TDT-2004: ADAPTIVE TOPIC TRACKING AT MARYLAND

Tamer Elsayed, Douglas W. Oard, David Doermann
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742
Contact author: telsayed@cs.umd.edu

Gary Kuhn
National Security Agency
9800 Savage Road, Suite 6514
Fort Meade, MD 20755-6514

ABSTRACT

A topic tracking system that combines elements from vector space and language modeling frameworks to compute document scores is described. The model is used for both the traditional TDT topic tracking evaluation design and the new supervised adaptive topic tracking evaluation. Results indicate that supervised adaptation and score normalization should be more closely coupled, and that current techniques for detection error tradeoff analysis may be of limited utility when supervised adaptation is performed.

1. INTRODUCTION

The adaptive topic tracking task at TDT-2004 offers a useful evaluation framework for a project that we have recently initiated in which online learning of user needs will be an important component. Our goals for participation in TDT-2004 therefore focused on infrastructure building; specifically: (1) to integrate our baseline ranking function into an adaptive topic tracking system, and (2) to explore the design of evaluation measures that are suitable to our ultimate tasks. Time constraints precluded parameter tuning using the TDT-4 collection, so the results reported below should be considered preliminary.

We submitted one non-adaptive topic tracking run for the required condition (one on-topic training story). At the time we submitted our non-adaptive run, score normalization was not yet incorporated into our system. As we show below, this adversely affected both our Detection Error Tradeoff (DET) curves and the resulting minimum detection error cost. Our actual detection error cost is below the computed minimum cost point on the DET curve because an implementation limitation at the time of submission resulted in a hard decision of "NO" for a few topics for which the training epoch was very near the beginning of the TDT-5 collection.

This is the first year in which adaptive topic tracking has been included in the Topic Detection and Tracking (TDT) evaluations. We submitted two runs, one with single-pass score normalization, and one in which score normalization was disabled (for contrast with our non-adaptive run). Both used only a single on-topic training story; topics with early training epochs were handled appropriately in these runs. Adaptation reduced the actual

detection error cost for the unnormalized condition from 0.65 to 0.24. Single-pass score normalization (with z-scores computed from the initial on-topic training story) increased the actual detection error cost from 0.24 to 0.38, suggesting a need for renormalization each time the topic representation is adjusted.

In prior topic tracking evaluations, we found the DET curves to be useful because our interests focused more on score computation than threshold selection [1,2]. Adaptive topic tracking introduces a natural dependence on threshold selection, and that dependence does not seem to be reflected well by the present way in which DET curves are constructed. We discuss this issue in greater detail below.

The remainder of this paper is organized as follows. In the next section, we describe the score computation function that was used for all three submitted runs and the implementation details for each run. Section 3 then presents our official results and some initial post-hoc analysis, including discussions of score normalization and DET curve interpretation for adaptive topic tracking. Finally, we conclude with a few remarks about future work.

2. MODELING TOPIC TRACKING

In this section we discuss the design of our tracking systems. We start with the language model in Section 2.1 followed by an overview of the system components in Section 2.2.

2.1. The Language Model

The "*n-gram* logodds" language model is a term frequency model, consisting of a lexicographically ordered list of character strings and log likelihood ratios. The strings in the list are exactly n characters long, hence the name *n-gram*, where the length n is a model parameter. These strings occurred in the on-topic and (assumed) off-topic training stories for the topic being tracked, T, and they may overlap. Any n-gram beginning with a character code value less than or equal to that of an ASCII space is not included in the model.

Other model parameters are the minimum number of occurrences of each N-gram, separately for the on-topic training set

(*MinCount$_{on}$*), the (assumed) off-topic training set *(MinCount$_{off}$)*, and overall (*MinCount$_{all}$*), *n*-grams that do not meet these minimum counts are excluded from the model.

The model is trained on two sets of stories. One set T_{on} contains on-topic stories, and the other set T_{off} contains (assumed) off-topic stories. In each set, the number of occurrences (term frequency) of each *n*-gram is counted. Those *n*-grams that meet the preset minimum counts are included in the model.

