TREC Blog and TREC Chem: A View from the Corn Fields

Yelena Mejova[c], Viet Ha Thuc[c], Steven Foster[d],
Christopher Harris[a], Bob Arens[c], Padmini Srinivasan[abc]

[a]Informatics Program
[b]Department of Management Sciences
[c]Department of Computer Science
The University of Iowa
Iowa City, IA, 52242

[d]Global News Intelligence
Montpelier, VT 05667

The University of Iowa Team, participated in the blog track and the chemistry track of TREC-2009. This is our first year participating in the blog track as well as the chemistry track.

BLOG Track

This year the Blog Track contained two tasks: Top Stories Identification and Faceted Blog Distillation Tasks. Our submissions for both tasks are described below. In this, our first entry into the blog track, we explore various strategies (latent Dirichlet relevance model, URL based ranking, query expansion etc.) for both tasks. We first indexed the blog data with Lucene and identified occurrences of Headline URLs in the permalink documents (which included the content of the posts as well as the side bars of the web pages). Text windows (+/- 800 characters including HTML code) surrounding the occurrences were harvested. The four runs submitted for the first task and the two for the second are described below.

Task 1: Top Stories Identification Task

The goal of this task is, given a unit of time (e.g. date), the system needs to identify the top news stories and provide a list of relevant blog posts discussing each news story. The ranked list of blog posts should have a diverse nature, covering different/diverse aspects or opinions of the news story[1]. Our system (IowaS) uses strategies built around two sub goals:

1. Rank headlines for a query date and
2. Rank relevant posts for top headlines.

We submitted four runs for this task.

Headline Ranking

Runs 1 and 2 are identical in ranking headlines, they differ in how they rank posts for a given headline. Exploring the idea that if a headline URL appears in a post then this indicates the post's relevance to the headline, we rank headlines by their URL frequencies in the blog collection. Sometimes it is the case that fewer than 100 headlines have URLs, then we randomly choose the rest of the headlines for submission. The distribution of the number of URLs citing a headline had a very long tail (see Fig 1).

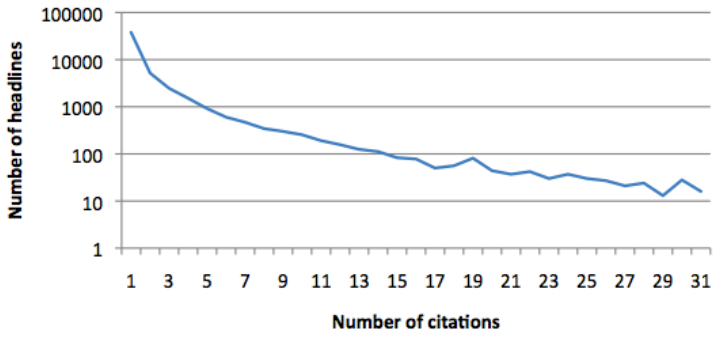

Figure 1: URL citation distribution for headlines

Because of the time differences between various time zones, a three day window was taken around the given query date. Comparatively few URL citations appear in posts within the 3-day window while many occur in followup posts or in permalinks dated *before* the 3-day window. On inspection, we found these predated URL references occurring in dynamically-generated sidebars; because the blogpost harvesting date was in most cases

[1]http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG

many months after the permalink date, a relevant headline citation could be found in the active sidebar of a permalink dated before the headline's publication date.

We noted that these non-contemporaneous URL citations are often very persistent throughout the feed, implying the "pinned" headlines were intended by the blog's author to be relevant to the blog's general topic orientation. We intend to further examine whether these citations indicate blog content which can be useful for language modeling and query expansion.

For Runs 1 and 2, URL citation ranks were created from the entire set of permalinks including those outside the 3-day window. Subsequently (after publication of the qrels) we compared runs 1 and 2 to results obtained using URL citations only from permlinks dated within the 3-day window. From Table 1, MAP and R-prec are .0867 and .1596 respectively in runs 1 and 2. Comparative results are MAP = 0.0868 and R-prec = .1599 when calculated with the 3-day permalink restriction.

