A Language Modeling Approach to Tracking News Events

Martijn Spitters, Wessel Kraaij
{spitters,kraaij}@tpd.tno.nl

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

ABSTRACT

This paper presents the TNO tracking system for the 2000 Topic Detection and Tracking evaluation project (TDT2000). The objective of the TDT tracking task is to track events of interest over time. Being a first year participant to the TDT project, our original goal for this year was to build a baseline tracking system based on a language modeling approach. This approach had proved to be powerful for the TREC adaptive filtering task and several other IR tasks. Whereas the focus of adapative filtering lies on individual threshold adaptation based on feedback, tracking requires a uniform decision threshold. The assumption that test document-topic scores are approximately normally distributed turned out to be a key ingredient. The results on the tracking task of TDT2000 surpassed our expectations: our baseline system was ranked second for both the basic required and the alternate challenge conditions.

1. INTRODUCTION

Topic tracking and detection (TDT) is a relatively new challenge for information retrieval technology. The TDT benchmark evaluation project was initiated and supported by the U.S. Government since 1996. Main purpose of the TDT project is to advance the state of the art in identifying and following events being discussed in multiple news sources, including both text and speech. These sources are newswires, radio and television broadcasts, and WWW sources. The 1999 TDT project (TDT3) introduced multilinguality as a required test. The source languages are English and Mandarin.

The Linguistic Data Consortium (LDC) provides three corpora to support TDT2000 research: the TDT Pilot corpus, the TDT2 corpus, and the TDT3 corpus, extended with 60 additional topics. For TDT2000, sites may use both the TDT Pilot corpus and the TDT2 corpus for the development of their systems. The extended version of the TDT3 corpus, spanning news items from the period Oct-Dec 1998, was used for formal TDT2000 evaluation.

The TDT2000 project embraces five key technical challenges, namely topic segmentation, topic tracking, topic detection, first-story detection, and link detection. Tracking and link detection are considered to be the primary tasks, representing core technology that is broadly applicable to many different TDT applications [1]. In this paper we report our work on topic tracking.

Given a number of stories used to define a certain target topic N_t, a tracking system has the task to detect which stories in a chronologically-ordered stream of news stories are on-topic and which are not. As each test story is processed, the system should output a decision whether this story discusses the target topic, as well as a score that indicates how confident the system is about this decision. No look-ahead is allowed. The test stories for each topic will include all sources for both English and Mandarin. In tracking, there will be 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 by the evaluation of topic tracking performance: reference story boundaries (manually determined), automatic story boundaries (automatically determined), or no story boundaries. Sites that participate in the tracking task are required to perform at least one evaluation under common shared conditions. The TDT2000 basic required topic tracking conditions are English as the training language, both English and Mandarin as the test languages, 1 training story, manual transcriptions of the audio sources, and reference boundaries. The alternate (“challenge”) conditions are 4 training stories, ASR transcriptions of the audio sources, and automatic boundaries. TNO submitted two official evaluations for the topic tracking task: one that was performed under the basic required conditions and one that was performed under the “challenge” conditions.

This year is the first year TNO participates in the TDT project. Our original goals for this year were to get acquainted with the specifications of the TDT tasks and data structures, and to build a baseline tracking system that would at least produce average results compared to the systems of the other participating sites. We continued using a language modeling approach as proposed by Hiemstra [2], with which we had built experience and which has been extended by tests in the TREC adaptive filtering, Ad Hoc, SDR and CLIR tasks [10] [3] [6]. Whereas the focus of adapative filtering lies on individual threshold adaptation based on relevance feedback, tracking requires a uniform decision threshold. The assumption that test document-topic scores are approximately normally distributed turned out to be a key ingredient.

The TNO tracking system computes the probability that a certain test story is sampled from the word distribution of a certain topic, represented by one or more training stories. We computed the individual word probabilities based on a mixture model. The basic model is the unigram word distribution of the training story/stories (the topic model) interpolated with a background unigram language model estimated on the entire TDT2 corpus. A more detailed description of the TNO tracking system will be presented in the following section.

The remainder of this paper is organized into three main sections. Section 2 describes in detail the algorithm of our tracking system. In this section we will also throw light on the main notions of the *language modeling* approach to information retrieval, the theoretical foundation of our method. In section 3 we describe some experiments we conducted on the development data as well as their results. Section 4 presents in short our official TDT2000 results. Finally, we will conclude with some of our plans for next year’s TDT evaluation project.

