Parameters for Topic Boundary Detection in Multi Party Dialogues

Margot Mieskes[1], **Michael Strube**[2]

[1]European Media Laboratory GmbH, Heidelberg, Germany
http://www.eml-d.de/english/homes/mieskes

[2]EML Research gGmbH, Heidelberg, Germany
http://www.eml-research.de/∼strube

Abstract

We present a topic boundary detection method that searches for connections between sequences of utterances in multi party dialogues. The connections are established based on word identity. We compare our method to a state-of-the art automatic topic boundary detection method that was also used on multi party dialogues. We checked various methods of preprocessing of the data, including stemming, lemmatization and stopword filtering with a text-based as well as speech-based stopword lists. Using standard evaluation methods we found that our method outperformed the state-of-the art method.

1. Motivation

The task of summarizing meetings requires a chain of preprocessing steps (see e.g. Zechner (2002)). Dialogue segmentation is crucial especially for longer meetings. In this paper we introduce a dialogue segmentation module that automatically puts topic boundaries in meeting data[1]. As reference data we use the ICSI Meeting Recorder Project Data (*ICSI data*) (Janin et al., 2003).

Recent work on automatic dialogue segmentation was done not only on text, but also on meeting data (Galley et al., 2003; Georgescul et al., 2006). The first uses Lexical Cohesion for detecting topic breaks, based on word identiy. The second uses Support Vector Machines trained on manually annotated data. The features are based on $tf * idf$, $tf * normal$ and $log * entropy$ of word frequencies.

The motivation behind our work was to develop a method to test the influence of various parameters (e.g. stemming vs. lemmatizing) in order to find a reliable method for topic boundary detection. The motivation behind topic boundary detection in general is to be able to divide the meetings into topically related subsegments. The project in which framework these experiments were carried out aims at the automatic summarization of multi party dialogues. Following manual procedures to summarize meetings, topic boundaries have to be detected, especially for longer meetings. Our hypothesis is that smaller entities of the meeting are easier to summarize than the whole meetings. Therefore a reliable dialogue segmentation method is necessary.

2. Related Work

A lot of work has so far been done on the task of automatic topic boundary detection. The approaches fall into three categories: the first type of approach works with statistical and supervised learning methods (Kan et al., ; Passonneau & Litman, 1997; Reynar, 1999; Beeferman et al., 1999; Utiyama & Isahara, 2001; Tür et al., 2001; Lagus & Kuusisto, 2002). The second type of approach is based on lexical cohesion via lexical chain building using external knowledge sources (like thesauri) (Morris & Hirst, 1991; Kozima, 1993; Foltz et al., 1998; Landauer et al., 1998; Galley et al., 2003; Stokes, 2003; Popescu-Belis et al., 2004; Olney & Cai, 2005). The third type combines statistical with similarity measures as e.g. Hearst (1997) and Choi (2000). Two most recent approaches are presented by Hsueh et al. (2006), where the approach presented by Galley et al. (2003) is extended in two ways: first, it used automatic speech recognition output, and second it aimed at detecting subtopic boundaries as well as maintopic boundaries. Another recent approach is presented by Georgescul et al. (2006), where Support Vector Machines were used.

These approaches have several drawbacks. The supervised learning based approaches need annotated data. These can either be artificial, as in the research which uses concatenated texts or breaks provided by the authors of the texts (e.g. chapters in a book) or they can be manually annotated. This takes time and in general there are few topic breaks within one document, resulting in a data bottleneck problem. Additionally with manual annotation there is the problem of evaluating the manual annotation. Similarity measures use external knowledge sources such as a thesaurus. These provide some improvement but as with most collections they are limited and mostly rather general in context, which can be a problem when dealing with specialized texts. By combining statistical methods and similarity measures TextTiling (Hearst, 1997) was able to overcome these limitations somewhat. But most approaches so far have only been developed and tested on written texts rather than speech or transcribed speech.

3. The ICSI Meeting Recorder Project Data

The meetings that serve as basis for this work have been collected within the ICSI Meeting Recorder Project (Janin et al., 2003) (*ICSI data*). This collection contains 75 meetings recorded at ICSI during research meetings.