The relative frequency of each *n*-gram i_n is computed for the on-topic, the off-topic, and the overall sets:

$$p(i_n \mid T_{on}) = \frac{tf(i_n, T_{on})}{\sum_{j_n \in T_{on}} tf(j_n, T_{on})} \quad (1)$$

$$p(i_n \mid T_{off}) = \frac{tf(i_n, T_{off})}{\sum_{j_n \in T_{off}} tf(j_n, T_{off})} \quad (2)$$

$$p(i_n \mid T_{all}) = \frac{tf(i_n, T_{all})}{\sum_{j \in T_{all}} tf(j_n, T_{all})} \quad (3)$$

where *tf(i$_n$,X)* is the term frequency of i_n in the set *X*.

A weight, λ_{i_n}, is computed for each N-gram i$_n$ in the model, based on its overall count:

$$\lambda_{i_n} = \frac{2}{1 + e^{-k \cdot tf(i_n, T_{all})}} - 1 \quad (4)$$

The experimenter chooses *k* in this equation. As *k* is increased from 0, λ reaches 0.5 at smaller and smaller values of the overall count. λ tells us how much weight to give to the potentially very small (and therefore less reliable) class-dependent relative frequency for *n*-gram i_n , ($p(i_n \mid T_{on})$, and $p(i_n \mid T_{off})$) and the potentially very large (and therefore more reliable) overall relative frequency for *n*-gram i_n, $p(i_n \mid T_{all})$, in the following convex combinations $p(i_n|T_{on})_{sm}$ and $p(i_n|T_{off})_{sm}$, which define the smoothed "likelihood" of *n*-gram i$_n$ in both of the training sets. Note that these values need to be normalized after being smoothed.

$$p(i_n \mid T_{on})_{sm} = \lambda_{i_n} p(i_n \mid T_{on}) + (1 - \lambda_{i_n}) p(i_n \mid T_{all}) \quad (5)$$

$$p(i_n \mid T_{off})_{sm} = \lambda_{i_n} p(i_n \mid T_{off}) + (1 - \lambda_{i_n}) p(i_n \mid T_{all}) \quad (6)$$

$$p(i_n \mid T_{on})_{norm} = \frac{p(i_n \mid T_{on})_{sm}}{\sum_{j_n \in T_{on}} p(j_n \mid T_{on})_{sm}} \quad (7)$$

$$p(i_n \mid T_{off})_{norm} = \frac{p(i_n \mid T_{off})_{sm}}{\sum_{j_n \in T_{off}} p(j_n \mid T_{off})_{sm}} \quad (8)$$

Finally, the "logodds" or log likelihood ratio for each *n*-gram is computed from the class-dependent likelihoods:

$$\log odds(i_n, T) = \log \frac{p(i_n \mid T_{on})_{norm}}{p(i_n \mid T_{off})_{norm}} \quad (9)$$

The score of story *S* for topic *T* was then computed as follows:

$$tf\_score(i_n, S) = \frac{tf(i_n, S)}{1.5 \frac{L_S}{L_{avg}} + tf(i_n, S) + 0.5} \quad (10)$$

$$score(S, T) = \frac{1}{n_s} \sum_{i_n \in S} \log odds(i_n, T) * tf\_score(i_n, S) \quad (11)$$

where i_n is an *n*-gram occurred in S, n_S is the number of different *n*-grams in S, L_S is the total number of *n*-grams in S, and L_{avg} is the average length of all stories seen so far (in both training and testing sets). In an earlier version of this model, we used the raw *tf* value; for our TDT-2004 experiments we replaced that with the Okapi BM25 term frequency component in equation (10) [3].
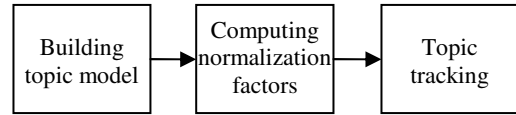
2.2. System Design

Figure 1: System overview.

