Story Segmentation for Speech Transcripts in Sparse Data Conditions

Laurens van der Werff
University of Twente, HMI group
P.O. Box 217
7500AE Enschede, The Netherlands
laurensw75@gmail.com

ABSTRACT

Information Retrieval systems determine relevance by comparing information needs with the content of potential retrieval units. Unlike most textual data, automatically generated speech transcripts cannot by default be easily divided into obvious retrieval units due to a lack of explicit structural markers. This problem can be addressed by automatically detecting topically cohesive segments, or stories. However, when the content collection consists of speech from less formal domains than broadcast news, most of the standard automatic boundary detection methods are potentially unsuitable due to their reliance on learned features. In particular for conversational speech, the lack of adequate training data can present a significant issue. In this paper four methods for automatic segmentation of speech transcriptions are compared. These are selected because of their independence from collection specific knowledge and implemented without the use of training data. Two of the four methods are based on existing algorithms, the others are novel approaches based on a dynamic segmentation algorithm (QDSA) that incorporates information about the query, and WordNet. Experiments were done on a task similar to TREC SDR unknown boundaries condition. For the best performing system, QDSA, the retrieval scores for a $tfidf$-type ranking function were equivalent to a reference segmentation, and improved through document length normalization using the $bm25$/Okapi method. For the task of automatically segmenting speech transcripts for use in information retrieval, we conclude that a training-poor processing paradigm which can be crucial for handling surprise data is feasible.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing Methods*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval Models*

SSCS'10, October 29, 2010, Firenze, Italy.

General Terms

Algorithms, Experimentation, Performance

1. INTRODUCTION

Spoken Document Retrieval (SDR) is generally taken to mean matching a user information need as expressed in a textual query to the content of spoken documents. In many cases, this will require the application of Information Retrieval (IR) technology to Automatic Speech Recognition (ASR) transcripts. Almost all IR approaches rely on explicit boundaries between potential retrieval units, often referred to as stories. Speech recognition transcripts however, lack a suitable division into retrieval units. Thus, there is a need for an automatic approach to segmenting speech transcripts.

Initially the main challenge for SDR was expected to be the performance of the speech recognition system, but formal benchmarking in the late 1990's indicated that this may not be the case for Broadcast News (BN)-type tasks that follow the TREC[1] conventions. ASR performance has improved tremendously over the years, making SDR on certain non-BN types of collections more feasible than it may have been a decade ago. Transcription quality remains an important challenge when it comes to collections containing large amounts of spontaneous speech. For such collections, word error rates can easily exceed 30% potentially causing degradation of retrieval performance.

In the TREC IR benchmarks, the search task was carried out on a large collection of individual stories. A story was a cohesive segment of news that included two or more declarative clauses about a single event. A topic was a user need statement, which is a more explicitly defined version of a query. The basic task was to match topic descriptions (or queries) to individual stories. One of the fundamental assumptions in traditional TREC-style IR was that all textual content could be approached as a collection of stories and that a user need could be served by a subset of that collection, as defined by a topic or query. Because TREC-style IR is such a well researched subject, SDR has often been approached in a similar fashion [7].

Speech recognition errors are not the only challenge for SDR. Several approaches to ranking of retrieval results in IR have been attempted, including $bm25$/Okapi [15] and language models [11]. These and most other ranking algorithms have a reliance on explicit story boundaries between retrieval units. SDR, as a special case of information retrieval, also

[1] `http://trec.nist.gov`

needs such story boundaries for determining the relevance of speech fragments.

Spoken content, especially spontaneous speech, due to its very nature can be quite unstructured, making it difficult to determine where a conversational topic begins and ends. Besides this inherent lack of structure in its content, automatic speech transcripts do not contain any of the structural cues one would expect in textual content, such as chapters, paragraphs or sentences. Since current approaches to automatic speech recognition do not address this issue, it must be dealt with in a post-processing step.

Automatic segmentation of speech transcripts can be done either by using algorithms that can be applied directly, or with a system that relies on features that are (automatically) gathered from a set of labelled training data. The latter approach has proven most succesful in the TDT benchmarks [17], but requires matching training data for setting its parameters. For spoken content, suitable training data is generally from the BN domain, a type of speech that is quite similar to written content.