2. A LANGUAGE MODELING APPROACH TO TOPIC TRACKING

In the past we have used a language modeling approach to IR and filtering [2] [6]. The idea in IR is to reformulate $P(D\text{=rel}|Q)$ by application of Bayes' rule:

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)} \tag{1}$$

In the Ad Hoc IR task, we want to rank documents D with respect to a fixed query Q. P(Q) is a constant for every document D, and can be left out. Thus a document ranking scheme simply consists of the product of $P(D|Q)$ and the a priori $P(D)$. One of the reasons to apply Bayes and compute $P(Q|D)$ instead of $P(D|Q)$ is because we know much more about a document than about the query.

The tracking situation is different though. Unlike the TREC Ad Hoc and filtering task, the situation is reversed: we generally have more knowledge about a topic than about a story. Thus, in this case we do not apply Bayes, and simply work with $P(S\text{=rel}|T)$ (where S represents a test story and T a topic). Moreover, the task is not to rank documents but to decide whether a document is on a topic or not. Such a classification task can be modeled by a probabilistic hypothesis test, the likelihood ratio:

$$LR(S|T) = \frac{P(S|T)}{P(S)} \tag{2}$$

In Formula (2) both numerator and denominator represent probabilistic hypotheses in the form of a likelihood. The basic idea of statistical language models is to model these likelihoods using simple unigram language models. Hypothesis H_{1i} represents the situation that a story is on a certain topic i . H_0 represents the hypothesis that a story is off-topic. Both hypotheses are likelihoods over a generative unigram model. If we assume independence over terms, we arrive at:

$$\frac{H_{1i}}{H_0} = \frac{P(S_1, S_2, \cdots, S_n|T_k)}{P(S_1, S_2, \cdots, S_n)} = \prod_{i=1}^{n} \frac{P(S|T)}{P(S)} \tag{3}$$

$S_1, .., S_n$ represents the sequence of n terms S_i that make up a test story and T_k a topic model for topic k.

If the likelihood ratio exceeds a certain threshold, the tracking system will reject H_0 and decide that the test document is relevant for that topic.

2.1. Orientation of the model

The model we propose to use is thus quite similar in spirit to the one we used for TREC Ad Hoc and filtering tasks. There are two differences though. Firstly, the tracking decision model is based on a likelihood ratio. This is essential for tracking, since stories differ and thus have a different a priori probability. The a priori $P(Q)$ in the TREC-8 filtering model (Equation (1)) is a constant and can therefore be ignored. Secondly, the orientation of the conditional probability is reversed. For TREC adaptive filtering, we scored documents (= stories) on $P(Q|D)$, for tracking we compute the reverse $P(S|T)$. This choice is motivated by the fact that in tracking, since we have several stories to train the topic, there is actually more data to describe the topic's 'aboutness' than the aboutness of the story, making it the most economic choice.

BBN has previously applied similar models for tracking, referring to the model based on computing the probability of a test story given a topic model as the **TS** model and the model where the probability of the topic is computed given a test story under the assumption that it is relevant as the **IR** model [9].

2.2. Building the Language Models

Because the topic language model is sparse, we apply linear interpolation with a background language model:

$$P(S_1, S_2, \cdots, S_n|T_k) = \prod_{i=1}^{n} (\lambda_i P(S_i|T_k) + (1 - \lambda_i)P(S_i)) \tag{4}$$

The probability of observing the sequence of terms which make up the test story is assumed to be equal to the product of the probability of observing the individual terms while sampling from topic model T_k (a simple unigram model), in other words we assume term independence. The probability of sampling a term S_i from topic model T_k is estimated on the set of training stories for T_k using a maximum likelihood estimator. This estimate is interpolated with the marginal $P(S_i)$ which is computed on a large background corpus (the entire TDT2 corpus). An intuitive interpretation of such a mixture model is a two state zeroth order Markov model, a Markov model without memory [9].

