Topic Tracking with the PRISE Information Retrieval System

Douglas W. Oard

College of Library and Information Services
University of Maryland
College Park, MD 20742

ABSTRACT

Information retrieval systems offer an attractive alternative to construction of a topic tracking system from scratch. The freely available PRISE vector space text retrieval system was applied to the TDT-2 topic tracking task. A simple version of the Rocchio formula was used for profile formulation and a retrieval status value threshold was used in conjunction with a temporal cutoff to make hard decisions. The results indicate that our simple approach produced a credible system, but comparison with results achieved by other systems indicates that there is room for improvement. The paper concludes by identifying some promising directions for further work that would be compatible with our approach.

1. Introduction

The topic tracking problem exhibits strong similarities to what has been called *information filtering* in the field of information retrieval [4]. In both cases, the goal is to process information objects arriving in a stream from some source based at least in part on observations of the user's reactions to previously seen objects. Two major variants of the text filtering problem exist, one in which hard decisions must be made to accept or reject information objects as they are processed and a second in which a buffered collection is arranged in a way that facilitates periodic review by an end user. The topic tracking task in TDT-2 adheres to this paradigm, processing information objects consisting of newswire texts and speech recognition transcripts, using binary relevance judgments of previously seen stories to represent observations of user behavior, and presenting both a ranked list and a set of hard decisions for use by the end user. It is thus natural to ask how experience gained with information filtering can be applied to the the topic tracking problem.

Despite the similarities, topic tracking poses some challenges that extend beyond those addressed to date in information filtering research. Most work on information filtering has focused on processing relatively homogeneous electronic text from newswire articles or postings to Internet discussion groups. Selection algorithms may need modifications to perform well in the face of speech recognition errors, and the requirement to handle sources with markedly different characteristics (newswire text and radio news stories, for example) adds an additional degree of complexity. Perhaps more importantly, information retrieval research in general has traditionally sought to optimize a static criterion, topical relevance. The shift to a generational criterion, stories that were created due to the occurrence of some event, may favor development of selection algorithms that are sensitive to the temporal structure of the source. Finally, the resources created for TDT-2 make it possible to investigate a wide range of interesting questions. The availability of audio materials, for example, would facilitate experiments with speech recognition systems that are optimized for a tracking effectiveness criterion rather than an application-independent criterion such as word error rate.

Our goal in TDT-2 was to determine the extent to which existing information filtering techniques are suitable for use in topic tracking applications. Our approach included three essential components: ranking based on vector similarity, profile creation using a simplified version of the Rocchio formula, and hard decisions based both on temporal factors and and on vector similarity.

2. Implementation Issues

In previous work on information filtering we have found it convenient to use a text retrieval system to provide the fundamental infrastructure for our experiments [3]. Text retrieval systems typically provide scalable components for tokenization, language-specific processing such as automatic suffix removal, and creation of story representations that are optimized to support selection decisions. Research-oriented systems typically also provide batch processing capabilities, control languages, and output formats that facilitate repeated runs under a variety of experimental conditions. For the experiments reported here we used PRISE, a vector space text retrieval system that is freely available from the National Institutes of Standards and Technology (NIST) [2].

Information retrieval systems that are designed to work with relatively static collections generally calculate the fraction of the collection in which each term appears (the "document frequency") and use that as a measure of specificity when calculating term weights. Since such collection statistics are not known in advance for filtering applications, the usual approach is to calculate them instead on a preexisting collection and then use the frozen values (perhaps with periodic updates) when computing term weights for newly arrived stories [1]. NIST added that capability to PRISE to support these experiments.

2.1. Profile Construction

In information filtering, the information need specification is normally referred to as a profile. Perhaps the simplest approach to profile construction is the Rocchio formula. Originally designed for interactive relevance feedback applications in which a query statement and examples of relevant and nonrelevant stories are typically available, Rocchio formed a linear combination of the query vector with the vector centroids of known relevant stories and known nonrelevant stories [6]. In our case the formal topic specification is not allowable as a search cue, so the first factor is eliminated. The relative weight of the relevant and nonrelevant centroids is a free variable in the Rocchio formula, and many operational systems use only the centroid of the relevant stories. We adopted that approach for our

experiments because it is easily implemented using PRISE. We ran only the $N_t = 4$ case, and presented every word in each of the four training stories as the query, selecting the option to retain duplicate terms. This produced the same query vector that would have been formed by creating a vector for each story and then computing the centroid of those vectors.

