segmentations, it successfully distinguishes the lexically informed HCWM segmentation from baseline segmentations.

As part of this study, I make use of previously developed linear segmentation algorithms for the purpose of hierarchical segmentations. Since many of the state-of-the-art linear segmentation algorithms, like C99 and CWM, make use of hierarchical representations of the text, these algorithms can be adapted to performing hierarchical segmentation, now that we have a reliable evaluation measure.

Caveat whatever problems may remain in the reference segmentation, the high error rates on the lecture data indicate that expository speech is much more difficult to segment than the newswire data and expository text used in other studies. The algorithms do detect some of the structure, performing better than baseline but much worse than on newswire data and the artificial segmentation.

6.2 DISCUSSION

Annotation by untrained readers has the advantage of being theory-neutral, in the sense that annotators are not working with an explicit model of discourse structure. They may be pursuing slightly different goals and employing different criteria, but they are operating with an intuitive model. The annotations represent a sampling of the different ways untrained readers interpret the structure of the discourse and the segmentation task.

However, in compiling the gold standard, I have made two assumptions. The first is that the discourse structure is a tree, and not a DAG or something less restricted. While the graph structure may be more complex than a tree, much linguistic theory and conventions of outlining lectures suggest that a tree is a good approximation. Divergences from the tree approximation will become more apparent as segmentation algorithms become more sophisticated and we examine hierarchical segmentation in other registers.

The second assumption is that major branchings are more likely to have many annotators notice and mark them. However, it may be, since segment boundaries are fluid transitions, that the transitions between major discourse segments are more spread out than the transitions between minor ones, in which case the major transitions would actually be less likely to have many readers mark boundaries close together. If this is the case, the boundary prominence information in the resulting reference segmentation would be unreliable. The contrast between evaluation with and without the boundary prominences should, in principle, reveal if this is the case, but because the overall performance of the automatic segmentation algorithms is poor, and the statistical power of this small corpus is low, the evaluations presented here cannot address this issue.

6.3 FUTURE WORK

I have demonstrated a method for deriving a gold standard tree segmentation from the linear segmentation annotations of untrained readers, a more theory-neutral alternative to the segmentations produced by trained annotators. I furthermore suggest an evaluation metric for hierarchical segmentation that takes into consideration the segment boundaries' vagueness of location and differing importance. The procedure and metric show that two state-of-the-art algorithms developed for news data perform quite poorly on transcribed expository speech, but at least one of the two does perform better than baseline segmentations.

In complementary work (Carroll, 2010), I show that the evaluation measure applies equally well, if not better, to encyclopedia data, where explicit annotations of the discourse structure are available in the form of the headings and subheadings provided by the encyclopedia authors. State-of-the-art segmentation algorithms do not perform very well on the encyclopedia data either, though the performance is not far off from inter-annotator agreement on similar data.

There are several objectives to pursue in future research. The method used here for compiling the gold standard, involving untrained readers performing linear segmentations, should be compared directly to the method involving trained annotators, to evaluate the extent to which these annotation strategies differ in their concept of discourse structure. Furthermore, the hierarchical segmentation algorithms previously described in the literature should be evaluated in direct comparisons, on a variety of text types, and likewise compared

to the segmentations produced by individual trained annotators. Based on those evaluations, and related work in linear segmentation, better algorithms can then be developed, with machine learning and more linguistically-rich features.

References

Allan, J. (2002). *Topic detection and tracking: Event-based information organization*. Springer.

Angheluta, R., De Busser, R., & Moens, M.-F. (2002). The use of topic segmentation for automatic summarization. In *DUC 2002.* Proceedings of the Workshop on Multi-document Summarization Evaluation at the 40th ACL.

Asher, N., & Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

Aston, G., & Burnard, L. (1998). *The BNC handbook: Exploring the British National Corpus with SARA*. Edinburgh University Press.

Beeferman, D., Berger, A., & Lafferty, J. D. (1999). Statistical models for text segmentation. *Machine Learning*, *34*(1-3), 177-210.

Biber, D., Csomay, E., Jones, J. K., & Keck, C. (2004). A corpus linguistic investigation of vocabulary-based discourse units in university registers. In *Applied corpus linguistics: A multidimensional perspective* (p. 58-72). Amsterdam: Rodopi.

