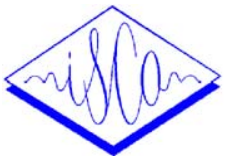
ISCA Archive
http://www.isca-speech.org/archive



6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

STORY SEGMENTATION AND TOPIC DETECTION FOR RECOGNIZED SPEECH

S. Dharanipragada *M. Franz* *J.S. McCarley* *S. Roukos* *T. Ward*

IBM T. J. Watson Research Center P. O. Box 218,
Yorktown Heights, NY 10598

10.21437/Eurospeech.1999-535

ABSTRACT

In this paper we present algorithms for story segmentation, topic detection, and topic tracking. The algorithms use a combination of machine learning, statistical natural language processing and information retrieval techniques. The story segmentation algorithm is a two stage algorithm that uses a decision tree based probabilistic model in the first stage and incorporates aspects of our topic detection system via an information-retrieval based refinement scheme in the second stage. The topic detection and tracking algorithm is an incremental clustering algorithm that employs a novel dynamic cluster-dependent similarity measure between documents and clusters. Performance of these algorithms are measured on the 1998 DARPA sponsored Topic Detection and Tracking Phase 2 (TDT2) evaluation task.

1. INTRODUCTION

In recent years we have seen a sudden explosion in the amount of audio and video content that is available online. Also, continuous speech recognition systems have come-of-age, making it possible to obtain effective transcripts of the audio cheaply. There is now need for algorithms that automatically organize multimedia content in a manner that facilitates applications such as browsing, searching and generating alerts. More specifically, we need algorithms that take raw text, that is often errorful, and automatically segment it into topics, detect new and old topics as they arise and track a specific topic if needed. In this paper, we present algorithms for story segmentation and topic detection. Both algorithms use a combination of machine learning, statistical natural language processing and information retrieval techniques.

The goal of a segmentation algorithm is to segment raw text, typically the output of an automatic speech recognizer (ASR), into its constituent stories. Our approach to story segmentation involves two stages. In the first stage we use a probabilistic model that computes the probability of a boundary at any word position given the text surrounding the position, $P(seg|text)$. This is similar to approach taken in [1], with the difference that we use a decision-tree based probabilistic model. The second stage refines the segmentation produced in the first stage via an information-retrieval based refinement scheme.

Given a collection of documents, with each document containing a specific topic, a detection algorithm imposes an organization on the collection such that the underlying topical structure in the collection is exposed. This is achieved by clustering the documents. For the algorithm to be online, clustering is done as soon as the document is seen i.e. without deferral. The topic detection algorithm presented in this paper is an incremental clustering algorithm similar to information-retrieval based clustering techniques described in [2, 3]. Essential to the clustering algorithm is a similarity measure between a document and a cluster, and a document normalization scheme that gives the measure a natural scale and prevents large documents from dominating the clusters. We present a novel dynamic cluster-dependent similarity measure between documents and clusters.

Topic tracking refers to the problem of finding topics that are similar to a given set of example topics. The tracking task is fundamentally similar to the detection task. Tracking is essentially a supervised clustering problem whereas detection is an unspervised clustering problem. In fact detection can be viewed as the concurrent tracking of all topics.

The segmentation, detection, and tracking algorithms are evaluated on the TDT2 corpus. A description of this corpus can be found in Linguistic Data Consortium (LDC) website [5]. The performance of these algorithms is measured in terms of costs C_{seg}, C_{det}, and C_{track} respectively which are defined by DARPA and NIST [6]. These metrics represent expected costs and are linear combinations of the probabilities of misses and false alarms.

2. STORY SEGMENTATION

2.1. System Outline

The segmentation system operates in two stages: the first stage hypothesizes boundaries, and the second stage removes boundaries. Nonspeech events play an important role in the processing: the ASR transcript has labeled nonspeech events (such as pauses, music, etc.) along with their duration. We regard this as a crude form of sentence detection, and perform part-of-speech tagging and morphological analysis on the "sentences" of recognized speech. This is similar to the document preprocessing that we have used successfully in our information retrieval system [7]. We then model the probability of segmentation at each "sentence" boundary.

The first stage of the segmentation system uses a binary decision tree based probabilistic model to compute the probability of a boundary at every point in the ASR transcript that has been labeled a non-speech event. The features proposed for the decision tree are extracted from finite windows to the left and right of the current point. The features used by the tree are selected automatically. To determine the document boundaries from the decision tree probabilities we first find the local probability maxima in an interval of several neighboring sentence boundaries. The interval peaks are compared with a threshold value to hypothesize document boundaries. We have also

incorporated a refinement stage, based on our detection metric, in which we remove posited boundaries between stories that are topically very similar.

