Evaluating Hierarchical Discourse Segmentation

Lucien Carroll
Linguistics Dept.
UC San Diego
San Diego, CA 92093
lucien@ling.ucsd.edu

Abstract

Hierarchical discourse segmentation is a useful technology, but it is difficult to evaluate. I propose an error measure based on the word error rate of Beeferman et al. (1999). I then show that this new measure not only reliably distinguishes baseline segmentations from lexically-informed hierarchical segmentations and more informed segmentations from less informed segmentations, but it also offers an improvement over previous linear error measures.

1 Introduction

Discourse segmentation is the task of identifying coherent clusters of sentences and the points of transition between those groupings. Discourse segmentation can be viewed as shallow parsing of discourse structure. The segments and the relations between them are left unlabeled, focusing instead on the boundaries between the segments (i.e., the bracketing).

Discourse segmentation is thought to facilitate automatic summarization (Angheluta et al., 2002; Boguraev and Neff, 2000), information retrieval (Kaszkiel and Zobel, 1997), anaphora resolution (Walker, 1997) and question answering (Chai and Jin, 2004). Automatic discourse segmentation, as shallow annotation of discourse structure, also provides a testing grounds for linguistic theories of discourse (Passonneau and Litman, 1997) and provides a natural unit of measure in linguistic corpora (Biber et al., 2004).

1.1 The structure of discourse

Research in discourse structure theory (Hobbs, 1985; Grosz and Sidner, 1986; Mann and Thompson, 1988; Kehler, 2002; Asher and Lascarides, 2003; Webber, 2004) and discourse parsing (Marcu, 2000; Forbes et al., 2003; Polanyi et al., 2004; Baldridge et al., 2007) has variously defined discourse structure in terms of communicative intention, attention, topic/subtopic structure, coherence relations, and cohesive devices. There is much disagreement about the units and elementary relations of discourse structure, but they agree that the structures are hierarchical, most commonly trees (Marcu, 2000), while others have argued for directed acyclic graphs (Danlos, 2004), or general graphs (Wolf and Gibson, 2004). In contrast, most of the segmentation research to date has focused on linear segmentation, in which segments are non-overlapping and sequential, and it has been argued that this sequence model is sufficient for many purposes (Hearst, 1994). I focus here on tree discourse segmentation, in which larger segments are composed of sequences of subsegments. This is potentially more informative and more faithful to linguistic theory than linear discourse segmentation is, but it poses a more challenging evaluation problem.

1.2 Hierarchical segmentation

Four studies have described hierarchical discourse segmentation algorithms, but none of them rigorously evaluated the segmentation in its hierarchical form. Yaari (1997) used a hierarchical clustering algorithm for hierarchical discourse segmentation, and to evaluate it, he linearized the tree (taking all boundaries equally) and compared the result-

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 993–1001,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

ing precision and recall to contemporary linear segmentation algorithms. Slaney and Ponceleon (2001) used scale-space segmentation (an image segmentation algorithm) on the discourse's trajectory in a Latent Semantic Indexing (LSI) space (Landauer et al., 1998). They evaluated the algorithm by visual comparison with the heading-subheading structure of the text. Angheluta et al. (2002) 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 (2009) 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.

Many state-of-the-art linear discourse segmentation algorithms also use hierarchical frameworks, making them applicable to hierarchical discourse 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 et al., 2001) applies the same procedure to a similarity matrix of LSI vectors. Using these algorithms for hierarchical discourse segmentation simply requires keeping record of the boundary ranking, but until now they have only been used for linear segmentation.

1.3 The Beeferman error measure

Studies of linear discourse segmentation have revealed that discourse boundaries are inherently fuzzy. Human annotators demonstrate frequent disagreement about the number of segments and exactly where the transitions between segments occur, while still demonstrating statistically significant agreement (Passonneau and Litman, 1997). Because of this, 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 bracketings. An 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 segment length in the reference segmentation, where distances are measured in words or sentences, depending on whether word boundaries or sentence boundaries are considered possible discourse segment boundaries. The error is the average disagreement, between the reference segmentation and the evaluated segmentation, about whether the two ends of the window are in the same segment. Formally,

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

where N is the total number of atoms (words or sentences) in the document, and k is the window width. The arguments r_i and h_i are the indices of the segments that contain atom i in the reference and hypothesized segmentations, respectively, and δ is the discrete delta function, evaluating to 1 if its arguments are equal and to 0 otherwise. Pevzner and Hearst (2001) 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 atoms. It is as sensitive to false positives as it is to false negatives, whereas P_k is more sensitive to false negatives.