Many collections that are seen as candidates for SDR are rather dissimilar to BN. Examples are interview collections, non-news radio broadcasts, and historical audio collections. This makes it sometimes more difficult to define a story, but still requires a similar type of segmentation for IR ranking. Matching training data for automatic segmentation is quite rare for non-BN speech.

In such cases, segmentation must be performed without the use of collection specific training data. It is of course a matter of fact that most LVCSR systems rely heavily on trained language models. Although such unmatched models are beneficial for the purpose of automatic transcription, it is unclear whether they could also be used for story segmentation of unmatched collections. All segmentation methods in our experiments were implemented without using any automatically trained features.

We are interested in the feasibility of automatic segmentation of speech transcripts for the purpose of IR, without the use of trained segmentation models. This paper presents the results of using four different algorithms for this task. Section 2 briefly reviews earlier approaches to segmentation and presents two new methods, Section 3 provides the experimental set up, followed by Section 4 which contains the results of the experiments. Finally, Section 5 contains a brief discussion of the results and a general conclusion.

2. APPROACHES TO SEGMENTATION

This section gives a brief overview of earlier approaches to segmentation, both from an algorithmic and from an evaluational point of view. This is followed by a description of the segmentation methods used in our experiments.

A commonly used method of story segmentation which does not rely on (large amounts of) training data involves the use of lexical chains [14]. The general idea behind this approach is to create links between terms in a body of text, based on thesaurus relations [14], WordNet [16], or plain repetition (TextTiling [9]). These links can then be interpreted as 'chains' that bind certain stretches of words together. Story boundaries are expected to occur where a high concentration of chain begin and end points exists. Most of the work on lexical chaining was done before the 'official' benchmarking. Thus, no direct comparisons on the same collection and using the same metrics for optimized implementations of these segmentation methods are available.

The TDT benchmarks [17] gave the issue of story segmentation some additional legitimacy, mostly because a clear goal was stated and a single collection was used by all participants allowing for true comparisons of systems. The goal for these benchmarks was to minimize the value of the segmentation cost-function (Equation 1) on a collection of transcribed broadcast news speech. Because of the availability of matching training material, most of the systems used methods such as language models [2] and lexical cues [18] to tackle the issue.

In this paper, automatic story segmentation is approached as a subtask of SDR. We define accuracy of boundaries as the inherent quality of the automatic segmentation, i.e., how close are the hypothesized story boundaries to a manually generated reference. Effectiveness is defined as the usefulness of the boundaries for the task at hand, i.e., SDR. The experiments differ in two main ways from earlier work: (i) effectiveness rather than accuracy is measured, and (ii) because many spoken word collections are quite different in content and form from BN, no suitable material for training of statistically motivated systems is assumed to be available.

The accuracy of story boundary generation methods was defined for the segmentation task of the TDT2 evaluation as a cost-function [4], see Equation 1. C_{Miss}, $C_{FalseAlarm}$, and p_{seg} are constants with values of 10, 1, and 0.1 respectively for the TDT2 evaluation. Effectively the value of C_{seg} is therefore only dependent on p_{miss} and p_{fa}. These are calculated by moving a window over the document and determining the likelihood of such a window containing a 'miss', meaning a boundary is found where none is present in the reference, or a 'false alarm', meaning a boundary is present in the reference but not in the automatic transcript [2]. For the official TDT2 segmentation task evaluation, the size of the moving window was 15 seconds.

$$C_{seg} = C_{Miss} \times p_{miss} \times p_{seg} + C_{FalseAlarm} \times p_{fa} \times (1 - p_{seg}) \quad (1)$$

Effectiveness is calculated using Mean Average Precision (MAP) [13]. It measures the performance of a retrieval system, based on the relevance of the retrieval results and the order in which they are presented. Our retrieval experiments follow the evaluation methods of TREC8 and TREC9-SDR unknown story boundaries condition [7]. Thus it is only required to produce a single position (a time-code) as a retrieval unit in the audio collection. The evaluation scripts map this position to a known story and score as if this story was presented as a retrieval result. Exact boundaries are not produced in the ranked result list, effectiveness therefore also ignores all potential benefits for a user that may come from having an exact starting and ending point for their search results.