As illustrated in Figure 1, the core system consists of three main modules: building the topic model using the technique described in Section 2.1, computing the factors that will be used in the context of cross-topic and cross-source score normalization, and finally tracking the on-topic stories in the incoming data stream.
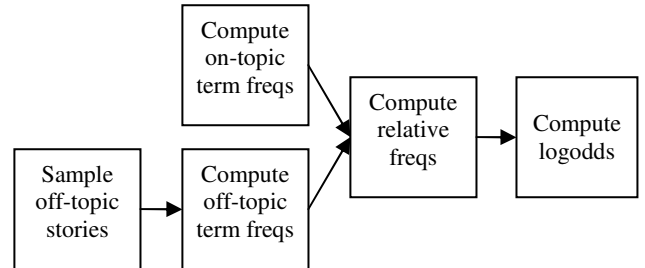
Building the Topic Model

Figure 2: Building the topic model.

A two-sided training set is needed to build our model for each topic. The required condition provides one on-topic training story and no confirmed off-topic stories. We assumed that stories that substantially predate the on-topic training story will be off-topic. To minimize the possibility of falsely selecting a story was actually on-topic, we formed a training epoch from the beginning of the TDT-5 corpus to the known on-topic story and then

randomly sampled N_{off} stories (if available) from the first 80% of that training epoch [1]. A cosine similarity measure was then computed between each of the N_{off} stories and the one on-topic training story. The top half of the most similar stories constituted the off-topic training set for that topic. This procedure was designed to identify off-topic stories that were fairly similar to the one on-topic story in the way in which they used terms. The logodds value for each term is then computed as described above. The whole process is depicted in Figure 2.
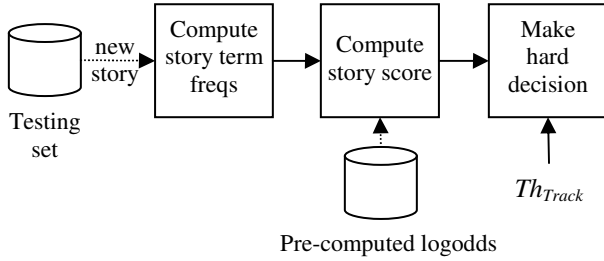
Non-adaptive Topic Tracking

Figure 3: The non-adaptive tracking approach.

The approach we adopted for the non-adaptive tracking task simply treats each story independently and does not take the story timing attributes into account. In other words, once the topic model has been trained as described above, each story in the incoming stream is scored with respect to the initial model (represented by the pre-computed logodds) as shown in Figure 3. Any given story would therefore receive the same score regardless of its order in the stream.

A hard (yes-no) decision for each story was made using a static threshold Th_{Track}. Because the non-adaptive system did not perform any unsupervised updates to the topic model, the computed score for each story does not depend on the specific value of Th_{Track}.

Supervised Adaptation for Topic Tracking

The new (supervised) adaptive tracking task is similar to the non-adaptive task, except that the true state (on-topic or off-topic) of a story S with respect to the current topic becomes known to the system immediately if the system makes a hard decision that the story is on-topic. This is intended to simulate an interactive application in which the user provides feedback for stories that the system elects to display.

We adopted a straightforward adaptation approach (illustrated in Figure 4) in which the new information was leveraged to enhance the current topic model. A story judged to be on-topic was added to the on-topic training set by merging its n-gram counts with the current counts in the topic model and then re-computing the

---

[1] In some cases, this resulted in fewer than N_{off} stories in the training epoch. For our adaptive runs, we added the entire TDT-4 corpus to the training epoch in order to avoid that problem.