There are still a few problems with these error measures. In penalizing false negatives and false positives equally, WindowDiff actually favors sparse segmentations. Whereas P_k scores the baseline strategies of no boundaries and all possible boundaries as within a few percent of 50% error, WindowDiff scores the all-boundaries baseline at 100% error for typical reference segmentations. Furthermore, in running the summation from $i = 1$ to $i = N-k$, both error measures count boundaries near the edges of the text less than boundaries near the middle of the text. A boundary that is $j < k$ atoms from the beginning or end of the text has weight $\frac{j}{k}$ relative to boundaries in the middle of the text. Finally, because of the hierarchical structure of many texts, it is quite possible that a reference segmentation might not include legitimate but fairly unimportant boundaries that a hypothesized segmentation does include. These unimportant boundaries should not count against the hypothesized segmentation, but in the linear segmentation paradigm, they necessarily do. The ideal error measure should distinguish more-informed algorithms from less-informed algorithms, treating equally uninformed baselines the same, and it should treat boundary placement errors according to the prominence of the boundaries, and not according to their positions within the text.

Building on work in evaluating linear segmentation, this study considers the evaluation of tree segmentations. I propose an error measure, derived from Beeferman et al.'s P_k (1999), for evaluating the alignment of a tree segmentation to a reference segmentation. I first show that this error measure is advantageous even for evaluating linear segmentations, and then I evaluate four hierarchical segmentation algorithms against a gold standard derived from encyclopedia articles.

2 A hierarchical measure

The proposed error measure is based on the intuition that prominent boundaries count more than less prominent boundaries. The hierarchical atom error rate E_{P_k} is the mean of Beeferman errors calculated over all linearizations of the segmentation tree (see Fig. 1). Assume a set R of reference boundaries and a set H of hypothesized boundaries each in rank order (prominent boundaries precede less prominent ones). The error is calculated as

$$E_{P_k} = \frac{1}{|R|} \sum_i c_i P_k(R_i, H_i)$$

where

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

The elements of H_i are chosen such that $|H_i| = |R_i|$ and no $b_n \in H \setminus H_i$ is more prominent than any $b_j \in H_i$. If the reference boundaries are completely ordered, then $c_i = 1$ for all ranks i, but if some reference boundaries share ranks, one P_k term is calculated for each rank level in the reference segmentation, and weighted (c_i) by the number of boundaries that were at that level. In the degenerate case of linear segmentation, all segments have the same rank, and E_{P_k} reduces to the original P_k.

When hypothesized boundaries share ranks, each affected term in the summation is theoretically the average over all combinations (n boundaries at the next rank *Choose* r boundaries to complete H_i). But 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.

When the set of hypothesized boundaries is smaller than the set of reference boundaries, we could simply permit H_i to be smaller than R_i for large values of i, 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, all at a single low rank, prevents incurring an undeserved penalty for false negatives.

In order to avoid undercounting boundaries near the beginning and end of the text, I consider the possibility of wrapping the window around from the beginning to the end of the text. In calculating P_k, the sum is understood to run from $i = 1$ to $i = N$, rather than stopping at $N - k$, and the atom index of the leading edge of the window $(i + k)$ generalizes to $((i + k) \bmod N)$.

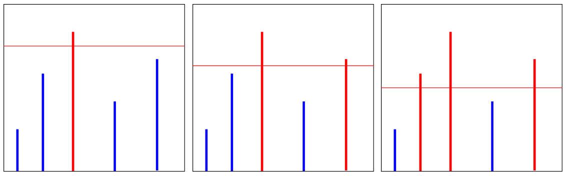

Figure 1: Sequential linearizations in computing hierarchical word error rate. The heights of the vertical lines represent the prominences of boundaries, and each horizontal line is one linearization. In the first step, only the highest boundary is used, producing just two segments. Each following step includes one more boundary.