The data has been transcribed manually and divided into "segments". These are turn-like elements, but very often they interrupt sentences and thoughts. Therefore, we created an additional division called "spurts" (Shriberg et al., 2001). If, for a certain speaker, the pause within his/her speech is longer than $500\ ms$, the amount of speech be-

[1]The work reported in this paper was done while the first author was affiliated with EML Research gGmbH.

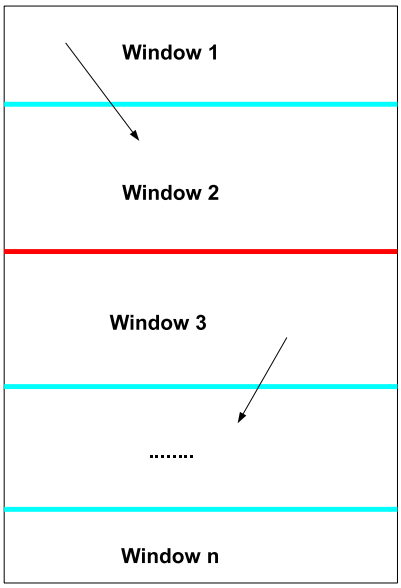

Figure 1: Illustration of the DiaSeg Method

tween two such pauses is one ”spurt”. Details on how the spurts were extracted can be found in Müller (2006).

4. DiaSeg vs. LcSeg

LcSeg (Galley et al., 2003) is based on the idea that lexical chains can represent the discourse structure. Lexical chains are so called “sequences of related words” which “provide the lexical cohesive structure of a text” (Barzilay & Elhadad, 1999).

Morris & Hirst (1991) described the first computational model for creating lexical chains, which was not implemented. The authors used lexical cohesion relations, which are (in their description) categories, index entries and pointers in Roget's Thesaurus. Chains are created by taking a new word from the text and finding a related chain for it according to relatedness. The distance between occurrences of related words are taken into account as well. The strength of a chain is determined using repetition, density and length. This method has the advantage that finding the appropriate chain for a word is equivalent to disambiguating the word. But Morris & Hirst (1991) did not require the words in the chains to which they belonged to have the same sense.

Hirst & St-Onge (1998) proposed a method for using WordNet (Fellbaum, 1998) for building lexical chains. WordNet is organised in synonym sets (synsets), which are the sets of all the words sharing a common sense. Polysemous words appear in more than one synset. Words of the same category are linked through semantic relations like synonymy and hyponymy.

Galley et al. (2003) used only identiy between terms to form the chains. Afterwards the chains are divided into subchains. A weighting scheme is also applied, where chains are weighted based on frequency and compactness. Next, the lexical cohesion is calculated at each turn break. The cosine similarity is calculated based on lexical chains that overlap two adjacent windows. The resulting function is smoothed and each local minimum in this function is treated as a possible boundary. Based on maxima of cohesion on both sides of the minimum a hypothesized segmentation probability is calculated. Finally, the boundaries with the highest probability are selected (Galley et al., 2003).

Our own segmentation algorithm *DiaSeg* is also based on word identity, but instead of building all chains and analyze probabilities of boundaries, we place boundaries at the same time as analyzing the texts. We use several parameters in exploring our method. ”Morphology” can either be stemming or lemmatizing. ”Filter” can either be based on a stopwordlist for text or for speech. After both preprocessing steps have been performed, we walk through the text and determine whether to place a boundary or not. The idea behind our method is that if two windows of data are connected, then there should not be a boundary between them. As every turn is a potential boundary, the method walks turn-wise through the whole meeting. Two windows of the same size are checked whether they are connected or not. If they are connected the method moves forward one turn and inspects the next two windows. If they are not connected, a boundary is placed.

Figure 1 shows an illustration of the *DiaSeg* method. As soon as a connection between two windows is detected, the two windows are assumed to belong together and a boundary is placed. The algorithm moves one step further. In case no connection is found between the two windows a boundary is placed between the two windows. This is illustrated between Window 2 and Window 3. Window 1 and Window 2 share at least one word and therefore they belong together.

5. Evaluation with P_k and WD

For the evaluation of the automatic method two metrics are used here: P_k (Beeferman et al., 1999) and WindowDiff (WD) (Pevzner & Hearst, 2002). Both have been developed to take into account that topic boundaries do not necessarily match exactly.