For many spoken document collections, no suitable training data is available. Methods that require trained models were therefore excluded from our experiments. Two existing methods and two novel methods were implemented for SDR. The existing methods are Equal-length segments, providing a baseline for untrained segmentation, and TextTiling which has been shown to work reasonably well on textual data. The two novel methods introduced here are the Query-based Dynamic Segmentation Algorithm (QDSA), in effect adapting the boundaries to the user need, and WordNet-based

boundaries which uses language (not collection) specific information for the task of segmentation.

The remainder of this section describes the existing techniques in overview and introduces our new methods.

Equal-length segments.

As a baseline system, the approach which was used by many systems for the TREC8 and TREC9 SDR benchmarks was implemented [1]. The incoming data is divided into segments of equal length, each optionally overlapping the previous. This method requires no knowledge of the language or the collection, and has the desired length of the segments and the amount of overlap as parameters.

TextTiling.

TextTiling [9, 10] uses a vocabulary shift as an indicator for topic shifts, and therefore of topic boundaries. It works by first determining the cohesion of text between either side of a potential boundary (called 'gap'). The best performing cohesion values are found by counting word repetitions. Then the cohesion for each gap is compared to that of surrounding gaps, resulting in a depth score. The gaps with the highest depth scores are selected as topic boundaries.

For our experiments, a Perl implementation of TextTiling from the CPAN library 'Lingua-EN-Segmenter-0.1' was used. This implementation uses some defaults for tunable parameters of TextTiling and requires paragraphs as initial segmentation units, i.e., gaps are hypothesized at paragraph breaks. The only parameter needed is the desired number of segments in the output.

Because speech transcripts do not contain an explicit paragraph structure, 'utterances' were used as such. Utterance breaks in our automatic speech transcripts were produced by the LVCSR system. They are generated fully automatically, mostly at silences in the audio signal. In practice, they often coincide with 'sentence' boundaries, but this is in no way guaranteed, as is borne out by our experiments.

Dynamic segmentation (QDSA).

Our proposal for a Query-based Dynamic Segmentation Algorithm (QDSA) does not impose a single set of boundaries, but rather optimizes them for the specific information need that is expressed in a query. As such, it does not divide the transcript into exhaustive and disjoint segments, but defines segments only for positions that match the information need. As users choose query terms because they expect them to occur together in relevant stories, the resultant 'stories' are tailored towards this information need.

It works by first determining all positions in a speech transcript that contain a query term. Then all of these positions which are within a certain minimum distance of each other are combined, including all terms that are positioned between them, forming the retrievable segments (or stories).

For the *bm*25 ranking function, each retrievable segment and each term in the query requires: (i) a token frequency (tf), (ii) a document frequency (df), and (iii) a document length (dl). In our implementation, this means that tf is the number of times the term occurs within this segment, df the number of segments containing the term, and dl the length of the segment, for which we use the number of terms between the term at the lowest and highest position in the segment (inclusively).

In contrast with the other methods, QDSA must be done during retrieval. The transcript cannot be presegmented, adding some processing time to the retrieval task, leading to a potentially less efficient system. The only parameter needed for this method is the minimum distance between individual segments.

WordNet-based segmentation.

WordNet [5] is a lexical database of English, but equivalents are available in many other languages including most official European languages. In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, each expressing a distinct concept. These sets are then hierarchically linked to express semantic and lexical relations.

It is possible to express the distance and relative position of two terms in the WordNet hierarchy as a similarity score. In the Perl CPAN module 'WordNet-Similarity' used in this work, several methods are available which compute a similarity score ranging between 0 and 1 for any input word-pair. The method of similarity we report on here (called 'jcn') is based on a sum of the *is-a* edges in the hierarchy between the two words that are compared, where each edge is given a weight based on its information content relative to its parent node [12].

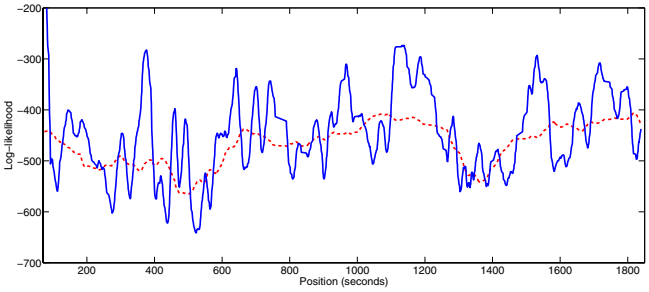

Figure 1: Example of MA-filtered versions of the log-likelihood curve. The solid line represents the short-term, the dashed line the long term average.