3 Hierarchical replication of Choi et al.

As a preliminary test of the error measure, I evaluated two algorithms from Choi et al. (2001) 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 (Francis and Kucera, 1979). The following results are based on the T_{3-11} subset, 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 evaluate hierarchical segmentation algorithms with the hierarchical measure, to show that treating linear segmentation as a special case of hierarchical segmentation solves the issue of unequal treatment of false positives and false negatives, and running the WindowDiff sum to N (wrapping the window around to the beginning) solves the problem of undercounting the boundaries near the text edges.

3.1 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 and 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, so I used a constant number of segments, greater than in 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.

3.2 Results and Discussion

The calculated E_{P_k} error rates are displayed in Fig. 2.[2] The error for HC99 in Fig. 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. 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 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 E_{WD} hierarchical error rates (calculated according to WindowDiff) are consistently higher (Fig. 2c, d) than the corresponding E_{P_k}. WindowDiff

[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.

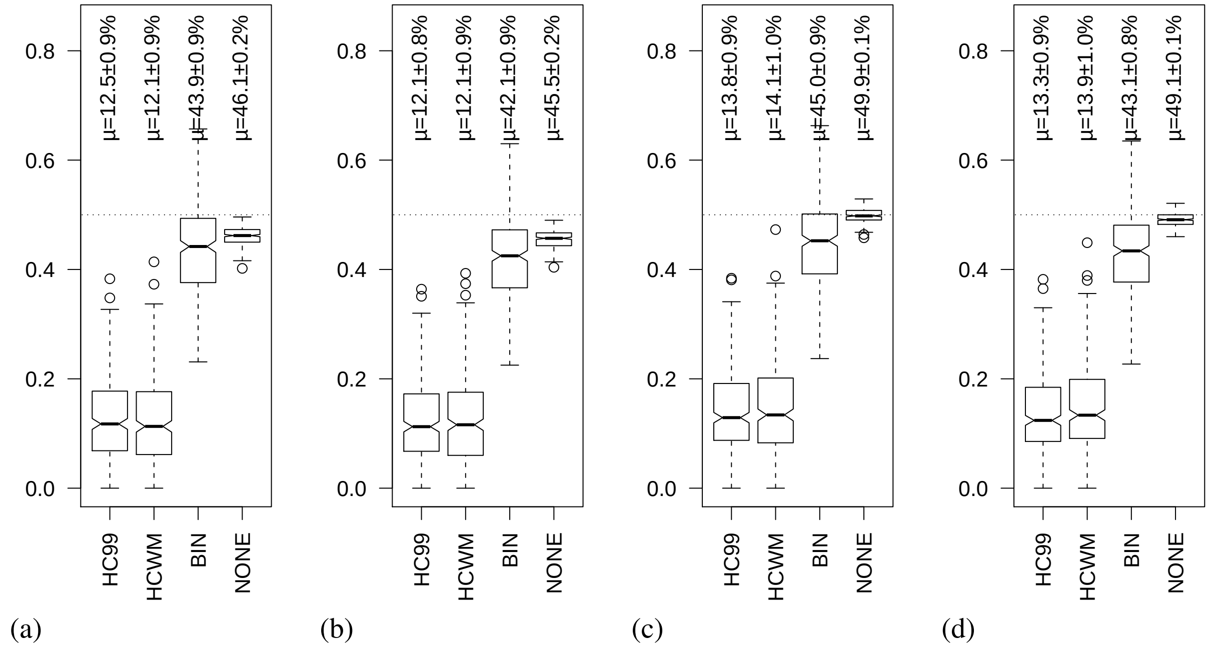

Figure 2: Distributions of E_{P_k} (a, b) and E_{WD} (c, d) for each of the hypothesized and baseline segmentation algorithms. The data in graphs (a) and (c) are calculated with sums that stop at $N-k$ (when the window reaches the end of the text), whereas (b) and (d) are calculated with sums that run to N (wrapping the window back to the beginning). The boxes indicate the quartiles, and the means with 95% confidence intervals are written above.