In runs 3 and 4, text windows of +/- 800 characters (including HTML tags) around the URL occurrences were extracted and used as pseudo-relevant documents for the corresponding headlines. We explore the use of our latent Dirichlet relevance model [2] to estimate a language model (LM) for each headline from these pseudo-relevant documents. In run 3, posts are ranked by a measure indicating intensity of discussion. This measure is computed as the cosine similarity between the headline language model and the content of each post in the 3-day window. Note that because of the time differences between various time zones, a three day window was taken around the given query date.

Run 4 combines headline URL frequencies and the headline intensity measure. We rank headlines by the posterior probability:

$$p(headline|posts) = p(headline) * p(posts|headline) \quad (1)$$

Here, prior probability $p(headline)$ is proportional to the headline URL frequencies as in Runs 1 and 2. The likelihood is estimated by the similarity between the headline language models and the content of the posts as in Run 3.

Table 1: Headline Ranking Results

Measure	Run 1	Run 2	Run 3	Run 4	TREC Median (all submissions)
MAP	0.0867	0.0867	0.0880	0.0882	0.0445
R-prec	0.1596	0.1596	0.1601	0.1606	0.1075

Table 1 shows the results of the four runs submitted. The last column shows the median performance over all the groups that participated in the Blog Headline Track. Though the numbers seem low overall, our system

performs well above median for both MAP and R-precision measures. Performance scores are higher for runs 3 and 4; these runs returned more relevant documents - 572 instead of 559. Also there are no appreciable differences between runs 3 and 4. Thus including consideration of URL frequency while using the LM intensity measure does not offer an added advantage.

Blog Post Ranking

For the second part of task 1, the retrieval and ranking of blog posts for specific headlines, we used Lucene[2] to index the documents (using permalinks only). After several trial runs using the headline text only (with an internally generated training set), it became apparent that headline terms are not optimal for query design. Most of the headlines are "attention grabbers", and many don't contain any important keywords related to the article itself ("*Out With The Old, In With The New*", "*From Cult Figures to a Band of the Moment*", or "*Twists and Turns, Finish Line in Sight*"). Thus we explore query expansion.

For query expansion too we utilize the latent Dirichlet relevance model built from the text windows surrounding headline URL appearances in blog posts. Terms ranked at the top by the model are used in conjunction with phrases extracted from the original headline title to build a query. The phrases were extracted by dividing the title text using stop words and punctuation as separators. The original headline text was weighted more than the expansion terms. For example, here is a query for headline NYTimes-20080511-0032 *"DITORIAL; Rethinking Ethanol"*:

```
‘‘EDITORIAL Rethinking Ethanol’’^10 OR
‘‘editorial’’^5 ‘‘rethinking ethanol’’^5 OR
editorial^2.51 rethinking^1.32 ethanol^1.37 OR
tag^4.0 rel^3.92 energyoutlook.blogspot.com^3.84
oil^3.81 label^3.59 search^2.83 global^2.32
ethanol^1.61 limit^1.61 nymex^1.61
```

Here, the quoted title has the most weight, then follow the phrases, then each (non-stopword) term in the title, and then ten expansion terms extracted by the latent Dirichlet relevance model, with weights determined by the model. In run 1 the top retrieved 10 posts are returned for each headline. In runs 2, 3, and 4 the rankings were adjusted to boost posts containing the headline URL toward the top.

Table 2 summarizes the strategies for the four runs submitted to Blog Track. The best performance was achieved with Run 4 (see Table 1 for it's task 1 results) with MAP score of 0.0882 and R-prec of 0.1606.

[2]http://lucene.apache.org/

Table 2: Blog post ranking strategies

Run	Headline Ranking	Post Ranking
Run 1	URL Ranking	Retrieval with expansion terms + phrases
Run 2	URL Ranking	Retrieval with expansion terms + phrases + URL boosting
Run 3	Headline intensity ranking	retrieval with expansion terms + phrases + URL boosting
Run 4	Combination of URL and headline intensity ranking	retrieval with expansion terms + phrases + URL boosting

Table 3: Blog post ranking results

Mean Scores Across All Headlines		
	alpha-ndcg@10	IA-P@10
Run 1	0.341	0.099
Run 2	0.322	0.094
Run 3	0.328	0.097
Run 4	0.328	0.097
Performance Difference Compared to TREC Median		
	alpha-ndcg@10	IA-P@10
Run 1	[181,74,3]	[177,79,2]
Run 2	[171,86,1]	[169,87,2]
Run 3	[175,82,1]	[173,83,2]
Run 4	[175,82,1]	[173,83,2]

The post ranking results can be seen in Table 3. The legend to the above table is as follows: [X,Y,Z] where X is number of queries for which our Run gave better results than median performance for the query; Y same as median and Z worse than median. The one headline that our system consistently underperformed was NYTimes-20090120-0009, and underperformed some of the time for NYTimes-20080830-0069 and NYTimes-20080830-0044. Further study of this phenomenon is needed. On the whole, our system performs as good as or better than the TREC median.