2.2. Collection Statistics

The temporal structure of the topic tracking task poses an interesting challenge with respect to the selection of the representative collection on which the frozen collection statistics are to be be computed. Clearly the known relevant stories should be included, because otherwise topic-specific vocabulary might be omitted entirely. And, just as clearly, some of the known nonrelevant stories should be included in order to capture representative statistics. But the simple expedient of using every known nonrelevant story would produce "representative" collections with very different densities of relevant stories because the number of relevant stories would be fixed while the number of nonrelevant stories would vary substantially across topics. We chose instead to count back a fixed number of stories from the last relevant training story. We did not know *a priori* how far back to look, so we chose the somewhat arbitrary figure of 1,000 stories (4 relevant, 996 nonrelevant).[1]

2.3. Hard Decisions

Vector space text retrieval systems compute a retrieval status value for each story that serves as the basis for rank ordering the stories in order of decreasing similarity to the profile. It is well known that these values are not comparable across collections with different collection statistics, but little additional information is available in the no-deferral case for which we designed our experiment. We thus chose to threshold the retrieval status value at a fixed value and to tentatively accept the stories that exceeded the threshold. The retrieval status values produced by PRISE are unnormalized inner products. Before thresholding the values we normalized the values to produce the cosine similarity measure by dividing by the retrieval status value of the profile itself. In order to choose a reasonable threshold we examined four topics in the development test collection for which relevance judgments were available and determined the first retrieval status value at which the density of relevant stories appeared by inspection to decrease markedly. The value we ultimately used, 0.27, was the average of the four values that we found in this way.

The temporal structure of the topic detection task also provides a second potentially useful basis for improving on threshold-based hard decisions. Inspection of the same four development test topics indicated that the density of relevant stories generally decreased relatively quickly, with the vast majority of the relevant stories found within 50 files of the fourth training story in three cases out of four.[2] We thus chose to accept only stories that were both above threshold and within 50 files of the last training story.

[1] We used only stories from the evaluation collection to develop the collection statistics, and in one case there were fewer than 996 known nonrelevant training stories.

[2] In the fourth case, there was a bimodal distribution with a second peak considerably further from the fourth training story.

3. Results

We ran the default condition, with known story boundaries, four relevant training stories, and either Dragon's one-best automatic transcription or newswire text (as appropriate to the source). PRISE was augmented with fully automatic scripts to prepare the profile for each topic, to make the hard decision for each story, and to postprocess the ranked list. PRISE normally produces output in TREC format (sorted by retrieval status value), and it omits stories that share no vocabulary with the profile. Our postprocessing script resorted the stories in the defined temporal order, adding a zero score for documents not returned by PRISE.

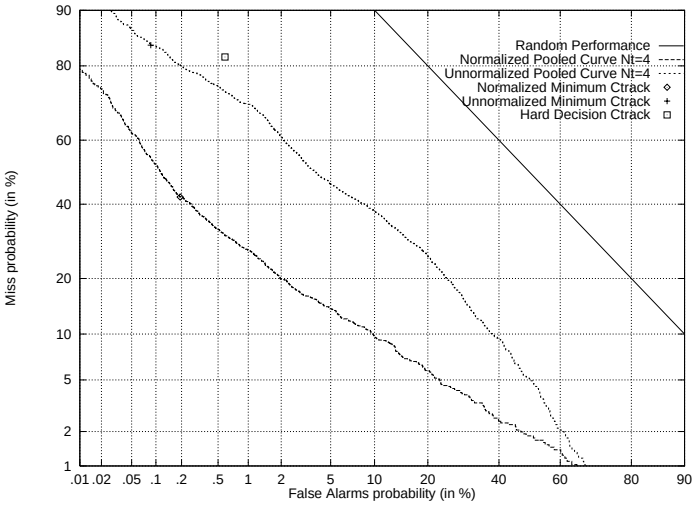

Figure 1: Official (upper) and corrected (lower) detection error tradeoff curves.