Boguraev, B., & Neff, M. S. (2000). Discourse segmentation in aid of document summarization. In *33rd HICSS.*

Carbone, M., Gal, Y., Shieber, S., & Grosz, B. (2004). Unifying annotated discourse hierarchies to create a gold standard. In *Proceedings of 4th SIGDIAL workshop on discourse and dialogue.*

Carroll, L. (2010). Evaluating hierarchical discourse segmentation. In *Proceedings of NAACL 2010* (pp. 993–1001). Association for Computational Linguistics.

Chai, J. Y., & Jin, R. (2004). Discourse structure for context question answering. In *HLT-NAACL 2004 workshop on pragmatics of question answering* (pp. 23–30).

Choi, F. (2000). Advances in domain independent linear text segmentation. In *Proceedings of NAACL-00* (p. 26-33).

Choi, F., Wiemer-Hastings, P., & Moore, J. (2001). Latent semantic analysis for text segmentation. In *Proceedings of 6th EMNLP* (p. 109-117).

Cochran, W. G. (1950). The comparison of percentages in matched samples. *Biometrika*, *37*, 256-266.

Cristea, D., Ide, N., Marcu, D., & Tablan, V. (2000). Discourse structure and co-reference: An empirical study. In *Proceedings of COLING 2000.*

Danlos, L. (2004). Discourse dependency structures as constrained DAGs. In *Proceedings of 5th SIGDIAL workshop on discourse and dialogue* (pp. 127–135).

Eisenstein, J. (2009). Hierarchical text segmentation from multi-scale lexical cohesion. In *Proceedings of NAACL09.*

Francis, W. N., & Kucera, H. (1979). *Brown corpus manual* (Third ed.). Brown University.

Grosz, B. J., Joshi, A. K., & Weinstein, S. (1995). Centering: a framework for modelling the local coherence of discourse. *Computational Linguistics*, *21*(2), 203–226.

Hearst, M. (1994). Multi-paragraph segmentation of expository text. In *32nd ACL* (pp. 9 – 16). New Mexico State University, Las Cruces, New Mexico.

Hirschberg, J., & Nakatani, C. (1996). A prosodic analysis of discourse segments in direction-giving monologues. In *34th ACL.*

Kaszkiel, M., & Zobel, J. (1997). Passage retrieval revisited. In *Proceedings of 20th ACM SIGIR* (pp. 178–185).

Kauchak, D., & Chen, F. (2005). Feature-based segmentation of narrative documents. In *Proceedings of the ACL workshop on feature engineering for machine learning in nlp.*

Kehler, A. (2002). *Coherence, reference and the theory of grammar*. CSLI Publications.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*, 259–284.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization*. MIT Press.

Over, P., & Liggett, W. (2002). *Introduction to DUC-2002: an intrinsic evaluation of generic news text summarization systems.* Slides, NIST.

Passonneau, R. J., & Litman, D. J. (1996). Empirical analysis of three dimensions of spoken discourse: Segmentation, coherence and linguistic devices. In E. Hovy & D. Scott (Eds.), *Computational and conversational discourse.* Springer-Verlag.

Passonneau, R. J., & Litman, D. J. (1997). Discourse segmentation by human and automated means. *Computational Linguistics*, *23*(1), 103-139.

Pevzner, L., & Hearst, M. (2001). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, *16*(1).

Polanyi, L., Culy, C., Berg, M. van den, Thione, G. L., & Ahn, D. (2004). A rule based approach to discourse parsing. In *Proceedings of SIGDIAL.*

Simpson, R. C., Briggs, S. L., Ovens, J., & Swales, J. M. (2000). *The Michigan corpus of academic spoken English.* Ann Arbor, MI: The Regents of the University of Michigan.

Slaney, M., & Ponceleon, D. (2001). Hierarchical segmentation: Finding changes in a text signal. *Proceedings of SIAM 2001 Text Mining Workshop*, 6-13.

Walker, M. A. (1997). Centering, anaphora resolution, and discourse structure. In A. K. J. Marilyn A. Walker & E. F. Prince (Eds.), *Centering in discourse.* Oxford University Press.