Our WordNet-based approach interprets these weights as a probability measure for the likelihood that the terms are part of the same story. If two terms have a higher similarity score, it is assumed that they are more likely to belong to the same story than when they have a low similariy score. All indexable terms were used, stopwords were ignored.

The likelihood of a set of terms being part of a single story was calculated using Equation 2. For each potential boundary position, the likelihood of it being a story boundary is determined by the likelihood of all terms in its left context (the extent of which is determined by n) being a single story, and its right context being a single story, and the likelihood that both left and right-side context *do not* form single story, see Equation 3.

$$\widehat{p}_{story}(1,..,M) = \prod_{l=1}^{M-1} \prod_{m=l+1}^{M} \widehat{p}_{sim}(l,m) \qquad (2)$$

$$\widehat{p}_{bound}(t) = \widehat{p}_{story}(t-n+1,..,t) \times \widehat{p}_{story}(t+1,..,t+n) \times \widehat{p}_{!story}(t-n+1,..,t+n) \qquad (3)$$

This gives a likelihood for a story boundary for each position in the transcript. The result is quite noisy, but can

be smoothened by using a moving average filter to calculate a short-term and a long-term average over these data points (Figure 1 was generated for one of our TDT2 audio files). Whenever the short-term average exceeded the long-term average, a boundary was hypothesized at the position where the distance between the two was highest. The number of generated boundaries was controlled by the sizes of the moving-average filters, which along with the context length are the parameters for this segmentation method.

3. EXPERIMENTAL SETUP

The goal of the experiments was to determine the effectiveness of four existing and novel story segmentation techniques, in the context of an SDR system. We decided that only differences that show up in retrieval performance as measured through MAP are relevant for our purposes.

Test Collection.

Arguably the most interesting collections for use in SDR are those that contain a substantial amount of spontaneous speech. However, to do retrieval experiments, one also needs queries and relevance judgements (qrels). TDT-2 is one of the few spoken document collections available that is suitable for retrieval experiments, but its content is exclusively made up of BN [3]. Nonetheless, since it also has clearly defined story boundaries, it was very suitable for the task at hand. It was also the collection used for TREC8 and TREC9 SDR. The provided manually generated TDT-2 story boundaries were used to perform a reference retrieval run, which was expected to result in the best possible retrieval for the collection given the IR configuration used.

The TDT-2 collection contains 902 documents (news programs) with 21,754 distinct stories stretching over 398 hours of English language speech. Contained in between these stories are 173 hours of audio which was not labelled as being part of a story, but often did contain speech. All speech transcripts that were used in our experiments were produced by the LIMSI Large Vocabulary Continuous Speech Recognition (LVCSR) system [8] in 2008, configured for fast processing rather than performance. The Word Error Rate (WER) for the set was below 20%. Our experiments used TREC8 and TREC9 SDR [7] query sets (henceforth referred to as trec8 and trec9), containing 50 topics each. The trec8 set had 1818 relevant documents for this collection, the trec9 set had 2216.

Evaluation.

The effectiveness of boundary generation algorithms in our experiments was determined through retrieval performance using MAP. By using two different ranking functions, $bm15$ and $bm25$, it is possible to distinguish between several properties of the generated boundaries.

The well-known $bm25$ function is often used in IR benchmarks and can be calculated using Equation 4, where tf is the count of a term in a story, df the count of stories containing the term, dl the story length, k_1 a parameter to weigh down additional matches of a term in a story (set at 1.1 for all our experiments) and b is used to tune the importance of story length normalization.

$$bm25_{t,d} = \frac{tf_{t,d}(k_1+1)}{k_1 \times ((1-b) + b \times \frac{dl_d}{dl_{avg}}) + tf_{t,d}} \times log\frac{N}{df_t} \quad (4)$$