In the actual implementation we work with logarithms, converting to a log likelihood ratio:

$$LLR(S_1, S_2, \cdots, S_n|T_k) = \sum_{i=1}^{n} \log\left(\frac{\lambda_i P(S_i|T_k) + (1 - \lambda_i)P(S_i)}{P(S_i)}\right) \tag{5}$$

2.3. Normalizing Scores

The topic tracking task turned out to be quite different than the filtering tasks of TREC. For the adaptive filtering task of TREC we trained an individual threshold for every topic given incremental relevance feedback information [6]. This threshold was optimized for a certain utility function using simulated on-line relevance feedback information. The TDT tracking task is quite different. The task might look simpler because it does lack the threshold adaptation component, but is in some ways more difficult, because apart from a few sample stories which describe the event of interest, there is no on-line relevance information to refine the language model or threshold. This situation forces the experimenter to work with a uniform decision threshold, which requires that the story scores are comparable across topics. Unfortunately, usually similarity scores are not comparable across topics, and $LR(S|T)$ scores are no exception, because the individual language models have different probability characteristics. E.g. a topic can be described by a few very specific terms "Van den Hoogenband beated Thorpe in 200 m freestyle" or by much more common terms: "President Clinton visits China". This has the result that the probability of relevance distribution given a certain score computed for the set of test stories is quite different for each topic. This fact makes it impossible to use a single decision threshold for non-normalized topic models.

We tested two normalization techniques. The first method starts from the observation that scores are linearly depedent on the length of the test documents. An obvious normalization step is thus to divide the score by the length of the test document. Such an operation boils down to taking the geometric mean of the (initial) probabilities.

Another way to normalize the score distributions across topics, is to look at the score distributions themselves, disregarding the probability theory underlying the tracking model. Because we work with log probabilities, the tracking score can be seen as a sum of independent random discrete variables. The central limit theorem states that the resulting distribution can be approximated by a gaussian distribution if the number of addends is sufficiently large [9] [7] [8]. Figure (4) shows a histogram for the scores of 5000 randomly chosen test documents on topic 20023. Indeed this distribution is quite close to normal, though it lacks a left tail[1].

We have used this assumption to normalize the score distribution of each topic to a standard normal distribution. For each topic we calculated the scores of 5000 stories taken from the TDT Pilot corpus. We subsequently computed the mean and standard deviation of this set of scores. These distribution parameters were used to normalize the raw score τ in the following way:

$$\tau' = (\tau - \mu)/\sigma \tag{6}$$

3. EXPERIMENTS

Our baseline tracker was finished shortly before the official evaluation took place. Therefore, we only had time for a few basic experiments to tweek the system. Most of the experiments reported in this section were conducted after the official evaluation. We ran some tests with different values for λ_i (see Equation (5)). We performed contrastive tests to see whether stemming enhances the results. We also compared different methods for score normalization. The assumption that test document-topic scores are approximately normally distributed turned out to be a very important issue here. Finally, for runs where more than one training story is available, we conducted an experiment to contrast two different methods for merging the training stories. The experiments were conducted using the Jan-Apr part of the TDT2 corpus as training and development data, and the May-Jun part as the evaluation data.

3.1. Interpolation Parameter

In our experiments we took a constant for the interpolation parameter λ_i in Equations (4) and (5). A more refined model, based on term specific values for λ_i will be tested next year.

We performed several runs with different values for λ_i. Figure (1) shows that the DET-curves for the different values of λ_i are almost

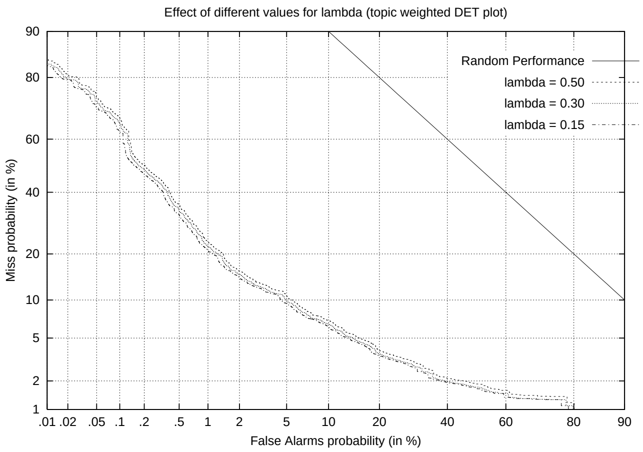

Figure 1: The effect of different values for λ on tracking performance on development data, using manual audio transcriptions, reference boundaries, and one on-topic training story.

similar. Using a λ_i of 0.15 produces the best results, but the improvement over the other tested values (0.30 and 0.50) is minimal. Earlier work by Hiemstra [6] has shown that learning term specific values for λ_i, which are sometimes called *relevance weights*, can improve performance because relevant terms can be boosted. A possible approach to relevance weighting is the expectation maximization algorithm. The algorithm iteratively maximizes the probability of the test story, given the training set.