Webber, B. (2004). D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science*, *28*, 751-779.

Wolf, F., & Gibson, E. (2004). Representing discourse coherence: A corpus-based analysis. In *20th COLING.*

Yaari, Y. (1997). Segmentation of expository texts by hierarchical agglomerative clustering. In *Proceedings of RANLP'97.*

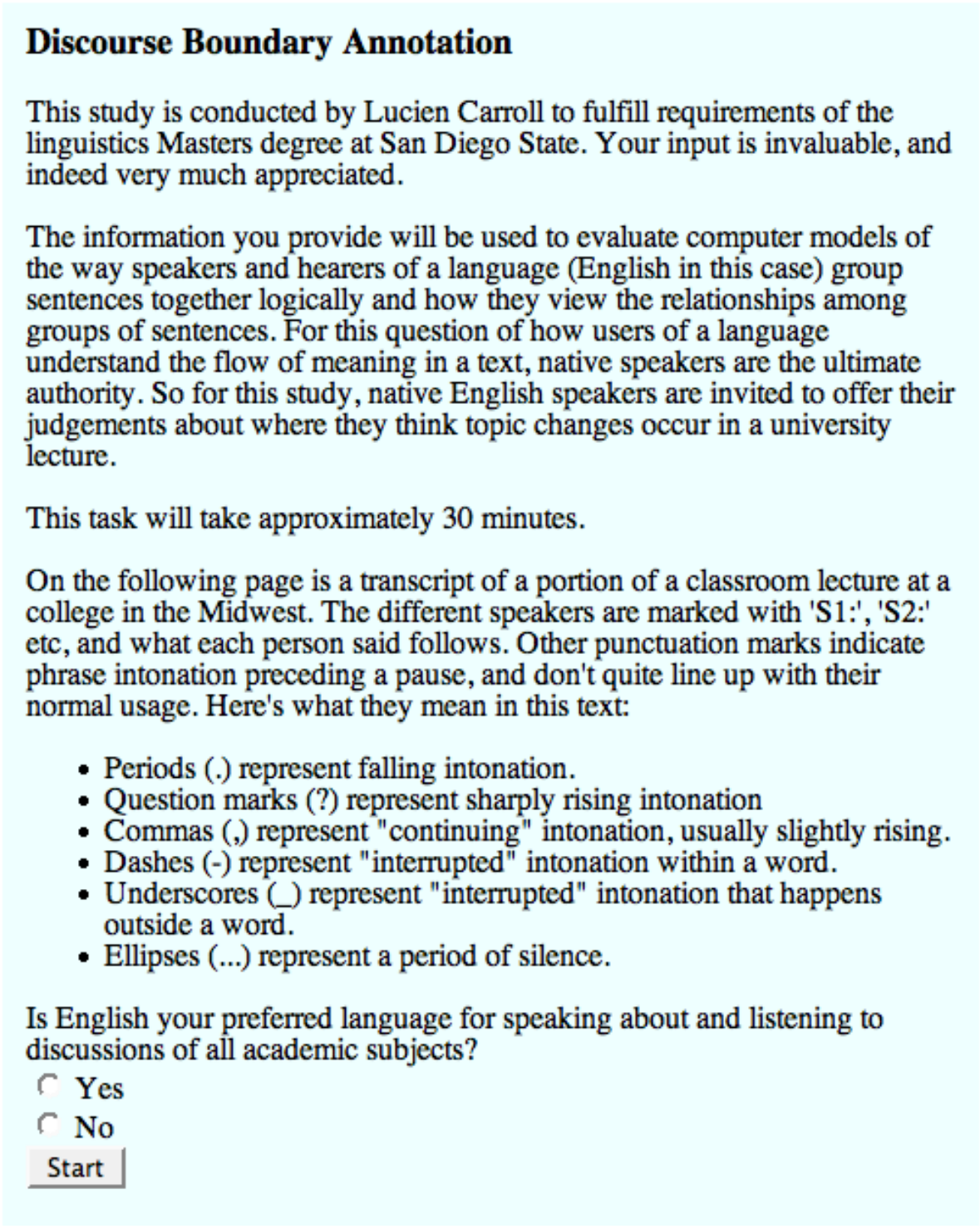


Figure 1. The first page of the annotation interface