If story length normalization is not used, by setting b to zero, the $bm25$ function reduces to $bm15$ (Equation 5). In this case, the ranking will only be affected by the query-term counts within the retrieved segments. The difference between the MAP scores of the $bm25$ and $bm15$ is indicative of the effectiveness of length normalization. The differences in retrieval performance between $bm15$ runs on the human reference segmentation and the automatic ones show the effectiveness of the segmenatation methods for determining tf and df.

$$bm15_{t,d} = \frac{tf_{t,d}(k_1+1)}{k_1 + tf_{t,d}} \times log\frac{N}{df_t} \quad (5)$$

MAP was calculated using the standard `trec_eval` version 9.0 program, after converting the retrieved positions into known documents using the `UIDmatch.pl` script. This was done by matching the retrieved time-code to the reference story that contains it.

Significance was determined using a paired sample sign test, since MAP does not necessarily behave according to a normal distribution. Differences were deemed significant when there was a less than 5% chance that they were produced by equally performing systems. For comparison purposes, p_{miss} and p_{fa} [2] were calculated to give an indication of the accuracy of the boundaries.

Automatic segmentation produced 572 hours of stories, the full TDT-2 English spoken content collection. However, the reference segmentation which was used to generate the relevance judgements only considered 398 hours. The difference is unlabelled speech and noise for which no reference segmentation was available. To make sure that our segmentation results are comparable with the reference system, the automatically generated stories are only included in the results if they have any overlap with manually labelled stories. Experiments (not shown) indicated that this filtering resulted in an improvement between 0 and 0.05 of MAP, depending on how the unlabelled sections were segmented.

The *Reference* run was done using manually created story boundaries. The *Reference/u* run used the same boundaries, but mapped them to the closest (LVCSR generated) utterance boundary. This gives the optimal segmentation taking utterances as the smallest available segmentation units. To compare against a system based on training data, the boundaries as generated by the IBM automatic segmentation system in 1999 [6] were also evaluated (*IBM automatic '99*). This system was trained using TDT-2 as its training corpus (it was developed for use on TDT-3), skewing the results in its favour as compared to a system trained independently.

4. RESULTS

This section gives results for our investigation of the alternative segmentation algorithms for our chosen SDR task. For reasons of space and clarity, only the results for the best performing parameter settings are presented in Table 1.

Boundaries placed at fixed intervals of 30 seconds (Equal-length) and with a 50% overlap (Equal-length/o) had p_{miss} and p_{fa} which are consistent with performance from randomly generated boundaries. MAP for Equal-length, Equal-length/o and Equal-length/u/o is significantly worse than for the reference segmentation. There was no gain to be had from document length normalization, as is demonstrated by the $bm25$ values which are not different from the $bm15$ results. The best performance was achieved by over-generating

		trec8		trec9			
	b	$bm25$	$bm15$	$bm25$	$bm15$	#stories	p_{miss}/p_{fa}
Reference	0.55	**0.4215**	0.3370	**0.3346**	0.2743	21754	
IBM automatic '99	0.55	**0.3031**	0.2413	**0.2507**	0.1973	32263	22.40/17.59
Equal-length (30s)	0.20	0.2960	0.2949	0.2503	0.2488	51986	50.9/50.1
Equal-length/o (30s)	0.15	0.2940	0.2941	0.2519	0.2482	103113	0/100
Dynamic - QDSA (30s)	0.10	0.3535	0.3413	**0.3016**	0.2754		
WordNet	0.80	0.2824	0.2735	0.2398	0.2413	31393	67.98/30.00
Reference/u	0.60	**0.3759**	0.3161	**0.2952**	0.2503	18199	35.2/5.7
Equal-length/u/o (60s)	0.65	**0.2610**	0.2334	**0.2127**	0.1997	55696	37.1/51.8
TextTiling (120s)	0.65	**0.2813**	0.2757	0.2119	0.2078	14288	78.6/8.2

Table 1: Retrieval and segmentation results on TDT-2, bold indicates when $bm25$ was significantly better than $bm15$. b indicates the weighting of story length in $bm25$ and was tuned on the trec8 topics.

segment boundaries (15, 45, 60, and 120 second segments were also tried but performed worse). Using overlapping segments did not improve performance.