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 EP_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 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 E_{WD} error, extra boundaries are sampled or ignored, and so every false positive has a corresponding false negative, which limits the divergence between E_{WD} and E_{P_k} and keeps the E_{WD} error of informed segmentations below baseline errors. As with E_{P_k}, continuing the sum to N (Fig. 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.

3.3 Conclusion

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.

For these reasons, I adopt a version of E_{WD} that continues the sum to wrap the window around the end of the text. In addition, when I refer to the linear error measure in the following sections, I mean the special case of E_{WD} in which the information in the reference segmentation about the ranking of the marked boundaries is ignored, but boundary ranking information in the hypothesized segmentation (both marked and unmarked boundaries) is still used to select as many segment boundaries as are in the reference segmentation.

4 Wikipedia Evaluation

In this section, I compare the same two algorithms and baselines with two additional hierarchical segmentation algorithms, using a hierarchical reference segmentation. The reference segmentation corpus is derived from encyclopedia articles, and I use the hierarchical error measure developed in the previous sections. I also constrast the hierarchical error rates with measurements that ignore the boundary ranking information in the hypothesized or reference segmentations in order to highlight the difference between the performance on boundary position and the performance on boundary ranking.

4.1 Corpus and Algorithms

The evaluation corpus is derived from the *2006 Wikipedia CD* release.[3] The html pages were converted to flat text, removing boilerplate, navigation, info-boxes, and image captions. Heading text was replaced with a boundary marker, indicating the heading depth. The subcorpus used for this evaluation consists of articles with a heading depth of four (i.e. having html elements h2 through h5), a total of 66 articles. The texts were reformatted with an automatic sentence detector[4] to have one sentence per line, and then tokenized.[5]

In addition to the HC99 and HCWM algorithms used in the previous section, I use two algorithms described by Eisenstein (2009). The HIERBAYES algorithm (here, HBT) uses a multi-level latent topic model to perform joint inference over the locations and prominences of topic change boundaries. The GREEDY-BAYES algorithm (here, GBEM) uses a single-level latent topic model to find a linear segmentation, and recursively divides each of the segments.[6] Both algorithms internally decide the number of hypothesized boundaries, sometimes underestimating it and sometimes overestimating.[7]

4.2 Results and Discussion

The E_{WD} error rates for each of the hypothesized segmentations are presented in Fig. 3. As with the Choi data, the NONE baseline has an error rate at chance (50%), while the lexical algorithms perform better than that (highly statistically significantly ($p < .0001$) less than 50%, according to individual two-sided one-sample t-tests). However, they perform much worse than they did on the Choi data.

In spite of the relatively high error rates, the discriminating power of the evaluation measure is revealed by comparison of the fully hierarchical error rates (Fig. 3a) with the error rates that ignore the ranking information in the reference (Fig. 3b) or hypothesized (Fig. 3c) segmentations. For each of the lexical algorithms that were originally designed as linear segmentation algorithms (HC99, HCWM, and GBEM), the mean error is less in Fig. 3b against the linear standard (when reference segmentation boundary prominences are ignored) than in Fig. 3a under the fully hierarchical measure (two-tailed paired t-tests, each $p < .0001$). In contrast, HBT, designed as a hierarchical segmentation algorithm, obtains a lower error rate under the fully hierarchical E_{WD} measure (though the difference does

[3] Available from `http://schools-wikipedia.org/2006/`.

[4] From Ratnaparkhi's 'jmx' (`ftp://ftp.cis.upenn.edu/pub/adwait/jmx/jmx.tar.gz`).

[5] The evaluation code and corpus can be downloaded from `http://idiom.ucsd.edu/~lucien/segmentation`

[6] Both algorithms are part of the HBayesSeg package, available at `http://people.csail.mit.edu/jacobe/naacl09.html`

[7] Options for HBT were set to produce 3 levels of text-internal boundary prominence. Attempts to obtain more boundaries and more depth levels lead to deteriorated performance, because the search space grows geometrically with the number of levels (Eisenstein, p.c.)