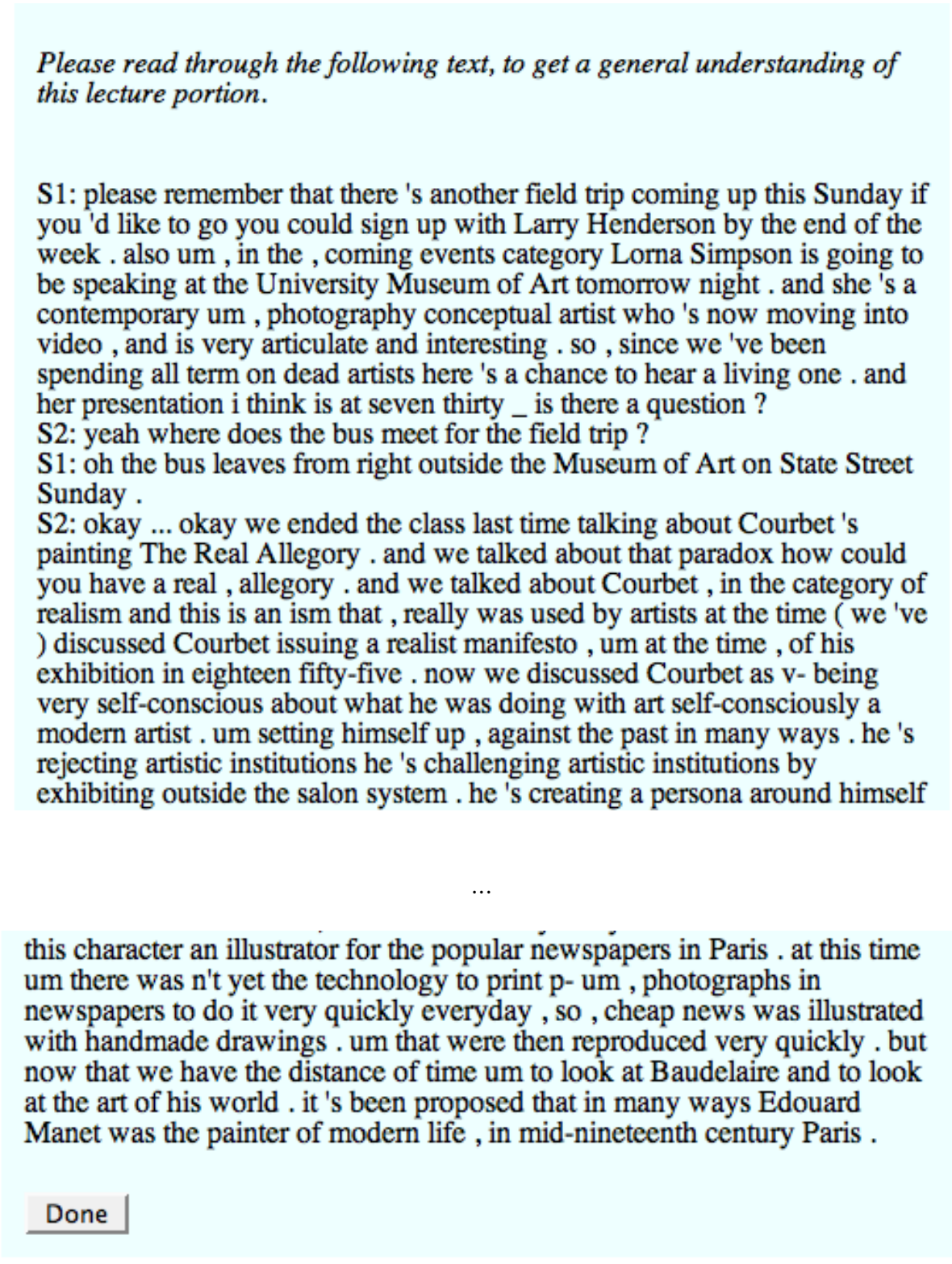


Figure 2. The second page of the annotation interface

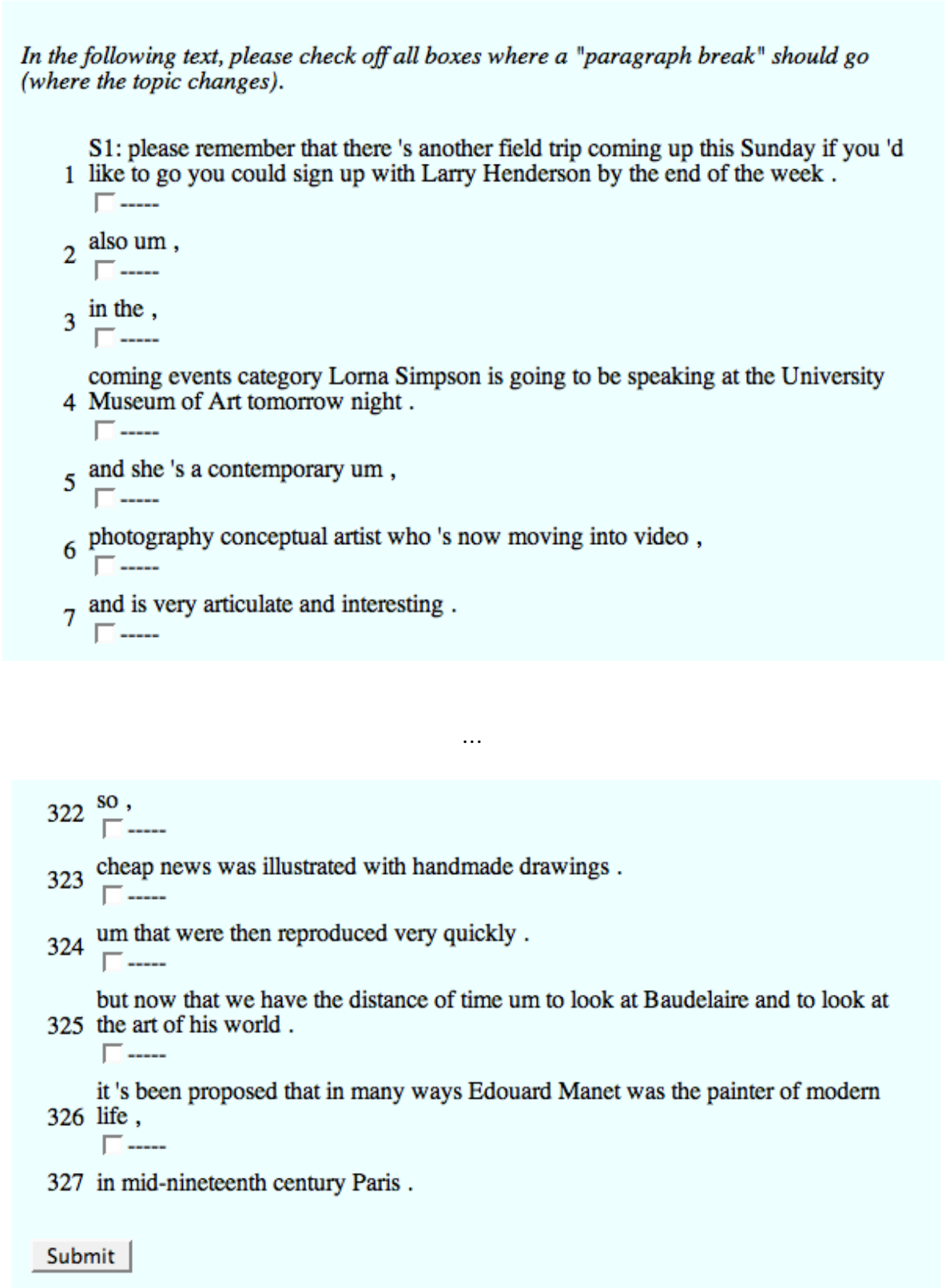


Figure 3. The third page of the annotation interface

