TNO at TDT2001: Language Model-Based Topic Detection

Martijn Spitters, Wessel Kraaij

TNO TPD
P. O. Box 155, 2600 AD Delft
The Netherlands

ABSTRACT

Topic detection is concerned with the unsupervised clustering of news stories over time. The TNO topic detection system is based on a language modeling approach. For the grouping of stories we combined a simple single pass method to establish an initial clustering and a reallocation method to stabilize the clusters within a certain allowed deferral period. The similarity of an incoming story S_n to an existing cluster C is defined as the average of the similarities of S_n to each story $S_i \in C$. These individual similarities are computed by taking the sum of the generative probabilities $P(S_n|S_i)$ and $P(S_i|S_n)$ where S_i and S_n are modeled as unigram language models. Because these story language models are based on extremely sparse statistics, the word probabilities are smoothed using a background model.

1 INTRODUCTION

This paper describes the design and development of a system for the unsupervised grouping of news stories according to the events they discuss. The system has been evaluated on an augmented version of the TDT3 corpus which contains approximately 80.000 stories from multiple news sources, including both text and speech. These sources are newswires, radio and television broadcasts, and Internet sites. The source languages are English and Mandarin. The TDT3 corpus is annotated for 120 topics, each of which spans both English and Mandarin sources. There are three alternative choices for the form of the audio sources to be processed, namely manual transcriptions, ASR transcriptions, or the sampled audio signal. Three story boundary conditions are supported: reference story boundaries (manually determined correct boundaries), automatic story boundaries (automatically determined errorful boundaries), or no story boundaries (the system must provide its own boundaries) [1].

The topic detection and tracking (TDT) benchmark evaluation project embraces a variety of technical challenges for information retrieval research. The goal of *story segmentation* is to segment a stream of data into homogeneous regions, discussing certain events. Given a small number of stories that discuss a certain event, a *tracking* system has the task to detect which stories in the data stream are related to this event and which are not. In *topic detection* there is no knowledge of the events to be detected. A detection system must both discover new events as the incoming stories are processed and associate incoming stories with the event-based story clusters created so far. A task which is very similar to topic detection is *first-story detection*. The goal of this task is to detect, in a chronologically ordered stream of stories, the first story that discusses a certain event. Finally, in *link detection*, the question to be answered is whether or not two stories discuss the same event. In this paper we report our work on topic detection.

The TNO topic detection system is based on a language modeling approach. We had good experience with the application of language models for different IR-related tasks, like ad hoc retrieval (including cross language and spoken document retrieval) [3, 4, 5], filtering [2], and multi-document summarization [6]. We also successfully applied language models for the topic tracking task at the TDT2000 evaluation [7]. However, due to the substantially higher computational complexity of the topic detection task, it was not trivial to convert our tracking approach into a detection algorithm. In the topic tracking task, events are to be followed individually. Each target event is defined by a small set of training stories that discuss it. Our tracking system estimates a single unigram language model based on the union of these on-topic stories and computes for each incoming story the likelihood according to this topic model. The computational complexity of this process is linear to the input. However, the topic detection task is a highly dynamic process. The topic models are constructed on the fly from the incoming stories. Each incoming story is added to a cluster, and thus changes the corresponding topic model. Experiments showed that reclustering the already processed stories (within the allowed deferral window) is essential for a good performance. Reclustering is a computationally demanding process, since every change in cluster membership lists is reflected in changes in the cluster models, which form the basis for the similarity computation. Therefore we have chosen for a clustering approach which is independent of the (global) cluster models and instead is based on the similarities between individual stories. The advantage of this approach is that the inter-story similarities can be cached, resulting in a significant speed-up of the clustering process.

The remainder of this paper is organized as follows. For readers who are not familiar with the TDT evaluation metric, section 2 describes how detection performance is characterized. In section 3 we describe in detail our language model-based approach to topic detection. This section also contains a short study into the influence of two different smoothing methods for language models on the detection performance of our system. Section 4 concludes with our plans for future work.