3.2. Stemming

For the official tracking evaluations we stemmed the words from the English stories (using Porters stemmer) and used a stoplist to eliminate common words. However, in our earlier work on text categorization [4] we had to conclude that stemming significantly decreased precision[2]. Stemming can do harm to certain word-conjugations that are very typical for certain topics. Comparable conclusions have been drawn by Riloff who showed that stemming and removing common words can deteriorate the accuracy of text categorization [5]. Her experiments suggest that stopwords and stemming algorithms may remove or conflate many words that could be used to create more effective indexing terms. On the other hand, stemming usually increases recall, which is an important aspect of the official TDT evaluation methodology. To find out whether it was a good idea to stem the words, we performed two runs, one with stemming and one without stemming. Note that these runs were performed with both English and Mandarin as the test languages. We used the SYSTRAN translations of the Mandarin stories. Figure (2) shows that stemming slightly improves tracking performance.

3.3. Score Normalization

Figure (4) shows the effect of several normalization steps. To show the necessity of the likelihood ratio (i.e. normalizing by $P(S)$), we plotted the basic likelihood $P(S|T)$. This run is hardly better than a random system. Secondly, we plotted a plain likelihood ratio system, a system enhanced with story length normalization, a system with gaussian normalization, and one which inludes both story

[1] The gamma distribution probably would yield a better fit.

[2] These experiments were conducted on Dutch newspaper-articles, using an implementation of Porters stemmer for Dutch.

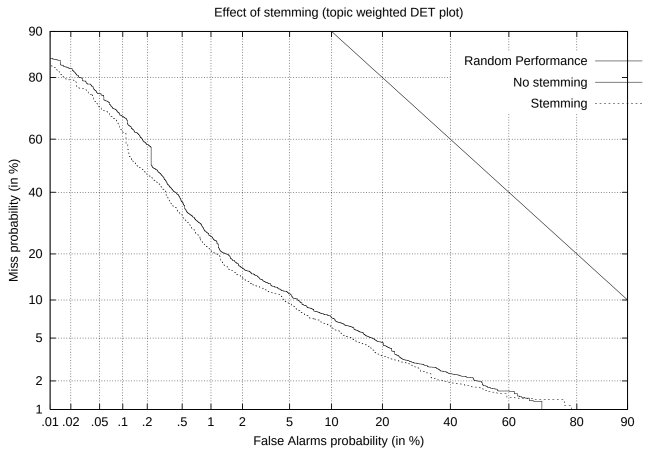

Figure 2: The effect of stemming on tracking performance on development data, using manual audio transcriptions, reference boundaries, and one on-topic training story.

length normalization and normalization to the standard normal distribution. The gaussian normalization proves especially effective on a score which has been normalized on test document length, in other words both normalization techniques add up.

3.4. Merging Training Stories

When more than one training story is available, we can compute the parameters of the topic model in several ways. We applied two different methods. The first approach simply concatenates all training stories and subsequently estimates the word probabilities over the concatenated text. This method takes no account of the individual story lengths. For example, in case one of the training stories is much longer than the other training stories, the probability estimates would be biased to the long story. We experimented with another merging strategy which is unbiased. This method first computes the word probabilities for the individual training stories, which are subsequently averaged over all the training stories. The experiment was performed under the "challenge" conditions (ASR transcriptions of

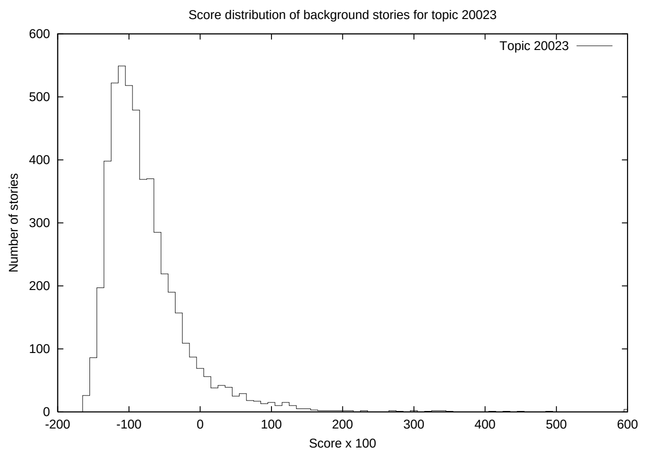

Figure 3: The score distribution of 5000 background stories (taken from the TDT Pilot corpus) for topic 20023.