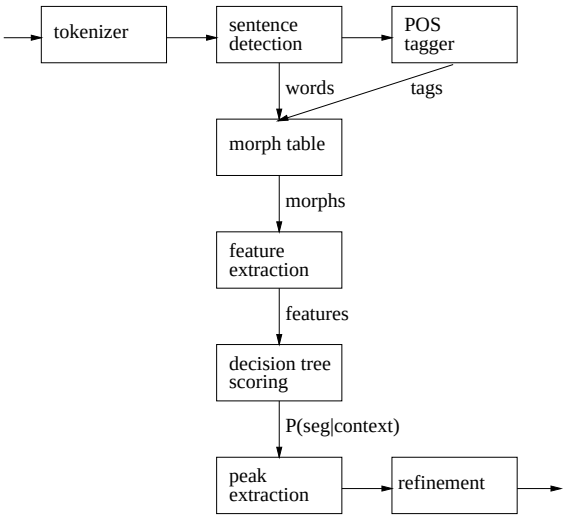


Figure 1: Components of the Segmentation System

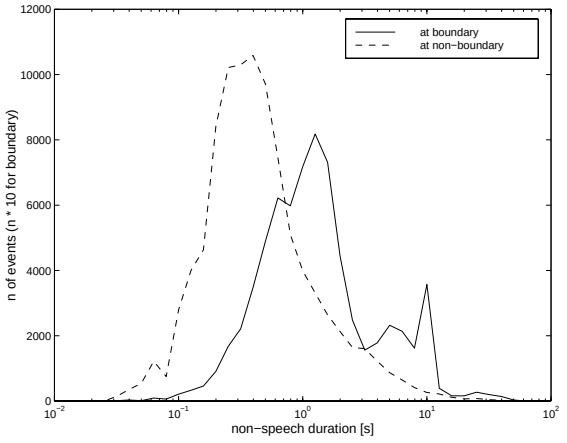


Figure 2: Duration of non-speech Events (note different scales for curves)

2.2. Decision Tree Features

There are three principal types of features. First, the single most important feature is the duration of events marked as non-speech in the ASR transcript. In many cases these events are silences, which tend to be longer between stories, as shown in Fig. 2. The next group of features is based on the presence of words and word pairs (bigrams) which are highly correlated with the document boundaries. These features, called *key* unigrams and bigrams are learned automatically from the training data based on a mutual information criterion. A related feature incorporates their average distance from the boundary. The final group of features is targeted to capture the degree of difference or similarity of the material in the window to the left and right of the current point. These features count and theshold the nouns appearing exclusively in the left and right window and in both of them - changes in subject matter are often accompanied by the introduction of many *new nouns* (i.e. nouns exclusively in the window to the right of the boundary.) A similar feature is based on left and right window semantic overlap, estimated using a symmetrized version of the Okapi formula. The top three layers of the decision tree, the feature questions, and the resulting probabilities of segmentation are shown in Figure 3.

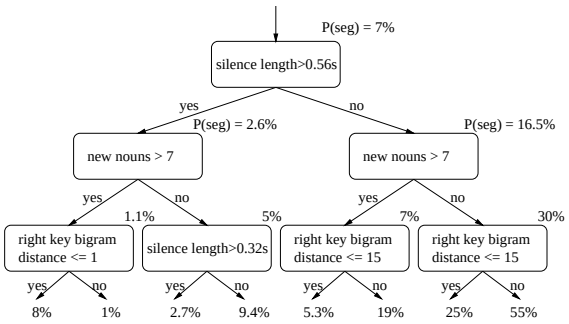


Figure 3: Top Layers of the Segmentation Decision Tree

2.3. Refinement

Our segmentation system is a two-stage process: after the story boundaries have been hypothesized, a second stage (within the deferral period) removes some of them in order to reduce the false-alarm rate. The second stage uses the document-document similarity score of our detection system (discussed below) to determine if adjacent stories are similar topically, and reject the hypothesized boundary between them. The refinement step is applied iteratively. The reduction in $P(fa)$ is of course offset by an increase $P(miss)$, but the result is a significant reduction in C_{seg}. Interestingly, the coupling between our segmentation and detection systems is more effective in the second stage than in the first stage: a similar document-similarity decision tree feature is not an important feature in the final decision trees.