relative frequencies and the logodds values for each term. A story judged to be off-topic was handled similarly; since our initial model was built using "assumed" off-topic stories (as described above), we removed one of the assumed off-topic stories each time a newly judged off-topic story became available and again retrained the logodds. In either case, the modified model would be used until the next time the system elected to declare an on-topic story (at which time a new judgment would become available). We again used a static threshold, Th_{Track}, to make these hard decisions for each story.
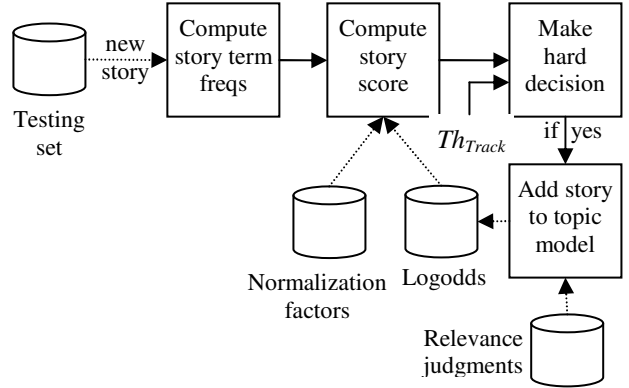

Figure 4: The adaptive tracking approach.

Computing Normalization Factors

In our second adaptive tracking run, we augmented our supervised topic tracking model by incorporating score normalization. In order to get comparable scores across topics, languages and news sources, we adopted a variant of the z-score normalization method introduced in [4]. The method assumes that the true scores of off-topic stories would follow a common Gaussian distribution regardless of condition; therefore, a score can be normalized given estimates for the mean and standard deviation of off-topic stories. We assumed for this purpose that most TDT-4 stories would be off topic for any TDT-5 topics. For each news source that existed in both TDT-4 and TDT-5, we sampled N_{norm} stories uniformly distributed across all data files for that source and computed scores for each story using each topic model separately; that resulted in initial estimates of μ_{off} and σ_{off} for each topic. In order to guard against the possibility that a few on-topic stories might exist in the TDT-4 corpus, we then removed any sampled story with a score greater than μ_{off} by more than $Th_{Norm}*\sigma_{off}$ and then recomputed final estimates for μ_{off} and σ_{off}. Those estimates (we call them normalization factors) are then used to normalize the raw scores as described below. For those cases in which TDT-5 contained a source that was not present in TDT-4, we averaged the normalization factors for all TDT-4 sources in the same source language and used those values as an estimate for what we would have obtained had training data for the new source been available.

Finally, the normalized score for each story was computed as follows:

$$score_{norm}(S,T) = \frac{score(S,T) - \mu_{off}(src(S),T)}{\sigma_{off}(src(S),T)} \quad \textbf{(12)}$$

where $src(S)$ is the source of the story S, and $score(S,T)$ is computed as in the non-adaptive approach. Note that the normalization factors were computed only once before the tracking phase and used throughout the testing process.

Implementation and Parameters Settings

Context	Parameter	Value	Parameter	Value
Model	N	6	K	0.37
	$MinCount_{on}$	0	$MinCount_{off}$	0
	$MinCount_{all}$	1		
Normalization	Th_{norm}	5	N_{norm}	2000
Training	N_{off}	100		

Table 1. System parameters

Our systems were all implemented in Java2. We didn't make use of any previously developed information retrieval system components, so the infrastructure has been built from scratch.

Table 1 summarizes the specific values of our systems parameters. All these values were statically used in all submitted runs.

3. RESULTS

This year, we submitted 3 systems, one for the non-adaptive tracking task, and two for the adaptive supervised task. Here we discuss the performance of these systems.

Task	System	Th_{Track}	Actual Cost	Min DET Cost
Non-adaptive	UMD1	0.0	0.6527	0.6733
Adaptive	UMD1	0.0	0.2438	0.2441
	UMD2	2.0	0.3789	0.3342

Table2. Actual and minimum DET costs for the submitted runs

3.1 Non-adaptive System

At the time of the non-adaptive task submission, the normalization phase was not read so the cost shown in Table 2 and the DET curve illustrated in Figure 5 are achieved in the absence of score normalization.