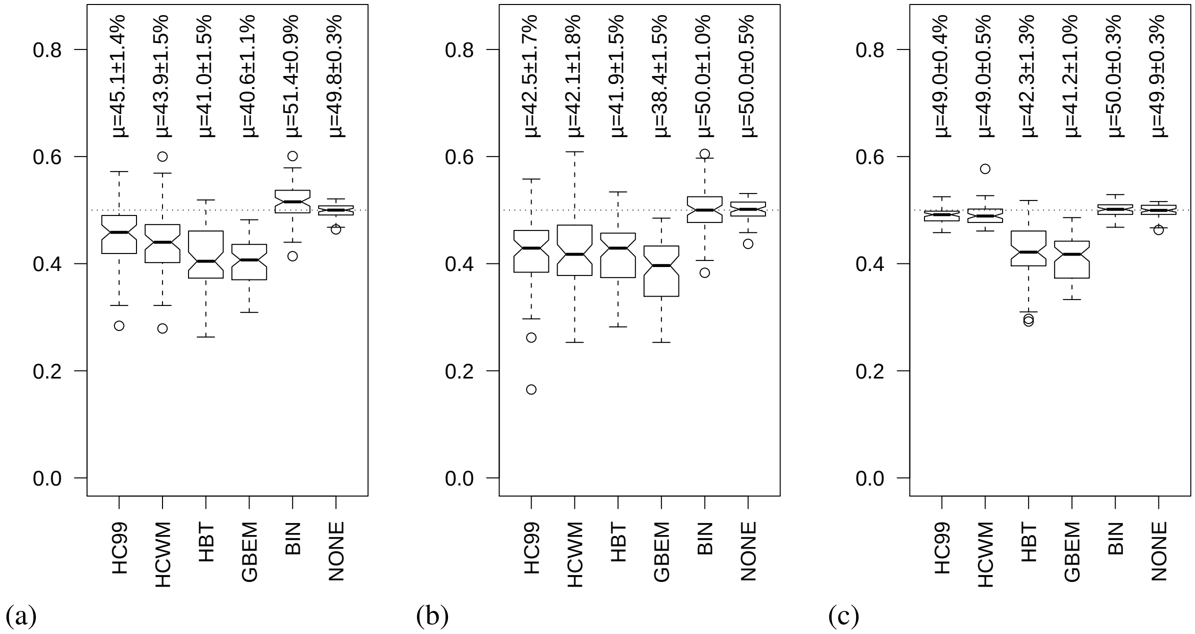

Figure 3: E_{WD} error rates for each of the segmentation algorithms. (a) Hierarchical error (b) Linear error (ignoring reference segmentation prominences) (c) Hierarchical error ignoring hypothesized segmentation prominences. Boxes show quantiles and means are written above, with 95% confidence intervals.

not reach significance: $p = 0.1$, two-tailed paired t-test). When instead the hypothesized boundary prominences are ignored (Fig. 3c), reducing them to linear segmentations but still evaluating against the hierarchical standard, the error rates of all the lexical algorithms are raised (in two-tailed paired t-tests, each $p < .0001$), but HBT and GBEM are only slightly affected, whereas HC99 and HCWM are almost raised to chance. While HBT and GBEM hypothesize about the same number of boundaries as the reference segmentation (13 and 22 text-internal boundaries on average, compared to 22 text-internal boundaries in the reference corpus), the HC99 and HCWM algorithms were made to hypothesize 54 boundaries for each text. The difference between their error rates in (Fig. 3a) and (Fig. 3c) shows that the HC99 and HCWM boundaries given the highest prominences corresponded much more closely to the reference boundaries than the hypothesized boundaries given the lowest prominences.

The mean scores for the BIN baseline are over 50% on the encyclopedia data. In contrast, the mean score for BIN on the Choi standard data (Fig. 2) was 45% for the linear measure and 43% for the hierarchical measure. Why did BIN do so poorly here when it performed well above chance on the Choi data? The difference is in the distributions of segment lengths. As seen in Fig. 4, the Choi data segment lengths are well-defined by their mean, because they were constructed with uniform distributions of segment length. On the other hand, the distribution of segment lengths in the encyclopedia data is more skewed, with many quite short segments and a few quite long segments.