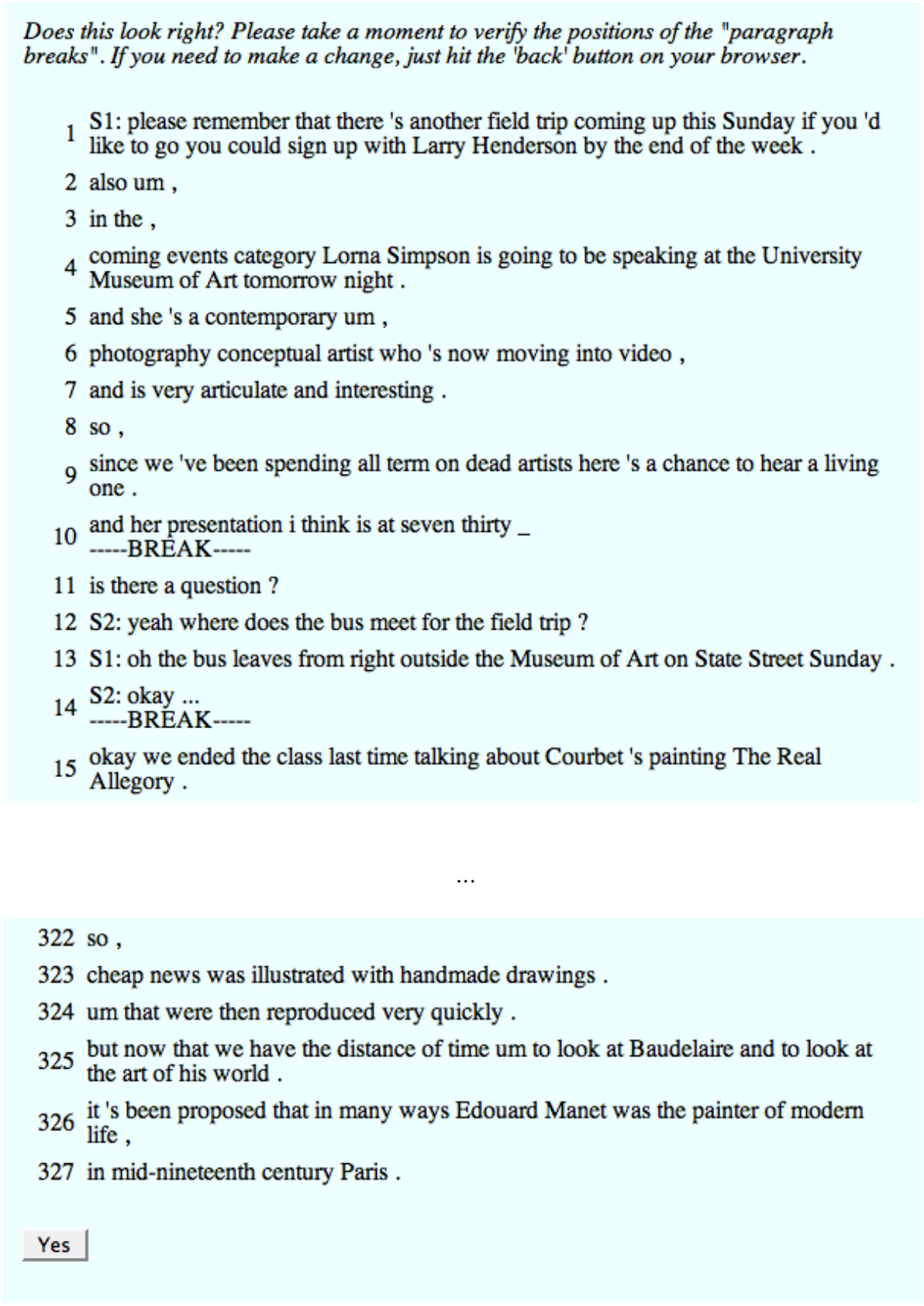


Figure 4. The fourth page of the annotation interface

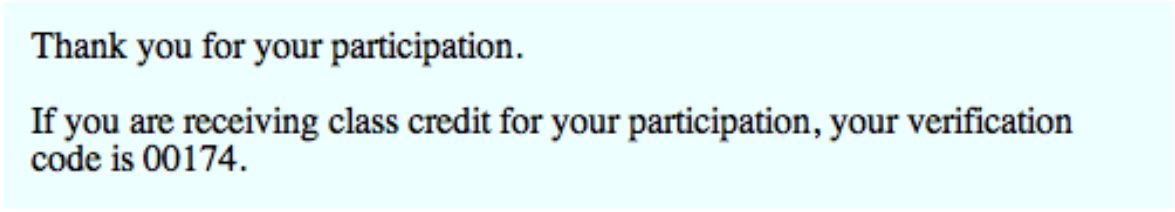


Figure 5. The last page of the annotation interface