The real cost is less than the computed cost because of the hard "No" decision taken for 4 topics whose unique on-topic story was located at the very beginning of the TDT-5 corpus. This problem is solved in the adaptive runs.

Surprisingly, our actual cost was less than the "minimum" DET cost. The reason is that our system provided no score for the stories whose length is one word of less than 6 characters. Unfortunately, the evaluation set includes more than a hundred stories (on average) per topic. Despite being judged off-topic, those stories caused that
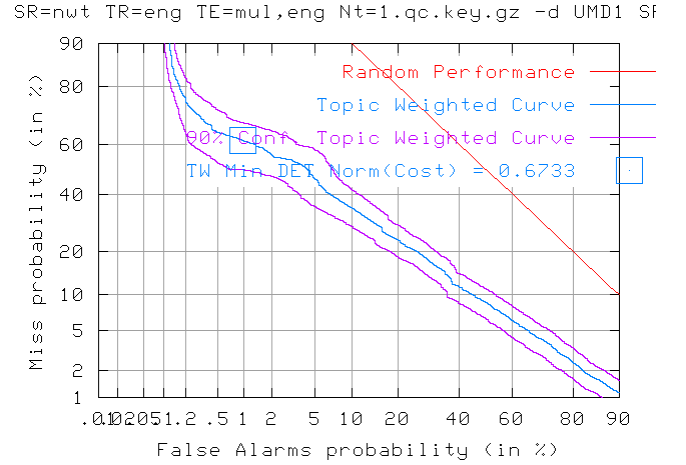

Figure 5. DET curve for the non-adaptive system.

abnormal result.

Our system built a static model trained with only one on-topic story, so we believe that that unique story might not be sufficient for our approach to build a reliable topic model. Had more time been available, we expect that a parameter tuning using the TDT-4 collection, would have yielded a stronger baseline. Score normalization is also expected to have a major effect.

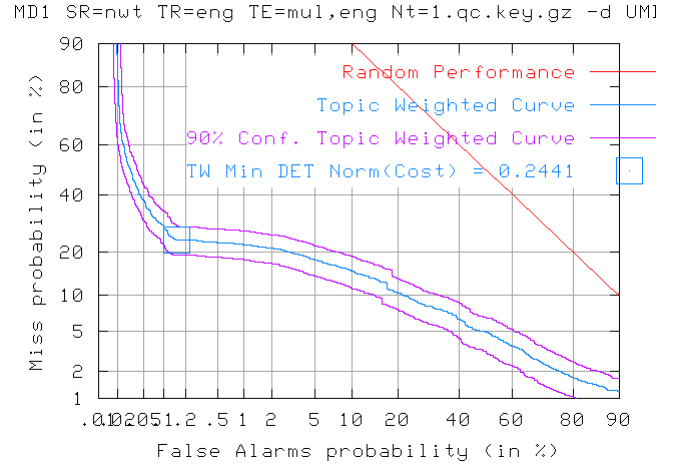## 3.2 Supervised Adaptation Systems

Figure 6. DET curve for the non-normalized adaptive system.

Figure 6 illustrates the performance of non-normalized supervised adaptive system. The actual cost dropped to 0.24 as shown in Table 2, about a 63% improvement over the non-adaptive cost. We attribute this to the ability of our system to learn evenly from both on-topic and off-topic stories. This result demonstrated the strength of our adaptation technique regardless of score
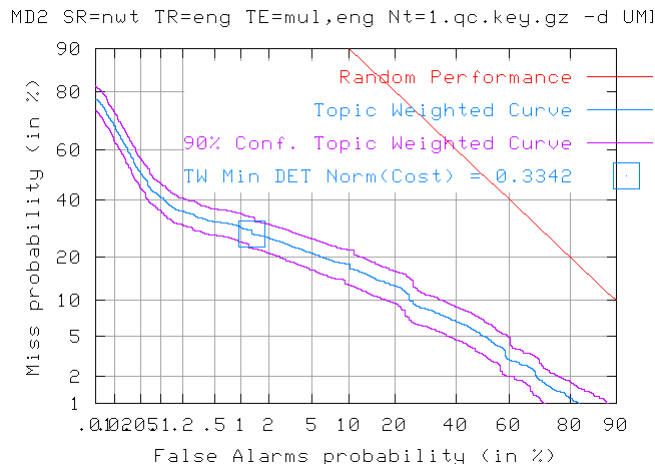
normalization.

