University of Texas School of Information at TREC 2007

Miles Efron, Don Turnbull, Carlos Ovalle
{miles, donturn, cjovalle}@ischool.utexas.edu
School of Information
1 University Station, D7000
Austin, TX 78712

Introduction

This was the first year in which the University of Texas' School of Information (UT iSchool) participated in TREC. We limited our attention to a single task (feed distillation) within a single track (Blog). Our goal was to obtain high-precision results within a principled theoretical framework.

Our system used Apache's Lucene library [1] for its core indexing and retrieval functions. We also relied on language modeling extensions to Lucene provided by the Informatics Institute at the University of Amsterdam [2]. However, We altered these libraries to enable our IR approach. In particular, our results rely on a variant of the Kullback-Leibler (KL) divergence model [3, 4]. Given a query q we derive a score for each feed f in the corpus by the negative KL-divergence between the query language model and the language model for f. In the interest of maximizing precision at low numbers of documents retrieved, we limited our analysis to each feed's RSS posts, as opposed to its complete HTML representation.

Sources of Evidence and Preprocessing

As first-year participants, our chief goal was to establish a baseline for our general IR approach. Within this goal, we thought it most realistic to concentrate on improving retrieval at high levels of precision.

In hopes that we could avoid introducing noisy data into our system, we indexed only text in the RSS feeds of each blog. Thus we ignored information that appeared only in blogs' HTML-coded content, which included posts, blog software formatting, blogrolls, plugin-related content and other link text.

To improve retrieval at high precision, we did not use stemming of any kind. Nor did we employ any query expansion. Rather than cast our net widely by stemming or expanding, our attention was focused on finding a few high-quality results for each query.

Though the blog data set contains a good deal of spam and other putatively non-relevant material, we did not focus on spam detection or data cleaning. Our only effort to remove documents from consideration was a short, hand-crafted word blacklist. This list consisted mainly of obscene words and phrases; blogs with any of these words in their *title* element were not considered relevant.

Retrieval Model

The goal of the blog distillation task was to find RSS feeds that a reader interested in a particular topic would be likely to add to his or her RSS reader. Feeds were relevant if they evinced "a recurring, principle interest" in the particular topic. Thus the task required systems to generalize from specific blog posts to the feed that they appear in.

With this framework in mind, our approach relied on the notion of Kullback-Leibler divergence. Given a query *q* and a feed *f* we induced two language models over the words in the indexing vocabulary. Based on these models, our retrieval function matched feeds to queries based on the negative KL divergence between their language models:

$$\begin{aligned} s(q,f) &= -\mathrm{D}(\hat{\theta}_q \,||\, \hat{\theta}_f) \\ &= -\sum_w p(w \,|\, \hat{\theta}_q) \log p(w \,|\, \hat{\theta}_f) \end{aligned}$$

where the sum is taken over the *w* words in the indexing language. In all of our runs, language models were simple multinomials, fitted with Dirichlet smoothing, with a hyperparameter *m*=1000.

The appeal of the KL model for the topic distillation task lies in its flexibility when inducing the language model for a given feed. We experimented with two main approaches to defining the feed model.

Feed-Level Models

The simplest language models we used were calculated by considering all text encountered in a feed as a large "bag of words." Thus for the *ith* feed, the maximum likelihood estimate for the probability of the *jth* word was simply the frequency of word *j* divided by the number of tokens in feed *i.*

Post-Level Models

A more interesting approach to inducing each feed's language model lay in what we termed the 'post-level' approach. For these models, the probability of the *jth* word in the *ith* feed was estimated by the proportion of posts generated by feed *i* that contained word *j*.

Results

We submitted four runs to the blog distillation task, summarized in Table 1.

Name	*Priority*	*Description*
UTLC	1	A linear combination (evenly weighted) of feed- and post-level language model results

UTPLMU100	2	Post-level language model
UTBLNRR	3	Feed-level language model
UTBLRR	4	Feed-level language model, with an ad hoc result re-ranking applied

Table 1. UT iSchool runs submitted for the Blog Distillation Task

The ad hoc re-ranking algorithm applied in run UTBLRR increased the similarity of feeds based on the location of query terms within the text of a particular feed. i.e. Feeds with query terms near the beginning of their text (especially inside the *title* element) were given higher credence than feeds whose shared query terms appeared elsewhere in their text. This run was included primarily to see if adding several intuitive heuristics would improve our results over the more principled design of the other runs.

Table 2 summarizes our results. Mean average precision for our baseline run (UTLC) was 0.212. Only one other run showed statistically significant difference in MAP; UTBLRR gave MAP 0.179. A paired t-test over 50 queries yielded p=0.004, suggesting that the ad hoc re-ranking approach actively degraded performance. However, this effect did not present itself with respect to precision at 10 documents returned. In this case, the *p*-value was 0.523.