3. TOPIC DETECTION

3.1. System Description

The topic detection algorithm is an incremental clustering algorithm. A document (which has been part-of-speech tagged and morphologically analyzed, as in segmentation) is assigned to a cluster as soon as it is seen i.e. without deferral. Essential to the clustering algorithm is a similarity measure between a document and a cluster. We measure the similarity between documents d^1 and d^2 using a symmetrized form of the Okapi formula:

$$\mathrm{Ok}\left(d^1, d^2\right) = \sum_{w \in d_1 \cap d_2} t_w^1 t_w^2 \mathrm{idf}(w, cl) \qquad (1)$$

where the term counts t_w^i of word w in document d^i have first been normalized for document length and then

warped by an $x/(a+x)$ form to prevent overweighting repeated words. The cluster-dependent inverse document frequency $\mathrm{idf}(w, cl)$ is initially estimated as the standard (cluster-independent) $\mathrm{idf}_0(w)$. We represent a cluster by its centroid, i.e., the term counts t_w^{cl} of a cluster are the mean warped term counts of its constituent documents

$$t_w^{cl} = \frac{1}{|cl|} \sum_{d \in cl} t_w^d \tag{2}$$

($|cl|$ is the number of documents belonging to cluster cl.) We note that the document-cluster score can be written as a mean of document-document scores.

We further allow the weightings of the words to vary both from cluster-to-cluster, and as the cluster evolves in time. Writing $\mathrm{idf}(w, cl) = \mathrm{idf}_0(w) + \Delta\mathrm{idf}(w, cl)$, we propose that $\Delta\mathrm{idf}(w, cl)$ should be a measure of the similarity of two sets of documents: $\mathcal{D}_w$, the set of documents that contain the word w, and the set of documents in cluster cl. In fact, we choose

$$\Delta\mathrm{idf}(w, cl) = \lambda \frac{2n_{w,cl}}{|\mathcal{D}_w||cl|} \tag{3}$$

where $n_{w,cl}$ is the number of documents in $\mathcal{D}_w \cap cl$, which can be interpreted as a harmonic mean of a "recall" and a "precision" (if $\mathcal{D}_w$ is interpreted as a set of relevant documents, and cl as a set of retrieved documents.) Note that $\Delta\mathrm{idf}(w, cl) = 0$ if and only if $\mathcal{D}_w \cap cl$ is empty and $\Delta\mathrm{idf}(w, cl) = \lambda$ if and only if $\mathcal{D}_w = cl$.

The clustering proceeds as follows: Each document d is compared with all existing clusters. Three options are available: The decision to *merge* a document with an existing cluster and update the cluster's statistics, *label* it as associated with an existing cluster, or seed a *new* cluster is accomplished by choosing the cluster cl^* that maximizes $\mathrm{Ok}(d, cl)$ and thresholding $\mathrm{Ok}(d, cl^*)$, with some exceptions. Generally, if $\mathrm{Ok}(d, cl^*) > \Theta_m$, we *merge*. If $\Theta_c \leq \mathrm{Ok}(d, cl^*) \leq \Theta_m$, we *label*. However, we only form a *new* cluster if $\mathrm{Ok}(d, cl^*) < \Theta_c$ and d contains more than 20 distinct words. We have noted that smaller documents are less stable as cluster seeds. Although our system allows Θ_m and Θ_c to be independent parameters, we have found no advantage to choosing $\Theta_m \neq \Theta_c$, except in the case of one- document clusters: then we have $\Theta_m > \Theta_c$, making it harder to merge a second document into a singleton cluster. Apparently, the C_{det} penalty for misses associated with singleton clusters is milder than the C_{det} penalty for false alarms associated with an impure initial cluster.

3.2. Tracking as Detection

In this section we exploit the fundamental similarity of the detection and tracking tasks. We view detection as the concurrent tracking of *all* topics and address the question of how much the cost of detection is lowered by concentrating resources and tracking only a few topics. In this view, the C_{track} score of a detection run is an upper bound on the score attainable by a tracking system. The prescription for using a detection system as a tracking system is as follows: after the last of the N_t training stories of the topic has been encountered, our system lists all clusters in which those N_t stories appeared. Then, any subsequent story in any of these clusters is marked as belonging to that topic. Note that even though a story can belong to only one detection system cluster, it may belong to more than one tracking topics if the training stories for more than one tracking topic appeared in the same cluster. In practice, this procedure amounts to simply reformatting the output of the detection program.