Task 2: Faceted Blog Distillation Task

For the Faceted Blog Distillation Task we submitted two retrieval runs using the same Lucene index of blog posts.

In run 1 queries were composed using the `<query>` field and up to 10 terms extracted from the `<narrative>` field using TFIDF as the term ranking measure. Two hundred posts were retrieved and analyzed according to the querys facets. The top 100 satisfying a facet forms our submitted result set.

Three facets were explored: *opinionated* vs. *factual*, *in-depth* vs. *shallow*, and *personal* vs. *official*.

For the *opinionated* vs. *factual* facet a Lingpipe[3] classifier was used. The classifier was trained on 5000 "objective" and 5000 "subjective" sentences drawn from the Internet Movie Database (IMDB) archive[4] and the Rotten Tomatoes customer reviews[5]. Each post was classified as *subjective* or *objective*. Each set of posts were then returned preserving the search engine ranking.

For the *in-depth* vs. *shallow* facet the length of the posts was used (excluding stop words). The intuition is that in-depth discussion will produce longer documents than a shallow one. This hypothesis was examined manually with our training data, and post length was determined to be one of the surest ways to identify this facet. Again the set of posts were returned preserving the search engine ranking.

Finally for the *personal* vs. *official* facet we used the number of personal pronouns to rank posts. Here, we counted the occurrences of personal pronouns such as *I*, *mine*, *my*, etc. in the top 200 posts returned by the search engine. We then found the median number of personal pronouns and returned all posts above the median as personal and below - official, while conserving the original search engine's ranking.

In run 2 the same run 1 queries were first used to identify the top ranked 50 posts for each headline. Again using our latent Dirichlet relevance model to add ten expansion terms to the original query. The expanded query was searched against the Lucene index to retrieve again two hundred posts. These are then analyzed for facets using the same methods as in run 1.

Table 4: Summary Faceted Blog Distillation Task Results

	Run 1			Run 2			TREC Median
Measure	none	first	second	none	first	second	(all submissions)
MAP	0.07	0.0390	0.0262	0.0785	0.0467	0.0439	0.1265
R-prec	0.13	0.0394	0.0401	0.1368	0.0483	0.0662	0.1867

Table 4 shows the results for our two runs. The runs were evaluated first without looking at the facet (the "none" columns for each run), then looking at the first (i.e. *opinionated*, *in-depth*, and *personal*), and then looking at the second (*factual*, *shallow*, and *official*). A median score for all TREC submissions appears in the last column. The break down of the results by facet is shown in Table 5.

[3]http://alias-i.com/lingpipe/

[4]http://www.imdb.com/

[5]http://www.rottentomatoes.com/

Table 5: Faceted Blog Distillation Task Results by Facet

Opinionated Vs Factual				
	Run 1		Run 2	
Measure	opinion.	factual	opinion.	factual
MAP	[7,1,5]	[7,2,4]	[7,0,6]	[5,2,6]
R-prec	[6,6,1]	[2,9,2]	[5,6,2]	[1,10,2]
Personal Vs Official				
	Run 1		Run 2	
Measure	personal	official	personal	official
MAP	[6,1,1]	[5,1,2]	[5,2,1]	[5,1,2]
R-prec	[4,3,1]	[1,6,1]	[4,3,1]	[1,5,2]
In-depth Vs Shallow				
	Run 1		Run 2	
Measure	in-depth	shallow	in-depth	shallow
MAP	[9,4,5]	[13,3,2]	[7,4,7]	[13,2,3]
R-prec	[4,11,3]	[5,13,0]	[4,9,5]	[3,14,1]

The legend to the above table is as follows: [X,Y,Z] where X is number of queries for which our Run gave better results than median performance for the query; Y same as median and Z worse than median. On average we see a performance improvement between Run 1 and Run 2. For all facets our system performs better on the R-prec scores than MAP compared to the median runs. In the case of *in-depth* vs. *shallow*, our system performs better on the *in-depth* facet than the *shallow*. Thus, our strategies may favored one facet and not the other, which suggests separate approaches for each of the facet's values.