Figure 7. DET curve for the normalized adaptive system.

The DET curve shown in Figure 7 illustrates the performance of our adaptive system that normalizes the scores. We did only one normalization pass, before starting the evaluation epoch. Unexpectedly, the performance cost increased about 37% with respect to the non-normalized run, but was still improved over the non-adaptive system (about a 50% cost drop). Three reasons could be behind that degradation. First, the normalization sample size might not be large enough to obtain good estimates of the off-topic distribution. Second, scores resulting from our approach for off-topic stories might not follow a Gaussian distribution, falsifying a basic assumption in the Z-score normalization technique. Finally, as the model changes, the pre-computed normalization factors become less and less representative because they were computed given a different model representation. We expect better performance to be achieved if we recompute these factors periodically.

The scores reported here are threshold–dependent, simply because the topic model changes dynamically[2]. If we use another threshold, we expect to have different scores, hence probably different DET curve and minimum cost. This makes the usefulness of DET curve in the evaluation of adaptive systems doubtful.

4. CONCLUSION AND FUTURE WORK

Our participation in TDT-2004 has yielded both a deeper understanding of evaluation issues for adaptive topic tracking and a substantial part of the evaluation infrastructure that we plan to leverage as we explore the design of adaptive topic tracking systems. We are particularly interested in the effect of alternative initial conditions (e.g., explicit queries and/or substantial numbers of marked on-topic and off-topic documents) and in modeling imperfect evidence of user preference (e.g., reading and/or

retention behavior); we therefore plan to explore possibilities for enriching the collection in ways that may also be of interest to other TDT participants. Before we commit to doing so, however, we will need to better understand the effect of incomplete judgments on the utility of the TDT-5 collection as a basis for system comparison. The large number of topics in the TDT-5 collection makes it somewhat better suited to evaluation designs that require extending the training epoch (because some topics with few relevant documents would be lost), but if there are concerns about the stability of effectiveness measures we might ultimately choose to use the more extensively annotated TDT-4 collection instead. We are, therefore, also particularly interested in characterizing the effect of incomplete judgments on the stability of both the TDT effectiveness measures and other measures that might offer a similar degree of insight (e.g., binary preference [5]). We look forward to meeting in Gaithersburg to discuss these issues!

REFERENCES

1. Levow, G.-A. and Oard, D.W. "Signal Boosting for Translingual Topic Tracking," in Allan, J., ed.., *Topic Detection and Tracking: Event-Based Information Organization*, Kluwer Academic Publishers, Boston, pp. 175–195, 2002.

2. He, D., Park, H.R., Murray, G.C., Subotin, M., and Oard, D.W., "TDT-2002 Topic Tracking at Maryland: First Experiments with the Lemur Toolkit," *TDT-2002 Working Notes,* Avail from http://www.umiacs.umd.edu/~daqingd/, 2002.

3. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M. and Gatford, M., "Okapi at TREC-3," Proceedings of TREC-3, Gaithersburg, MD, pp. 109-126, 1994.

4. Leek, T., Schwartz, R., and Sista, S., "Probabilistic Approaches to Topic Detection and Tracking," In James Allen, editor, *Topic Detection and Tracking: Event-based Information Organization*, chapter 4, pp. 67-84. Kluwer Academic Publishers, 2002.

5. Buckley, C. and Voorhees, E.M., "Retrieval Evaluation with Incomplete Information," Proceedings of SIGIR-2004, Sheffield, UK, pp. 25-32, 2004.

---

[2] Assuming the topic model has been changed in between.