4. RESULTS

4.1. TDT Corpora

The TDT corpus consists of material drawn from six English news sources during the first six months of 1998. Four of the sources are broadcast news which has been transcribed by Dragon Systems' speech recognizer. The remaining two sources are text. For segmentation, story boundaries have been marked by the LDC, following close-captioning practices. For detection and tracking, 100 topics have been selected by the LDC, and all stories belonging to these topics have been labeled. The corpus has been divided into equal thirds for training, development-test, and evaluation. Further details are available at [5]. Only a small fraction of the stories belong to labeled topics.

4.2. Evaluation Measures

All of the TDT metrics are expected values of the cost of performing the task which separately weigh the costs of the two types of errors: misses and false alarms [6]. For segmentation, misses and false alarms are defined by whether the endpoints of a 50 word moving window are within the same story. For detection, the clusters reported to the system are aligned with the reference clusters to minimize C_{det}, and misses and false alarms are defined with respect to the clusters in this alignment. The tracking cost C_{track} is identical to C_{det} except that the alignment is provided by the system rather than determined during the scoring process.

4.3. Segmentation Results

The results of the segmentation system, using the standard DARPA metric, are summarized in Table 1. The first five lines report experiments on the development-test set. Our baseline system is as described above, without the second stage refinement. To increase the size of the available training data, we constructed additional training material by shuffling the original articles into a pseudo-random order (thus increasing the number of boundaries.) We also grew several decision trees of various depths and with differing splitting criteria and then mixed the results. Both of these yielded modest improvements. The largest single improvement was the second stage refinement (line 4) The next-to-the-last line shows the results of our submission system, incorporating all of the above experiments, on the development test set. The last line shows results of the system trained on the first two-thirds of the data and evaluated on the evaluation set.

4.4. Detection Results

The performance of our detection algorithm on the 1998 TDT training, development test, and evaluation sets in terms of the topic-weighted C_{det} is tabulated in Table 2 In addition to the submitted results, we also replaced the ASR transcriptions with the manually close-captioned transcriptions (man_ccap) to check the effect of speech recognition errors on our system. We observed a modest but consistent improvement in performance across all

	$P(miss)$	$P(fa)$	C_{seg}
[1] base	0.4614	0.0833	0.1967
[1] base + Monte Carlo	0.4327	0.0897	0.1926
[1] base + tree mixture	0.4688	0.0693	0.1892
[1] base + refinement	0.4166	0.0565	0.1645
[1] all, TDT2 dev. test	0.0676	0.3734	0.1593
[2] all, TDT2 eval.	0.0741	0.3776	0.1651

Table 1: Summary of segmentation experiments; [1] results on dev.-test [2] results on eval. set

	trn	dev	eval
asr+nwt	0.0050	0.0021	0.0042
man_ccap+nwt	0.0047	0.0019	0.0039

Table 2: Detection results on the TDT2 training, development, and evaluation sets

three data sets. Apparently our system is somewhat robust to speech recognition errors.

5. CONCLUSIONS

We have presented a novel approach to segmentation that couples a probabilistic model for hypothesizing segmentation with an information- retrieval approach to removing boundaries within topically homogenous material. We have also presented an information-retrieval based approach to document clustering that incorporates a novel dynamic, cluster-dependent measure of document-cluster similarity.

6. ACKNOWLEDGEMENTS

This work is supported by NIST grant no. 70NANB5H1174.

7. REFERENCES

[1] D. Beeferman, A.Berger, and J.Lafferty, "Text Segmentation Using Exponential Models" in *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing* ,p.35-46, 1997.

[2] J. Allan, R. Papka, and V.Lavrenko, "On-Line New Event Detection and Tracking" in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* p.37, 1998.

[3] Y.Yang, T.Pierce, and J.Carbonell, "A Study of Retrospective and On-Line Event Detection" in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* p.28, 1998.

[4] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic Detection and Tracking Pilot Study Final Report", in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop.* February, 1998.

[5] http://www.ldc.upenn.edu/TDT

[6] "The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan", Version 3.7, Aug. 3, 1998, http://www.nist.gov/speech/tdt98/tdt98.htm

	P(Miss)	P(Fa)	Ctrack
dev	0.1435	0.0005	0.0033
eval	0.2994	0.0025	0.0085

Table 3: Tracking as detection

[7] M. Franz, J.S. McCarley, and S. Roukos, "Ad hoc and multilingual information retrieval at IBM", in *The 7th Text REtrieval Conference (TREC-7)* ed. by E.M. Voorhees and D.K.Harman.