2 TDT EVALUATION METHOD

Topic detection systems are evaluated in terms of their ability to cluster together stories that discuss the same event (or events and activities that are directly connected to the cluster's seminal event). Detection performance is characterized in terms of the probability of miss and false alarm errors (P_{Miss} and P_{FA}). To speak in terms of the more established and well-known precision and recall measures: a low P_{Miss} corresponds to high recall, while a low P_{FA} corresponds to high precision.

These two error probabilities are combined into a single detection cost C_{Det}, by assigning costs to miss and false alarm errors [1]:

$$C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot P_{\neg target} \quad (1)$$

where C_{Miss} and C_{FA} are the costs of a miss and a false alarm respectively; P_{Miss} and P_{FA} are the conditional probabilities of a miss and a false alarm respectively; P_{target} and $P_{\neg target}$ are the a priori target probabilities ($P_{\neg target} = 1 - P_{target}$).

Then C_{Det} is normalized to:

$$(C_{Det})_{Norm} = \frac{C_{Det}}{min(C_{Miss} \cdot P_{target}, C_{FA} \cdot P_{\neg target})} \quad (2)$$

Detection error probability is estimated by accumulating errors seperately for each topic and by taking the average of the error probabilities over topics, with equal weight assigned to each topic. A set of predefined topics is automatically mapped to the system output topics by choosing for each reference topic the system output topic which produces the lowest evaluation cost.

3 DESIGN OF A PROBABILISTIC TOPIC DETECTION SYSTEM

This section describes in detail the design of the TNO topic detection system. 3.1 describes our clustering approach. We combined a simple single pass method to establish an initial clustering and a reallocation method to stabilize the clusters within a certain allowed deferral period. In 3.2 we describe our story-cluster similarity measure. An incoming story is compared to an existing cluster by averaging the similarities of the new story S_n to each story in the cluster S_i. These individual similarities are defined as the sum of the generative probabilities $P(S_n|S_i)$ and $P(S_i|S_n)$ where S_i and S_n are modeled as unigram language models. Because these story language models are based on extremely sparse statistics, the word probabilities are smoothed using a background model. Section 3.3 reports on our experiments concerning the application of two different smoothing methods for language models and some contrastive tests with automatic versus manually determined story boundaries.

3.1 Clustering Method

Our clustering procedure combines a simple single pass method and a reallocation method. Because the clusters formed by the single pass method are dependent of the order in which the stories are processed, they are merely used to initiate reallocation clustering. However, because a topic detection system may defer its assignment of stories until a limited amount of subsequent source data (10 source files) is processed, the reallocation is restricted to the stories within that deferral period. More specifically, our clustering process involves the following steps:

1. For each new story within the deferral window, compute its similarity to each cluster the system has created so far. There are two options for a story:
   (a) if the similarity of the story to the closest cluster exceeds a certain threshold, assign the story to that cluster
   (b) else create a new cluster with the concerning story as its seed
2. When the end of the deferral window is reached, loop through the window stories again and compare each story to each existing cluster. There are three options for a story:
   (a) a story may switch to another cluster if the similarity to that cluster exceeds both the similarity to its current cluster and the threshold
   (b) if neither the similarity to its current cluster nor the similarity to any other cluster exceeds the threshold, create a new cluster with the concerning story as its seed
   (c) if the similarity to its current cluster exceeds the threshold as well as the similarities to all other clusters, the story stays in its current cluster

The combination of a cluster initialization step and a reallocation step has previously (successfully) been used for topic detection by a.o. BBN [8] and Dragon [9].

The reclustering step is essential for a good performance of the detection system. However, the fact that every change in a cluster membership list means that the cluster language model would have to be reestimated, makes it a computationally demanding process. Therefore we have chosen for an approach which does not use the global cluster language models (contrary to our topic tracking approach) but instead is based on the similarities between individual stories. The similarity of an incoming story S_n to an existing cluster C is defined as the average of the similarities of S_n to each story $S_i \in C$. The advantage of this approach is that the inter-story similarities can be cached, resulting in a significant acceleration of the clustering process. These inter-story similarities are computed using a two-way language modeling approach, which is discussed in detail in the following section.