The error rates for both HC99 and HCWM are much higher on the encyclopedia data than they are on the Choi data, and the error rates for HBT and GBEM are not much better. Choi's evaluation corpus was specifically designed to have obvious boundaries, whereas the boundaries in these discourse samples are much less obvious. As discussed by Kauchak and Chen (2005), even algorithms that obtain low error rates on newsfeed do not perform well on more fluid discourse, and while Ji and Zha (2003) reported quite low error on an expository text sample ($P_k = 12\%$), Kauchak and Chen (2005) report a best error rate of $P_k = 38.5\%$ on the encyclopedia corpus they used, and Malioutov and Barzi-

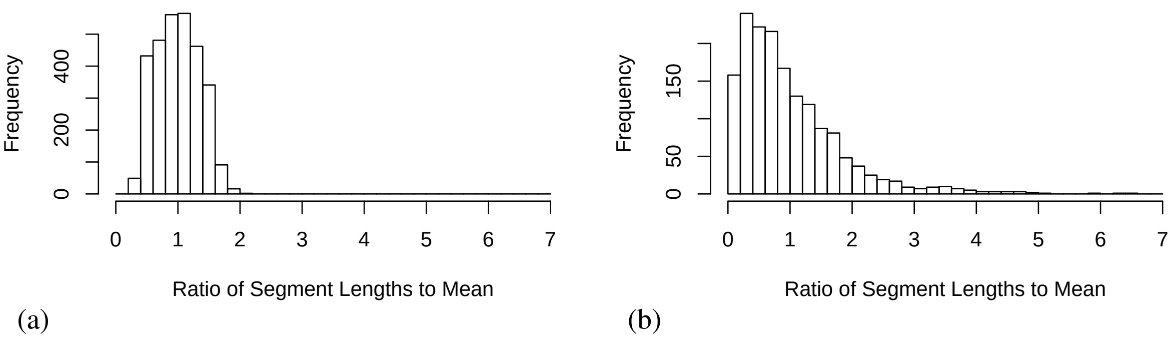

Figure 4: Distribution of sentences per segment for (a) Choi standard data (b) Wikipedia data

lay (2006) obtained P_k error rates between 30% and 40% on the lecture data they used, comparable to human annotator pairwise P_k ranging from 24% to 42%. C99 and CWM—like the other algorithms that make use of hierarchical representations of the text, such as Ji and Zha (2003) and Fragkou et al. (2004)—depend completely on lexical information. Another strand of research, including Galley et al. (2003) and Kauchak and Chen (2005), make use of a wide variety of linguistic and orthographic cues. And the discourse parsing systems take advantage of even more linguistic cues. The ideal segmentation algorithm needs to combine the advantages of each of these approaches, but the frameworks are not straightforwardly compatible. The Bayesian framework explored by Eisenstein and Barzilay (2008) is a potential route to a richer model, and they found their richer model beneficial for a meetings corpus but not for a textbook. The HBT and GBEM algorithms, which were based on that work, do not attempt to go beyond lexical cohesion, but it does provide a framework for hierarchical segmentation algorithms that take advantage of other cues.

5 Conclusions

In Section 2, 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 lexically-informed segmentations from baseline segmentations, and it distinguishes hierarchical segmentations from segmentations composed of the same boundaries but without the boundary ranking information. As a more reliable evaluation of both linear and hierarchical segmentation algorithms, this error measure will facilitate the development of more richly informed segmentation algorithms.

Acknowledgments

In this research, I have benefited from resources at both San Diego State University and UC San Diego. This work has been enriched by the questions and advice of many people, including Eniko Csomay, Rob Malouf, Lara Taylor, Rebecca Colavin, Andy Kehler, and the NAACL anonymous reviewers. I am also grateful to Freddy Choi and Jacob Eisenstein for making their code and data available, and to Jacob for additional help running his code. All errors are my own.