Using segment boundaries that coincide with utterance boundaries by mapping all boundaries from Equal-length/o to the closest utterance boundary (Equal-length/u/o) resulted in a loss of performance, but did give $bm25$ a clear advantage over $bm15$. In addition, it resulted in better p_{miss} and p_{fa}, implying that the Equal-length/u/o boundaries were closer to the reference boundaries, despite performing worse from a retrieval point of view.

TextTiling performed best when the number of segments was consistent with an average segment length of 120 seconds (60 and 180 seconds were also tried). Miss (p_{miss}) and false-alarm (p_{fa}) rates were improved over Equal-length performance, although due to the low amount of stories generated, p_{miss} was quite high. Retrieval performance was better than Equal-length/u/o for trec8 queries, but not for the trec9 set, and always worse than Reference/u.

The QDSA approach to segmentation gave the best MAP when segments were separated by at least 30 seconds (values of 15, 45, and 60 seconds were also tried). Using this configuration, $bm15$ performance was equivalent to the reference runs. For the purpose of determining tf and df, QDSA resulted in reference level performance using the trec8 and trec9 topics on TDT-2. Adding story length normalization (dl) could only improve on the $bm15$ results for the trec9 queries (the improvement on trec8 was not significant), but fell short of the reference $bm25$ performance.

The best results for the WordNet approach were achieved by using a context of 20 terms (15, 25, and 30 terms were also tried) and moving average filters of 26 (short-term) and 100 terms (long term). In general it performed worse than both the Equal-length and QDSA methods. Miss (p_{miss}) and false-alarm (p_{fa}) rates were similar to what is expected from randomly placed boundaries. The $bm25$ ranking function was unable to bring significant improvement over $bm15$ for the WordNet-based method.

5. DISCUSSION

The experiments in this paper were designed to answer three questions: (i) is a cost-function a good indicator of the effectiveness of automatic segmentation for IR, (ii) how effective are some of the methods of automatic segmentation that can be implemented in a training-poor environment, and (iii) can untrained automatic segmentation for SDR result in acceptable retrieval performance?

The results in Table 1 show that Equal-length and WordNet-based boundaries have an accuracy (as measured using p_{miss} and p_{fa}) that is consistent with randomly placed boundaries. The boundaries that were based on utterances (Equal-length/u/o and TextTiling) show performance that is somewhat superior to random, but still lag behind the accuracy of the IBM trained system (*IBM automatic '98*).

There seems to be no correlation between effectiveness (as measured with MAP) and segmentation accuracy: the best MAP scores are achieved on systems with the poorest accuracy, while the most accurate system (*IBM automatic '98*) is unable to outperform Equal-length segments for the $bm25$ ranking function, and scores significantly worse for the $bm15$ ranking function. The conclusion is that accuracy is not a suitable optimization criterion when the goal of a segmentation algorithm is to provide term and document counts for use in IR.

Two existing methods of untrained automatic segmentation and two novel methods were implemented. They were compared with an automatic (trained) segmentation by an IBM system. Looking only at effectiveness for IR and the $bm25$ ranking function, the worst performing systems used utterances as initial segmentation units. Both Equal-length/u/o and TextTiling performance seemed to be compromised due to the performance ceiling that resulted from a non-ideal placement of utterance boundaries by the LVCSR system. At the same time, these systems were able to gain something from length normalization, but the low $bm15$ MAP performance makes these methods unsuitable for use on our TDT-2 transcription.

Utterance-based approaches could become more successful when utterances better coincide with sentence boundaries. This would also be beneficial to the language modeling side of an LVCSR system. Given that the speech recognition system we used was state of the art, making improvements in the utterance segmentation may not be easily achieved.

Our assumption that WordNet similarity scores can be interpreted directly as a likelihood for belonging to the same story has been falsified by the results of these experiments. Performance of the WordNet-based method was worse than for Equal-length boundaries and there was no advantage for length normalization. Both as a method of grouping terms and as a method for estimating story length, the WordNet-based approach was unsuccessful.

This could be because the similarity measure we chose

(called 'jcn' in the Perl library) was unsuitable, but several other methods we tried ('path' and 'lin') gave equally poor results. This does not mean that WordNet similarity cannot work for automatic story segmentation, but this would require a different or at least a more discriminating approach. For example, by using the similarity score only for certain types of words or by limiting the maximum distance in the WordNet tree between two terms.