Beeferman et al. (1999) describes an error metric as a formalisation that one segmenter is better than the other if it more relyably identifies when two sentences belong to the same document and when they do not. P_D (which is the general form of P_k) calculates the “probability that two sentences drawn randomly from the corpus are correctly identified”. Formula 1 shows how this is being calculated between two topic annotations ref and hyp where δ_{ref} is an indicator function and δ_{hyp} is one if the two indices are hypothesised to belong to the same document and zero otherwise. The function D is a distance probability distribution, over the set of possible distances between sentences chosen randomly from the corpus.

Beeferman et al. (1999) present several possibilities for D. If D is uniform across the text, the metric might be too forgiving. Another possibility is $D = E_\mu$, which is an exponential distribution with μ^{-1} fixed to the mean document length for the domain. In practice $D = k$ is used, where the window size k is half the average segment length in words.

$$P_d(ref, hyp) = \sum_{1 \leq i \leq j \leq n} D(i,j)(\delta_{ref}(i,j) \; xnor \; \delta_{hyp}(i,j)) \quad (1)$$

$$WindowDiff(ref, hyp) = \frac{1}{N-k} \sum_{i=1}^{N-k} (|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0) \quad (2)$$

With this metric a number between 0 and 1 is achieved, where 0 is reached only if the boundaries match exactly.

Pevzner & Hearst (2002) discuss an alternative to P_k. They argue that P_k has a "flawed" fundamental premise and "significant drawbacks". Apart form the problems P_k has on penalising errors, the authors also argue, that P_k is unintuitive. Although a perfect systems scores $P_k = 0$ and baseline systems (like putting a topic break everywhere or none at all) score $P_k = 0.5$ it is "not clear how the scores are scaled".

Pevzner & Hearst (2002) offer an alternative to P_k, namely WindowDiff (WD), which is based on P_k. WD works as follows: "For each position of the probe, simply compare the number of reference segmentation boundaries that fall in this interval (r_i) with the number of boundaries that are assigned by the algorithm (a_i). The algorithm is penalised if $r_i \neq a_i$, which is computed as $|r_i - a_i| > 0$". Formula 2 shows the formula behind this description, where $b(i,j)$ represents the number of boundaries between positions i and j in the text and N represents the number of sentences in the text.

6. Experiments

In these experiments several parameters of *DiaSeg* were tested and evaluated. First, we tested the segmentation. In order to evaluate the differences between spurt- and segment-based annotations, we used the manual annotation provided by Galley et al. (2003) in its original version, which was done on segments. For the comparison of the spurt based data we used a gold standard generated over the manual annotation done on 12 meetings by at least 2 human annotators. The reasons for the usage of spurts was that segments very often interrupt thoughts and sentences. Therefore, the baseline of using every segment as a potential boundary is not exactly correct, as many segments are part of a larger sentence or part of a set of sentences. Therefore, using spurts as a basis of potential boundaries is more accurate.

Second we tested filtering for two different stop words lists, one that was based on texts and another one that was based on speech. The reason why we wanted to test these two lists independent of each other and not rely on one list lies in the data. Although multi party dialogues are speech, their language differs from casual telephone conversations. There are portions where the conversations are very casual, but in some occasions the language is quite formal, for example during a presentation. Therefore, it cannot be assumed, that the stopword list for speech automatically performs better than the one for text.

The third parameter was the usage of different methods for morphological analysis. Both lemmatization and stemming was used in the past, but as we use a different approach, testing which of the two performs better in our application was necessary to be tested in order to get the best quality possible.

6.1. Manual Annotation

Three human annotators were asked to mark topic boundaries in 12 randomly selected meetings from the whole corpus. Detecting topic boundaries in meetings is not a trivial task even for humans. In our data the boundaries between topics are rarely marked explicitly. In some cases statements like "the next topic ..." occur, but in most cases the annotators have to rely on implicit clues in detecting boundaries or they have to place them somewhere in a slow transition from one topic to another. Therefore, clear-cut true vs. false metrics like κ do not work in these cases. The evaluation with κ would run into several problems. First, with topic boundaries it is difficult to say whether a topic boundary is correctly placed or not. Second, a meeting has 1,000 possible topic boundaries, but only 10 real boundaries. So even if the annotators disagree on all topic boundaries, they do agree that 980 possible boundaries are none. Consequently κ will be close to or below 1. On the other hand if only the boundaries are considered which were marked by both annotators it can easily be the case that they agree on few or none at all. Therefore, κ would be close to 0.