References

Roxana Angheluta, Rik De Busser, and Marie-Francine Moens. 2002. The use of topic segmentation for automatic summarization. In *DUC 2002*.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Guy Aston and Lou Burnard. 1998. *The BNC Handbook: Exploring the British National Corpus with SARA*. Edinburgh University Press.

Jason Baldridge, Nicholas Asher, and Julie Hunter. 2007. Annotation for and robust parsing of discourse structure of unrestricted texts. *Zeitschrift für Sprachwissenschaft*, 26(213):239.

Doug Beeferman, Adam Berger, and John D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Douglas Biber, Eniko Csomay, James K. Jones, and Casey Keck. 2004. A corpus linguistic investigation of vocabulary-based discourse units in university registers. *Language and Computers*, 20:53–72.

Branimir Boguraev and Mary S. Neff. 2000. Discourse segmentation in aid of document summarization. In *33rd HICSS*.

Marco Carbone, Ya'akov Gal, Stuart Shieber, and Barbara Grosz. 2004. Unifying annotated discourse hierarchies to create a gold standard. In *Proceedings of 4th SIGDIAL Workshop on Discourse and Dialogue*.

Joyce Y. Chai and Rong Jin. 2004. Discourse structure for context question answering. In *HLT-NAACL 2004 Workshop on Pragmatics of Question Answering*, pages 23–30.

Freddy Choi, Peter Wiemer-Hastings, and Johanna Moore. 2001. Latent semantic analysis for text segmentation. In *Proceedings of 6th EMNLP*, pages 109–117.

Freddy Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of NAACL-00*, pages 26–33.

Laurence Danlos. 2004. Discourse dependency structures as constrained DAGs. In *Proceedings of 5th SIGDIAL Workshop on Discourse and Dialogue*, pages 127–135.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of EMNLP 2008*.

Jacob Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proceedings of NAACL09*.

Katherine Forbes, Eleni Miltsakaki, Rashmi Prasad, Anoop Sarkar, Aravind Joshi, and Bonnie Webber. 2003. D-LTAG system: Discourse parsing with a lexicalized tree-adjoining grammar. *Journal of Logic, Language and Information*, 12(3):261–279, June.

P. Fragkou, V. Petridis, and Ath. Kehagias. 2004. A dynamic programming algorithm for linear text segmentation. *Journal of Int Info Systems*, 23:179–197.

W. Nelson Francis and Henry Kucera. 1979. *BROWN Corpus Manual*. Brown University, third edition.

Michael Galley, Kathleen McKeown, Eric Fossler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *41st ACL*.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

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

Jerry R Hobbs. 1985. On the coherence and structure of discourse. In *CSLI 85-37*.

Xiang Ji and Hongyuan Zha. 2003. Domain-independent text segmentation using anisotropic diffusion and dynamic programming. In *SIGIR'03*.

Marcin Kaszkiel and Justin Zobel. 1997. Passage retrieval revisited. In *Proceedings of 20th ACM SIGIR*, pages 178–185.

David Kauchak and Francine Chen. 2005. Feature-based segmentation of narrative documents. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*.

Andrew Kehler. 2002. *Coherence, reference and the theory of grammar*. CSLI Publications.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 25–32.

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

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. MIT Press.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139.

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

Livia Polanyi, Chris Culy, Martin van den Berg, Gian Lorenzo Thione, and David Ahn. 2004. A rule based approach to discourse parsing. In *Proceedings of SIGDIAL*.

Malcolm Slaney and Dulce Ponceleon. 2001. Hierarchical segmentation: Finding changes in a text signal. *Proceedings of SIAM 2001 Text Mining Workshop*, pages 6–13.

Marilyn A. Walker. 1997. Centering, anaphora resolution, and discourse structure. In Aravind K. Joshi Marilyn A. Walker and Ellen F. Prince, editors, *Centering in Discourse*. Oxford University Press.

Bonnie Webber. 2004. D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science*, 28:751–779.

Florian Wolf and Edward Gibson. 2004. Representing discourse coherence: A corpus-based analysis. In *20th COLING*.

Yaakov Yaari. 1997. Segmentation of expository texts by hierarchical agglomerative clustering. In *Proceedings of RANLP'97*.