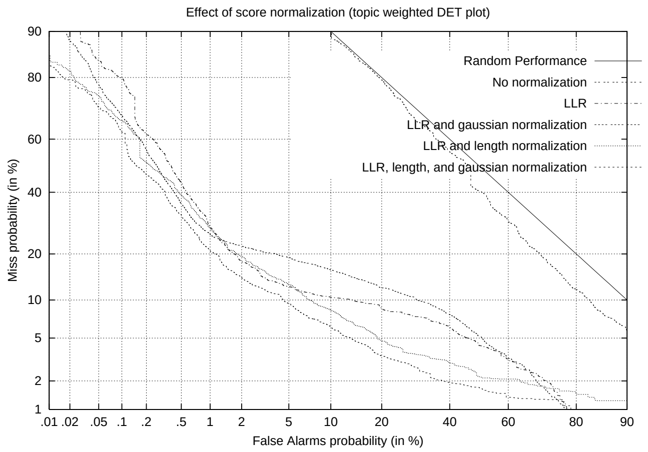

Figure 4: The effect of score normalization on tracking performance on development data, using manual audio transcriptions, reference boundaries, and one on-topic training story.

the audio sources, automatic boundaries, and four training stories). The DET-curves for both methods are almost similar, but when we take a closer look at the results for the individual topics, there are indeed some differences. The topics (from the May-Jun part of the TDT2 corpus) for which the results of the two merging strategies differed considerably, are presented in Table (1). The last column of the table contains the number of words in the four individual training stories. In the second row, "mrg1" represents the simple concatenation method, "mrg2" represents the unbiased story-merging approach. The results in Table (1) indicate that the unbiased merging method works better in case of large story length differences. However, because there were also topics with substantially differing story lengths, for which the results of the two merging methods were approximately the same, we can not draw definite conclusions from this experiment.

Topic	Pct. Miss		Pct. F/A		Story lengths
	mrg1	mrg2	mrg1	mrg2	
20005	.1702	.1489	.0082	.0083	314;360;157;1148
20076	.2273	.1970	.0167	.0159	564;58;358;504
20091	.0508	.0339	.0121	.0088	176;26;121;34
20096	.2113	.2113	.0300	.0079	658;242;192;223

Table 1: Results for different training story-merging strategies.

4. TDT2000 EVALUATION RESULTS

TNO submitted two runs for the official TDT2000 evaluation: one that was performed under the basic required conditions and one that was performed under the so-called "challenge" conditions. The DET-plots for these two evaluations are presented in Figure (5) (basic conditions) and Figure (6) (challenge conditions). With a normalized topic-weighted tracking cost of 0.1845, our system was ranked second for the evaluation performed under the basic conditions. For the evaluation performed under the "challenge" conditions, our system was also ranked second, with a normalized topic-weighted tracking cost of 0.1734.

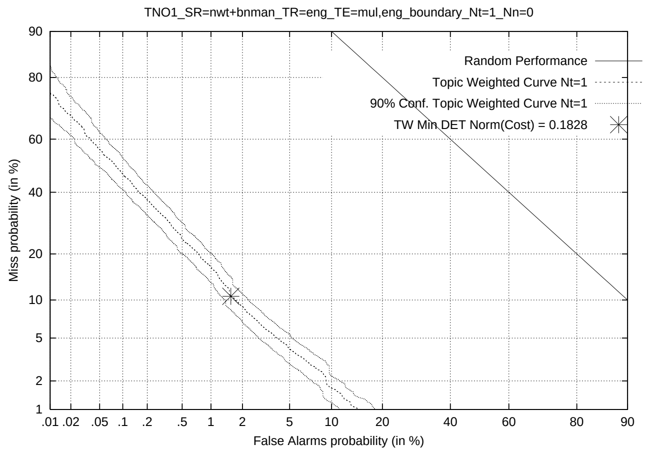

Figure 5: Official TDT2000 tracking results for TNO: manual audio transcriptions, reference boundaries, and one on-topic training story.