In Galley et al. (2003) Cochran's Q was suggested to provide a method to evaluate manual annotations for this case. Cochran's Q tests whether the assignment of boundaries by the human annotators is randomly distributed. This evaluation method has one advantage as compared to other tests: it can be even used when there is only a small amount of samples tested. Formally, Q is expressed as follows:

$$Q = \frac{(m-s)\left(m * \sum_{j=1}^{m} T_j^2 - \left(\sum_{j=1}^{m} T_j\right)^2\right)}{m * \sum_{i=1}^{N} L_i - \sum_{i=1}^{N} L_i^2}, \quad (3)$$

where N is the number of annotators, m the number of examples L_i is the number of positive answers of annotator i and T_j is the number of positive answers for all annotators for the jth example. The resulting number gives the degrees of freedom under which the significance level can be checked in the appropriate table.

Galley et al. (2003) report that on 19 of 25 (76%) meetings manually annotated for topic boundaries the interannotator reliability is significant on the 0.05 level. In our data we found that 9 out of 12 (75%) meetings, which were manually annotated, showed an interannotator reliability that is significant on the 0.05 level.

From the manual annotation we derived a Gold Standard annotation. This is used for the evaluation of the automatic method. The three manual annotations were examined and a Gold Standard annotation was put in the meeting if two

annotators placed a boundary within 10 segments of each other.

6.2. Results

Table 1 shows results on applying *LcSeg* both to the segment- and the spurt-based data. The results are comparable to results reported by Galley et al. (2003) for the same dataset. The single results though vary greatly in range. The results in Table 1 serve as a baseline for our experiments. The figures indicate that there is a 20% chance that a boundary is wrong, if this method is used to place topic boundaries. Using spurts the results indicate that the chance is closer to 30% that a boundary is wrong.

	Segment		Spurt	
File	P_k	WD	P_k	WD
mean	0.199	0.236	0.327	0.367
min	0.060	0.096	0.191	0.220
max	0.409	0.453	0.417	0.458

Table 1: Comparison between spurt-based evaluation and segment-based evaluation using *LcSeg* and comparing to manual annotation. Results reported in the original work are also given for comparison.

In the first experiment we wanted to determine the best window size to determine connections between windows.

Table 2 shows first results for applying *DiaSeg* to the data without pre-processing. For segments we achieve very good results with a window size of 40 segments. Results for smaller window sizes are also comparable to results reported in (Galley et al., 2003). The right half of the table shows the results for the spurt-based data. The best results are achieved with a window-size of 30 spurts. Comparing these results with the baselines we can see that for segments there is an improvement, but for spurts there is none, when evaluating with P_k. Using WD shows an improvement on most windowsizes.

	Segment		Spurt	
window	P_k	WD	P_k	WD
10	0.205	0.234	0.362	0.375
20	0.192	0.201	0.343	0.347
30	0.189	0.200	0.340	0.343
40	0.175	0.189	0.346	0.352
50	0.189	0.205	0.358	0.369
60	0.200	0.210	0.354	0.364
70	0.196	0.206	0.345	0.358
80	0.201	0.211	0.350	0.361
90	0.196	0.203	0.350	0.364

Table 2: Determining the window size both for spurts and for segments on plain text using *DiaSeg*.

As mentioned before, we experimented with various parameters. These parameters were morphological analysis and filtering. One filtering was based on a text stopword list (*StopSpeech*) and the second filtering was based on a speech stopword list (*StopText*). Both were provided by *LcSeg*. First, we tested the parameters independent of each other with *DiaSeg*.

Table 3 shows results for independent testing of the parameters stemming and filtering for *StopText* and *StopSpeech*. Lemmatizing performed considerably worse and was therefore not further considered. The results are given for the best window size in segments and spurts respectively. As the table shows the results are not very different for the segment-based data, apart from the window size, which increases considerably with *StopSpeech* filtering. But comparing these results with results from *LcSeg* we see an improvement.