	MAP	P10
UTLC	0.212	0.453
UTPLMU100	0.212	0.453
UTBLNRR	0.22	0.451
UTBLRR	0.178	0.44

Table 2. Summary Results for UT iSchool runs

Figure 1 shows MAP for each of our submitted runs on each of the 50 topics the comprised the blog distillation task this year. The results of Table 2 are borne out in Figure 1; while MAP varied widely over the 50 topics, across our retrieval models, results were fairly static. Most importantly, this suggests that computing feed language models at the post- or feed-level did not bear heavily on the accuracy of the resulting retrieval.

Conclusion

Because this is the first year of the blog distillation task, at the time of this writing we cannot judge the overall quality of our results. Finding MAP in the area of 22% seems low, and we hope to improve this number in subsequent years. However, we were happier with our observed precision at 10 documents retrieved (P10). While we do not know the distribution of P10 scores across participants this year, we were happy to find that we held the false positive rate to a fairly low number (P10~45%) at 10 documents retrieved. This was especially gratifying after our experience in the relevance assessment portion of the track. While judging document relevance, we were surprised by the quantity of spam and other junk posts in the data. We were thus pleased to hold the amount of unwanted information to a low level early in our rankings.

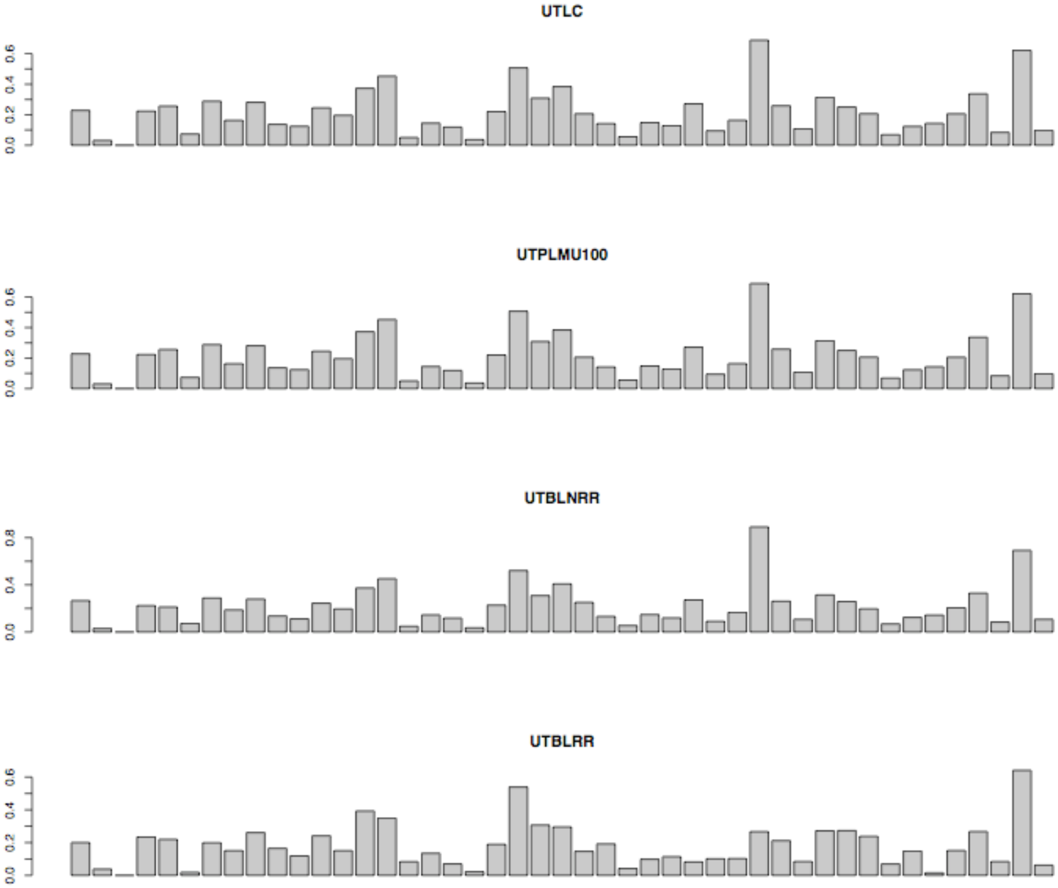

Figure 1. MAP for each of 50 Topics on each of UT iSchool's submitted runs

References

1. *Lucene*. 2007, Apache Foundation.
2. *Lucene Extensions for Language Modeling*. 2007, Informatics Institute. University of Amsterdam: Amsterdam.
3. Tao, T., et al. *Language Model Information Retrieval with Document Expansion*. in *Human Language Technology Conference of the North American Chapter of the ACL*. 2006. New York.
4. Zhai, C. and J. Lafferty, *A Study of Smoothing Methods for Language Models Applied to Information Retrieval.* ACM Transactions on Information Systems, 2004. **2**(2): p. 179--214.