A cluster which has not changed for an uninterrupted period of fifteen days is frozen, which means that it is no longer considered an 'active event'. The cluster is removed from the list of candidate clusters for new stories. This cluster evolution monitoring has two advantages. First of all it limits the computational complexity, because the number of clusters a story has to be compared with stays within certain bounds. Second, it can be argued that restricting the temporal extent of an event is beneficial for detection performance because it prevents different events with similar vocabulary (like different attacks or political elections) to be grouped together [10].

3.2 Language Model-Based Similarity

The basic idea behind the language modeling approach to information retrieval is to estimate a (usually unigram) language model for each document and to rank documents by the probability that the document model generated the query. Absolute probabilities are not important for ranking in the IR situation. For other applications, i.e. topic tracking and also topic detection, scores have to be comparable on an absolute scale. For tracking, we found that modeling similarity as a likelihood ratio and normalizing this likelihood ratio by the (test) story length was adequate [7]. This normalized likelihood ratio is presented in equation (3), where $\mathrm{LLR}_{Norm}(T_1, T_2, ..., T_n|S_k)$ denotes the normalized log likelihood ratio of a story consisting of the terms $T_1, ..T_n$ given the story S_k in comparison with background model $\mathcal{B}$.

$$\mathrm{LLR}_{Norm}(T_1, T_2, ..., T_n|S_k) = \frac{1}{n} \log \sum_{i=1}^{n} \frac{P(T_i|S_k)}{P(T_i|\mathcal{B})} \quad (3)$$

In our clustering approach, the similarity between two stories S_n and S_i is based on a combination of the probability that the language model representing S_n generated story S_i and the reverse: the probability that the language model representing S_i generated story S_n. This approach results in the symmetrical similarity measure, presented in the following equation:

$$Sim(S_n, S_i) = \mathrm{LLR}_{Norm}(S_n|S_i) + \mathrm{LLR}_{Norm}(S_i|S_n) \quad (4)$$

Because the language models are estimated based on very limited amounts of text (single stories), it is very important that the word probabilities are smoothed using some background model. We performed a short study into the influence of two different smoothing methods on the performance of our detection system: Bayesian smoothing using Dirichlet priors and Jelinek-Mercer smoothing. The details of these smoothing methods and the results of our experiments are described in the following section.

3.3 Smoothing

Recent experiments at CMU have shown that different smoothing methods have different characteristics [12]. For title ad hoc queries, Zhai and Lafferty found Dirichlet smoothing to be more effective than linear interpolation (Jelinek-Mercer smoothing). Both methods start from the idea that the probability estimate for unseen terms: $P_u(T_i|S_k)$ is modeled as a coefficient α_s times the background collection based estimate: $P_u(T_i|S_k) = \alpha_s \cdot P(T_i|\mathcal{B})$. A crucial difference between Dirichlet and Jelinek-Mercer smoothing is that the smoothing coefficient is dependent on the story length for Dirichlet, reflecting the fact that probability estimates are more reliable for longer stories. Formula (5) shows the weighting formula for Dirichlet smoothing, where $c(T_i|S_k)$ is the term frequency of term T_i in story S_k , $\sum_w c(T_i; S_k)$ is the length of story S_k and μ is a constant. The smoothing coefficient α_s is in this case $\frac{\mu}{\sum_w c(T_i;S_k)+\mu}$, whereas the smoothing coefficient is λ in the Jelinek-Mercer based model (formula (6)).

$$P(T_1, T_2, \cdots, T_n|S_k) = \prod_{i=1}^{n} \frac{c(T_i; S_k) + \mu P(T_i|\mathcal{B})}{\sum_w c(T_i; S_k) + \mu} \quad (5)$$

$$P(T_1, T_2, \cdots, T_n|S_k) = \prod_{i=1}^{n} \lambda P(T_i|\mathcal{B}) + (1-\lambda) P(T_i|S_k) \quad (6)$$