	Segments		Spurts	
	P_k	WD	P_k	WD
Stem	0.171	0.178	0.340	0.342
min	0.059	0.073	0.154	0.166
max	0.300	0.337	0.418	0.420
windowSize	40	40	30	30
StopText	0.177	0.189	0.338	0.340
min	0.062	0.062	0.155	0.162
max	0.336	0.395	0.421	0.421
windowSize	40	40	30	30
StopSpeech	0.186	0.203	0.350	0.355
min	0.057	0.057	0.155	0.155
max	0.313	0.313	0.455	0.480
windowSize	90	90	60	60

Table 3: Results for parameters independently of each other on segment- and spurt-based data using *DiaSeg*

For the spurt-based data using stemming alone does not improve the results. Filtering for *StopText* improves the results slightly as compared to Table 2, but they still do not reach the results comparable to those achieved with *LcSeg*. Filtering for *StopSpeech* does not only increase the window size but it also worsens the results.

Table 4 shows the results for the segment-based data (left half) and the spurt-based data (right half), using both P_k (upper half) and WD (lower half). Here the data is used with *DiaSeg* combining stemming with the two filtering methods. All results are averaged over the available data and compared with the respective manual annotations.

For the segment-based data we observe that compared to Table 3 the filtering with *StopText* combined with stemming gives an improvement for the same window size (40 segments). Using *StopSpeech* gives a more substantial improvement, but at a larger window size (90 segments). Additionally, compared to Table 3 stemming combined with filtering improves results, as compared to only stemming or only filtering.

Using stemming and filtering for *StopText* improves the results considerably, even in comparison with *LcSeg*. A window size of 60 is already enough to achieve the results achieved by *LcSeg*, but a window size of 90 improves the results further. The data indicate that an even larger window size could give a further improvement. However, in our experience this is not the case, in particular for shorter meetings. When filtering for *StopSpeech* the best results are achieved with a window size of 60. A further extension of the window size does not improve the results. For P_k the results are slightly below the results achieved with *LcSeg*, but for WD they are considerably better than for *LcSeg* and also better than when filtering for *StopText*.

	P_k			
	Segments		Spurts	
window	*StopText*	*StopSpeech*	*StopText*	*StopSpeech*
10	0.531	0.793	0.462	0.608
20	0.203	0.528	0.406	0.462
30	0.177	0.338	0.378	0.382
40	0.174	0.226	0.351	0.348
50	0.177	0.189	0.334	0.339
60	0.189	0.181	0.321	0.329
70	0.195	0.185	0.321	0.333
80	0.198	0.174	0.315	0.337
90	0.190	0.171	0.311	0.343
	WD			
	Segments		Spurts	
10	0.626	0.950	0.502	0.879
20	0.222	0.630	0.420	0.545
30	0.194	0.383	0.394	0.401
40	0.184	0.248	0.380	0.357
50	0.187	0.205	0.362	0.348
60	0.192	0.196	0.351	0.341
70	0.197	0.195	0.352	0.349
80	0.202	0.207	0.353	0.351
90	0.194	0.174	0.346	0.359

Table 4: Results for *DiaSeg* on segment and spurt based data, using various window sizes and combining stemming with different filtering methods

7. Conclusions

In this paper we reported several results. First, we showed that results depend highly on various factors: segmentation – (segments vs. spurts), filtering – (with a stopword list based on text (*StopText*) vs. a stopword list based on speech (*StopSpeech*)) and second whether the words are stemmed or lemmatized.

These factors were tested, using a computationally simple method *DiaSeg* which was compared to a state-of-the art tool *LcSeg*. *DiaSeg* relies on the idea that links between windows of text indicate that the portions of text covered by each window belong together. Even without preprocessing, we found that this method outperforms the reference methods. By applying various pre-processing steps to the data, we were able to further improve the results. On the segment-based data we found the combination of stemming and filtering with *StopText* best and for spurts the combination of stemming and filtering with *StopSpeech* gave the best results.

One reason why *DiaSeg* performed better than *LcSeg* is that *DiaSeg* places fewer boundaries than *LcSeg*. In average *DiaSeg* places 7.78 boundaries/meeting, whereas *LcSeg* places 8.32 boundaries/meeting. The manual annotation has in average 5.96 boundaries. The results indicates that *DiaSeg* not only places fewer boundaries, but also that the few boundaries are also placed correctly.

For future work testing and extending *DiaSeg* to text (e.g. Wall Street Journal, etc.) would be worthwhile to show whether this approach is competitive in this domain as well. Additionally, more sophisticated methods for finding links between adjacent windows could be used. Finally, the reasons for the differences between segment-based and spurt-based data should be explored further. As both evaluation methods have some drawbacks testing a method proposed by Georgescul et al. (2006) would be interesting as well.