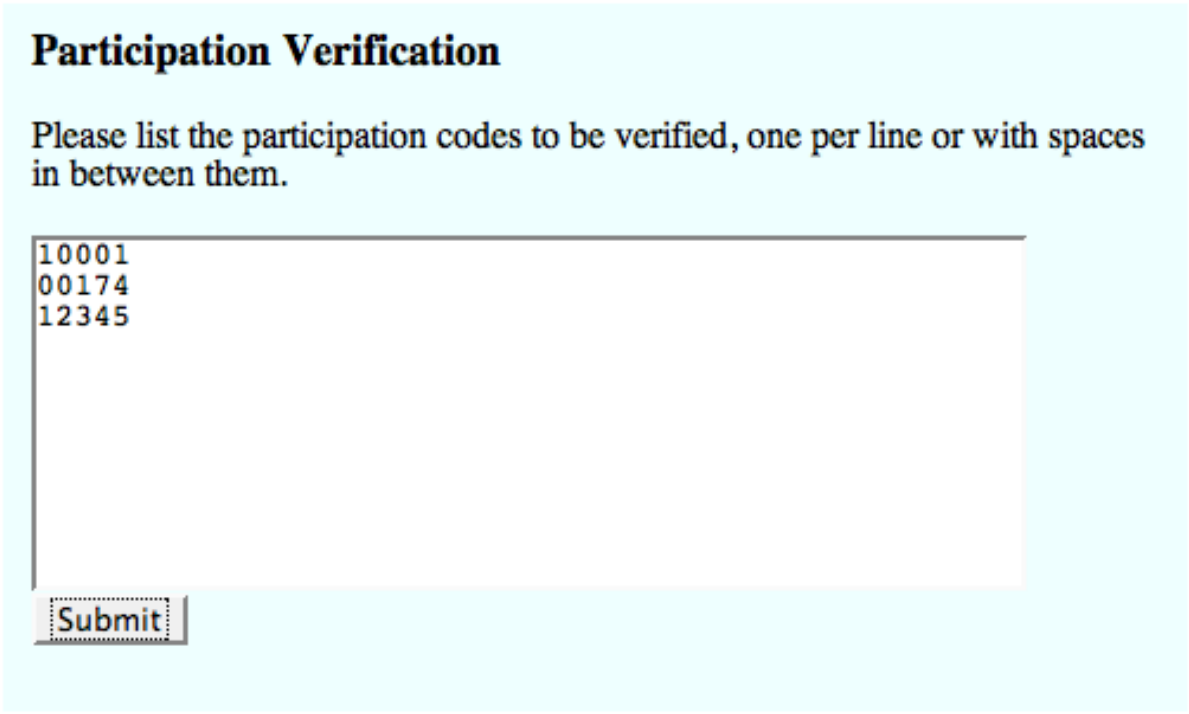


Figure 6. The first page of the participation validation interface

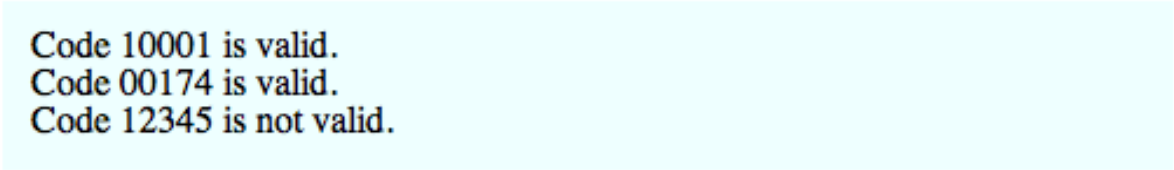


Figure 7. The second page of the validation interface