For our official TDT2001 detection run, we applied Dirichlet smoothing with $\mu = 2000$. Our hypothesis was that Dirichlet smoothing would lead to improved performance, since story lengths vary considerably in the TDT corpus, and Dirichlet performed better than Jelinek-Mercer smoothing on a small test corpus (one month of stories from the TDT2 corpus) using the automatic story boundaries and ASR transcriptions of the audio (the primary topic detection evaluation requires

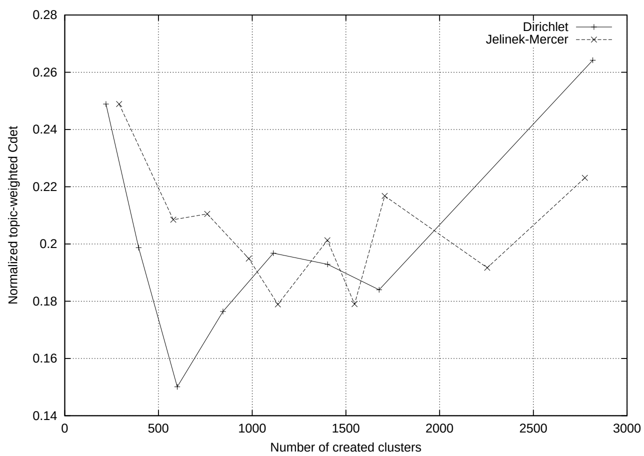

Figure 1: C_{Det} at different decision thresholds for two smoothing methods (Dirichlet and Jelinek-Mercer), performed on the TDT2 stories from April 1998, using automatic boundaries.

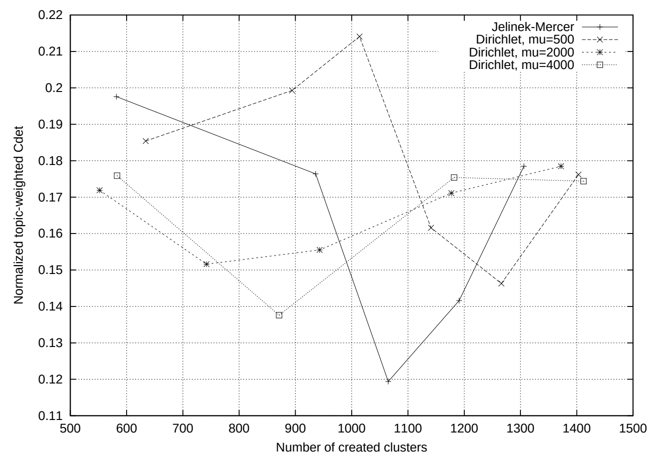

Figure 2: C_{Det} at different decision thresholds for two smoothing methods (Dirichlet and Jelinek-Mercer), performed on the TDT2 stories from April 1998, using reference boundaries.

these conditions). The results of this experiment are plotted in Figure (1).

We performed some post-hoc experiments on this same test set using reference story boundaries instead of automatic story boundaries and were surprised to find that Jelinek-Mercer performed better than Dirichlet under that condition, even when we varied μ (see equation (5)). Figure (2) shows the results. It is too early to draw conclusions from these experiments, since the test set was small and we did not explore the complete parameter space. However, one explanation could be the observation from Zhai and Lafferty [11, 12] that smoothing has two functions: i) improving the maximum likelihood estimates ii) generate common words in the query. The latter function is especially important for longer queries since they contain more common words.

In the topic detection task we use language models to generate stories instead of queries. Since stories are considerably longer than TREC title queries, it is probably important that the smoothed model generates common words with proper "idf"-like probabilities. The TREC experiments show that the two roles of smoothing have an inverse interaction with the query length. Dirichlet is a good strategy for the first smoothing role (avoiding the assignment of a zero probability to an unseen word) while Jelinek-Mercer is better for the second role (weighting query terms in an idf-like fashion) [12]. The longer the "queries" are, the more important the second function will become. This phenomenenon might be an explanation for the fact that Dirichlet performs best under the automatic story boundary condition, and Jelinek-Mercer under the reference story boundary condition, since the former has shorter stories than the latter (median: 62 versus 114). Further experiments are needed, including a validation of a combined Dirichlet/Jelinek-Mercer smoothing scheme for the TDT tasks.