5. CONCLUSIONS AND FUTURE PLANS

We developed a baseline tracking system, based on language modeling. We had built experience with the application of language models for information retrieval in previous TREC adaptive filtering, Ad Hoc, SDR, and CLIR tasks. We conducted several experiments concerning a.o. stemming, score normalization, and training story-merging methods. One of the most important issues has been score normalization, which significantly improved the basic model. Although our original goal was to build a baseline tracking system that would at least produce average results compared to the systems of the other participating sites, our system performed very well and was ranked second in the TDT2000 evaluation.

There are several simple ideas to improve the tracking system. We could try to make different background models for the American and Chinese sources. Given the fact that the cost of a missed story is much higher than a false alarm, it could be beneficial to perform some kind of expansion (of stories or topics) using a contemporary large background corpus.

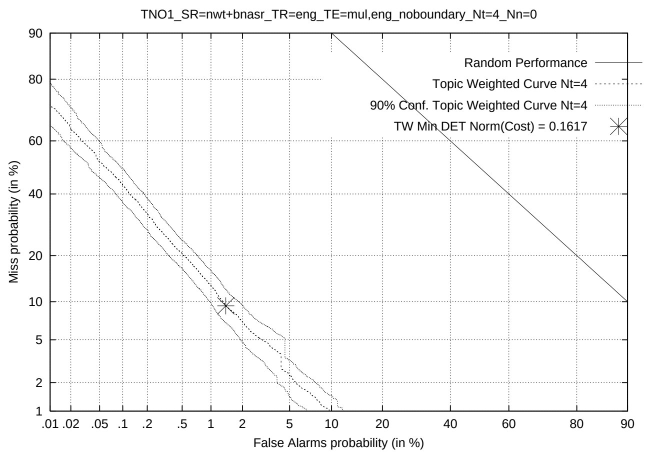

Figure 6: Official TDT2000 tracking results for TNO: ASR audio transcriptions, automatic boundaries, and four on-topic training stories.

As mentioned in section 3.1, for next year's evaluation, we will test a more refined model, based on learning optimal term specific values for λ_i (relevance weights), instead of taking a constant for the interpolation parameter.

Another issue is unsupervised adaptation. We want to use relevance feedback to enhance tracking accuracy by adding test stories that score exceptionally high to the topic language model.

6. ACKNOWLEDGEMENTS

We would like to thank Djoerd Hiemstra, Hans-Peter Kolb, and Stephan Raaijmakers for fruitful discussions and help with debugging.

References

1. "The Year 2000 Topic Detection and Tracking (TDT2000) Task Definition and Evaluation Plan," version 1. 4, August 2000.
2. Hiemstra, D., "A linguistically motivated probabilistic model of information retrieval," In: Christos, N. and C. Stephanides, Eds., *Proceedings of ECDL '98*, Springer Verlag, September 1998.
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. Spitters, M., "Comparing feature sets for learning text categorization," *Proceedings of RIAO 2000*, April, 2000.
5. Riloff, E., "Little words can make a big difference for text classification," *Proceedings of SIGIR '95*, pp. 130-137, July, 1995.
6. Kraaij, W., R. Pohlmann and D. Hiemstra, "Twenty-One at TREC-8: using Language Technology for Information Retrieval," *Proceedings of TREC-8*, 1999.
7. Baumgarten, C., "A probabilistic model for distributed information retrieval," *Proceedings of SIGIR '97*, pp. 258-266, 1997.
8. Baumgarten, C., "A probabilistic solution to the selection and fusion problem in distributed information retrieval," *Proceedings of SIGIR '99*, pp. 246-253, 1999.
9. Jin, H., R. Schwartz, S. Sista and F. Walls, "Topic Tracking for Radio, TV Broadcast and Newswire," *Proceedings of the DARPA Broadcast News Workshop*, Herndon, Va, 1999.
10. Ekkelenkamp, R., W. Kraaij and D. van Leeuwen, "TNO TREC-7 site report: SDR and Filtering," *Proceedings of TREC-7*, pp. 519-526, 1999.