Acknowledgments. This work has been supported by the DFG under grant STR 545/2-1,2 within the DIANA-Summ project and by the Klaus Tschira Foundation.

References

Barzilay, Regina & Michael Elhadad (1999). Using lexical chains for text summarization. In Inderjeet Mani & Mark T. Maybury (Eds.), *Advances in Text Summarization*, Chp. 10, pp. 111–121. Cambridge, Massachusetts, USA: MIT Press.

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

Choi, Freddy (2000). Advances in independent linear text segmentation. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, Seattle, Wash., 29 April – 3 May, 2000, pp. 26–32.

Fellbaum, Christiane (Ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, Mass.: MIT Press.

Foltz, Peter W., Walter Kintsch & Thomas K. Landauer (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25:285–307.

Galley, Michael, Kathleen R. McKeown, Eric Fosler-Lussier & Hongyan Jing (2003). Discourse segmentation of multiparty conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 7–12 July 2003, pp. 562–569.

Georgescul, Maria, Alexander Clark & Susan Armstrong (2006). Word distributions for thematic segmentaion in a support vector machine approach. In *Proceedings of the CoNLL-X conference held at HLT-NAACL 2006*, New York, NY, USA, pp. 101–108.

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

Hirst, G. & D. St-Onge (1998). Lexical chains as representation of context for the detection and correction malapropisms. In C. Fellbaum (Ed.), *WordNet. An Electronic Lexical Database*, Chp. 13, pp. 305–332. Cambridge, Mass.: MIT Press.

Hsueh, Pei-Yun, Johanna D. Moore & Steve Renals (2006). Automatic segmentation of multiparty dialogue. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, 3–7 April 2006, pp. 273–280.

Janin, Adam, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke & Chuck Wooters (2003). The ICSI meeting corpus. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, China, 6–10 April 2003, pp. 364–367.

Kan, Min-Yen, Judith L. Klavans & Kathleen R. McKeown. Linear segmentation and segment significance. In *6th Workshop on Very Large Corpora, Montral, Canada, 5–16 August 1998*, pp. 197–205.

Kozima, Hideki (1993). Text segmentation based on similarity between words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 22–26 June 1993, pp. 286–288.

Lagus, Krista & Jukka Kuusisto (2002). Topic identification in natural language dialogues using neural networks. In *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*, Philadelphia, Penn., July 2002, pp. 95–102.

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

Morris, Jane & Graeme Hirst (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–46.

Müller, Christoph (2006). Automatic detection on nonreferential *it* in spoken multi-party dialog. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, 3–7 April 2006, pp. 49–56.

Olney, Andrew & Zhigiang Cai (2005). An orthonormal basis for topic segmentation in tutorial dialogue. In *Proceedings of Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*, pp. 971–978.

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

Pevzner, Lev & Marti Hearst (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Popescu-Belis, Andrei, Alexander Clark, Maria Georgescul, Denis Lalanne & Sandrine Zufferey (2004). Shallow discourse processing using machine learning algorithms (or not). In S. Bengio & H. Bourlard (Eds.), *Machine Learning for Multimodal Interaction*, Vol. 3361, Springer Lecture Series in Computer Science, pp. 277–290. Springer.

Reynar, Jeffrey C. (1999). Statistical models for topic segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Md., 20–26 June 1999, pp. 357–364.

Shriberg, Elizabeth, Andreas Stolcke & Don Baron (2001). Observations on overlap: Findings and implications for automatic processing of multi-party conversation. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH '01)*, Aalborg, Denmark, 3–7 September 2001, Vol. 2, pp. 1359–1362.

Stokes, Nicola (2003). Spoken and written news story segmentation using lexical chains. In *Proceedings of the Human Language Technolgy Conference of the North American Chapter of the Association for Computational Linguistics Student Research Workshop*, Edmonton, Alberta, Canada, 27 May –1 June, 2003, pp. 49–54.

Tür, Gökhan, Dilek Hakkani-Tür, Andreas Stolcke & Elizabeth Shriberg (2001). Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27(1):31–57.

Utiyama, Masao & Hitoshi Isahara (2001). A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, pp. 1–8.

Zechner, Klaus (2002). Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics*, 28(4):447–484.