The best performing automatic segmentation method was QDSA, which clearly outperformed Equal-length boundaries as well as the trained comparison system. For one set of queries, QDSA was able to improve from length normalization, but there was no improvement on the other set. A potential explanation for the good performance is that the average query length was a rather high 6.6 terms. Because QDSA uses co-occurrence of query terms as its main information source for the segmentation, it is expected to perform best on long(ish) queries.

QDSA has an important drawback: because stories are only defined after a query is formulated, indexing is somewhat more complex. This may not be too much of a problem for typical spoken word collections which are generally small enough for online processing. If QDSA is to be used on larger collections, it may be necessary to develop an indexing system which can deal with its requirements.

Effectiveness for IR does not take presentation issues into account. Having a high effectiveness does not automatically mean that the best user experience is obtained. This will be highly dependent on the interface, user requirements, and collection specifics. High accuracy helps in providing a user with the correct starting point, but high effectiveness helps in directing the user towards the most interesting sections.

All of our experiments were performed on broadcast news data, but did not use training material. The idea was that without training data, the performance of the segmentation system would be less dependent on specific properties of the collection. However, when such a collection contains spontaneous speech, it may be that segmentation becomes intrinsically more difficult. Something which is not reflected in the results of our experiments. To better understand the importance of this issue, different evaluation methods and collections may be needed.

Automatic segmentation of speech transcripts for IR without using training data does not compromise performance as compared to a trained system from IBM, at least on a broadcast news collection. The QDSA method performed as well on a retrieval task as the reference segmentation when the $bm15$ ranking function was used and outperformed all other methods (including a trained method) on the intrinsically superior $bm25$ ranking function.

6. ACKNOWLEDGEMENTS

The research reported on here was funded by the research project CHoral[2] , part of the NWO-CATCH [3] program. We would like to thank the LIMSI for kindly providing us with a full transcription of the TDT-2 speech corpus.

[2]http://hmi.ewi.utwente.nl/choral
[3]http://www.nwo.nl/catch

7. REFERENCES

[1] D. Abberley, S. Renals, D. Ellis, and T. Robinson. The THISL SDR system at trec-8. In *Proc. of the 8th Text Retrieval Conference TREC-8, Nov 1999. Martine Adda-Decker, Gilles Adda*, pages 699–706, 1999.

[2] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Mach. Learn.*, 34(1-3):177–210, 1999.

[3] C. C. David, D. Graff, M. Liberman, N. Martey, and S. Strassel. The TDT-2 text and speech corpus. In *Proc. of DARPA Broadcast News Workshop*, pages 57–60. Morgan Kaufmann, 1999.

[4] G. Doddington. The topic detection and tracking phase 2 (TDT2) evaluation plan. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[5] C. Fellbaum. *WordNet An Electronic Lexical Database*. The MIT Press, 1998.

[6] M. Franz, J. S. McCarley, T. Ward, and W.-J. Zhu. Segmentation and detection at IBM: Hybrid statistical models and two-tiered clustering. In *Proc. of TDT-3 Workshop*, 1999.

[7] J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees. The TREC spoken document retrieval task: A success story. In *Proc. of RIAO: Content Based Multimedia Information Access Conference*, Paris, France, 2000.

[8] J. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Communication*, 37:89–108, 2002.

[9] M. Hearst. Multi-paragraph segmentation of expository text. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 9–16, 1994.

[10] M. A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, 1997.

[11] D. Hiemstra. *Using Language Models for Information Retrieval*. Taaluitgeverij Neslia Paniculata, 2001.

[12] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of Int. Conf. Research on Computational Linguistics*, pages 19–33, 1997.

[13] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1998.

[14] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Compu. Ling.*, 17(1):21–48, March 1991.

[15] K. Sparck-Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. parts 1 and 2. *Information Processing and Management*, 36(6):779–840, 2000.

[16] N. Stokes, J. Carthy, and A. F. Smeaton. Segmenting broadcast news streams using lexical chains. In *Proc. of 1st Starting AI Researchers Symposium (STAIRS 2002*, pages 145–154, 2002.

[17] C. L. Wayne. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *Proc. of LREC 2000*, 2000.

[18] J. C. Yiming. CMU report on TDT-2: Segmentation, detection and tracking, 1999.