4 CONCLUSIONS AND FUTURE WORK

We think that the choice to use normalized likelihood ratios as the basis of a similarity measure was the key for the good performance of our system. Like in the tracking task, a proper normalized similarity measure is of utmost importance. Simply adding the generative probabilities $P(S_n|S_i)$ and $P(S_i|S_n)$ proved to work well to "symmetrize" the similarity measure. The accuracy of a language model-based clustering approach which is independent of the (global) cluster models and instead is based on the similarities between individual stories surpassed our expectations. However, we intend to check whether a similarity measure based on the global cluster model would enhance the results. The results of some initial post-hoc experiments indicate that the Jelinek-Mercer smoothing method works better than Dirichlet smoothing for manually segmented data, while the Dirichlet method yields better performance than Jelinek-Mercer on automatically segmented data. Further investigation is necessary to draw definite conclusions.

5 ACKNOWLEDGEMENTS

The authors would like to thank Harry Wedemeijer for his intensive assistance with programming and debugging. We thank Hap Kolb and Stephan Raaijmakers for fruitful discussions and useful advice.

References

1. “The Year 2001 Topic Detection and Tracking (TDT2001) Task Definition and Evaluation Plan,” version 1.0, May 2001.
2. Ekkelenkamp, R., W. Kraaij and D. van Leeuwen, “TNO TREC-7 site report: SDR and Filtering,” *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*, 1999.
3. Hiemstra, D. and W. Kraaij, “Twenty-One at TREC-7: Ad Hoc and Cross Language track,” In: Voorhees, E. and D. Harman, Eds., *Proceedings of TREC-7*, pp. 227-238, 1999.
4. Hiemstra, D., W. Kraaij, R. Pohlmann and T. Westerveld, “Twenty-One at CLEF 2000: Translation Resources, Merging Strategies and Relevance Feedback,” *Proceedings of the CLEF 2000 Cross-Language Text Retrieval System Evaluation Campaign*, 2001.
5. Kraaij, W., R. Pohlmann and D. Hiemstra, “Twenty-One at TREC-8: using Language Technology for Information Retrieval,” *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, Gaithersburg, 2000.
6. Kraaij, W., M. Spitters and M. van der Heijden, “Combining a Mixture Language Model and Naive Bayes for Multi-Document Summarisation,” *Document Understanding Conference (DUC) 2001*, New Orleans, 2001.
7. Spitters, M. and W. Kraaij, “Using Language Models for Tracking Events of Interest over Time,” *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR) 2001*, pages 60-65, Pittsburgh, 2001.
8. Walls, F., H. Jin,S. Sista, and R. Schwartz, “Topic Detection in Broadcast News,” *Proceedings of the DARPA Broadcast News Workshop*, Herndon, 1999.
9. Yamron, J.P., S. Knecht, and P. van Mulbregt, “Dragon’s Tracking and Detection System for the TDT2000 Evaluation,” *Topic Detection and Tracking Workshop (TDT) 2000*, Gaithersburg, 2000.
10. Yang, Y., J. Carbonell, R. Brown, T. Pierce, B.T. Archibald, and X. Liu, ”Learning Approaches for Detecting and Tracking News Events,” *IEEE Intelligent Systems: Special Issue on Applications of Intelligent Information Retrieval*, Vol. 14(4), pp. 32-43, 1999.
11. Zhai, C. and J. Lafferty, “A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval,” *Prooceedings of SIGIR ’01*, New Orleans, 2001.
12. Zhai, C. and J. Lafferty, “Dual Role of Smoothing in the Language Modeling Approach,” *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR) 2001*, pp. 31-36, Pittsburgh, 2001.