The detection error tradeoff curve for our scored run (the upper curve in Figure 1) appeared to indicate that our system was substantially outperformed by other participants. Examining the "breakeven point" at which the miss and false alarm probability are equal, for example, two systems achieved approximately 2% and several others achieved between 4% and 8%. The breakeven point for our scored results, by contrast, exceeded 20%. Although we had normalized the PRISE scores for the purpose of making hard decisions, in our official submission we reported the unnormalized scores. We corrected this error and rescored the results locally, producing the lower detection error tradeoff curve shown in Figure 1, which has a breakeven point of about 10%. Although there is still room for improvement, our corrected results are now much closer to the pack.

Because the time we could devote to this task was limited, we made little use of the development test collection to tune parameters and to explore alternate strategies for making the hard decision before submitting our results. We have begun to explore alternative parameter settings, and Table 1 shows some preliminary results from that work.[3] It appears that a lower file cutoff would be worth exploring, but that the number of training stories has little effect on the results. A firmer assessment will need to await our detailed examination of these results.

[3] Again, only the evaluation collection was used, so the number of training stories is an upper bound. Topic 100 was omitted from these results.

Training Stories	File Cutoff	Threshold	Story Wt C_{track}	Topic Wt C_{track}
100	50	.27	0.0220	0.031
100	100	.27	0.0245	0.036
100	200	.27	0.0298	0.043
1000	25	.27	0.0210	0.024
1000	50	.27	0.0222	0.028
1000	100	.27	0.0249	0.033
5000	50	.27	0.0220	0.031
5000	100	.27	0.0245	0.036
5000	200	.27	0.0298	0.043

Table 1: Effect of parameter variations on C_{track}.

4. Future Work

Like any large-scale evaluation program, assembling the needed infrastructure is half the battle. We are now in a position to use that infrastructure to explore some interesting questions that our experience in TDT-2 has raised. Perhaps our most important new insight is that if a temporal cutoff might help to improve our hard decisions then we should also look for a principled way to use similar information to improve the scores that are used to compute the detection error tradeoff curve. One obvious approach would be to use the development test collection to learn parameters for a linear combination of the score and the elapsed time since the final training story. We also may be able to do a better job on profile construction, perhaps by using terms extracted from a single nonrelevant story as suggested in [5].

Our present experimental infrastructure is rather inefficient, and that hampers our ability to easily explore the parameter space using the development test collection. The index structures used by PRISE are optimized for retrospective retrieval from relatively static collections, but we may be able to turn that to our advantage by indexing the profiles rather than the stories as suggested in [7].

5. Conclusions

Our goal was to demonstrate that a freely available information retrieval system could be easily used to produce a competitive system for topic tracking. Our corrected results suggest that such a goal may be within reach. We now have both the necessary tools and a suitable set of benchmarks for measuring our success, and we have identified some potentially useful techniques that remain to be explored. Both the text retrieval system that we are using and the scripts that we have developed to run our experiments using that system are freely available to others, so our work can be easily leveraged to reduce the barriers to entry by new teams in TDT-3.

6. Acknowledgments

The author is grateful to Vernon Warnick and Ruth Sperer for their assistance with the experiments and to Darrin Dimmick and Will Rogers of NIST for making the modifications to PRISE that we needed and helping us up the learning curve with that system. This work has been supported in part by DARPA contract N6600197C8540.

References

1. James Allan. Incremental relevance feedback for information filtering. In Hans-Peter Frei, Donna Harman, Peter Schäuble, and Ross Wilkinson, editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Hartung-Gorre Verlag, August 1996. http://ciir.cs.umass.edu/info/psfiles/irpubs/james-sigir96.ps.gz.
2. D. Dimmick, G. O'Brien, P. Over, and W. Rodgers. Guide to z39.50/prise 2.0: Its installation, use, & modification. http://www.nist.gov/itl/div894/894.02/, 1998.
3. Douglas W. Oard. Adaptive filtering of multilingual document streams. In *Fifth RIAO Conference on Computer Assisted Information Searching on the Internet*, June 1997. http://www.glue.umd.edu/∼oard/research.html.
4. Douglas W. Oard. The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, 7(3), 1997.
5. Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. Technical Report TR88-898, Cornell University, February 1988. http://cs-tr.cs.cornell.edu.
6. Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
7. Tak W. Yan and Hector Garcia-Molina. Index structures for information filtering under the vector space model. In *Proceedings of the Tenth International Conference on Data Engineering*, pages 337–347. IEEE Computer Society, February 1994. http://www-db.stanford.edu/pub/yan/1993/sdi-vector-model.ps.