Closing Remarks

As a first year of participation in TREC, it has been a time to explore the tasks and the approaches possible. The new dataset distributed by the University of Glasgow[6] brings new opportunities and challenges. It took several weeks to index the permalink documents using a cluster of 14 machines. Further analysis is needed to determine the distribution of languages in the dataset, the relationship of the posting, commenting, and crawling dates, etc. Also, further study needs to be done on the nature of both tasks: are headlines sufficient for news story retrieval? what precisely does relevancy mean in the context of news publishing? how can blog community be leveraged to determine what stories are *really* important, and to whom? We hope to address these and other questions in the coming Blog Track years.

[6]http://ir.dcs.gla.ac.uk/test_collections/blogs08info.html

Chemistry Track

Strategy Overview

For our first year in the Chemistry track we chose to focus on Task 2: the Prior Art task. We were supplied with a dataset consisting of more than 100,000 chemical patents in XML format issued by the USPTO and EPO. We were also given 1,000 chemistry-related query patents and asked to return a list of up to 1,000 patents that could potentially invalidate a given query patent. We were not able to use the 'References Cited' field of the query patent - our task was to recreate this list with the provided patent dataset. For computationally-expensive submissions, participants also had the option of providing runs using only the first 100 query patents. Our team made one submission using all 1,000 query patents and two using the first 100 query patents.

Our initial intuition was that the claims section of a patent was central for invalidity searches, since claims are arguably the most important and most scrutinized part of a patent. Thus we began by producing two separate indexes using Lucene: one with patent claims alone; the other with the Title, Description, Abstract, and Classification Code portions (we use the acronym 'TDAC' to refer to this index). As claims are often nested within each other we first 'un-nest' them so they would each stand as an independent pseudo 'document' for indexing and retrieval.

Additional Training of the Patent Dataset

A training set of 15 EPO patents was provided to all participants. We noticed a majority of query patents were issued by the USPTO; hence we created a second training set of 15 randomly-selected US patents to train on as well (we only include those patents not in the 1,000-query test set). Indeed we found this second training set to be a better reflection of the results received from our submitted runs.

Description of Each Run

For our *first run* (UIowaS09PA1), we determined that each patent's claims should be run as individual queries against the Claims index and a separate query built from the Title, Description, Abstract, and Classification Code (TDAC) fields was run on the second index. While retrieving against the Claims index, in cases where multiple claims from the same patent appeared in the retrieved list, we took the most favorable, i.e., best rank/score for that patent. The two sets of results were merged through summarizing functions. We used our training queries to experiment with the relative weights to apply to the different summary functions. Table 6 shows the results on the first 100

queries using the trained function. We determined the best function to be one that weighted the score from the TDAC index as 10 times more important than the score returned by the claims index. This list of ranked patents was unduplicated where necessary. A key part of our retrieval strategy for this first run was to limit the retrieved patents to those with a priority date preceding the query patent's priority date as a threshold.

Our second and third runs were more experimental in nature and were run against the first 100 query patents. The *second run* (UIowaS09PA2) used only the primary classification information from the TDAC index to retrieve those patents with matching primary classifications. Priority dates were not used and they were ranked by ascending patent number. Our *third run* (UIowaS09PA3) was a refinement of our second run. Specifically, we worked on the assumption that patent numbers reflected a temporal sequence and only those candidate patent numbers lower than the target patent number were included. Thus, we made a better approximation of priority dates using this sequence at the expense of returning fewer patents per query.

A *fourth run*, also run against the first 100 queries, was not completed prior to the submission deadline. However, we believe it is interesting as we apply a technique that we had applied to a smaller dataset in previous research [1] and it shows promise for future examination. We apply this technique by first creating a machine-readable representation of the hierarchical IPC classification structure and then calculating a similarity measure between the primary classification code for each of the query patents and each of the 1,000 retrieved patents. We then re-rank these 1,000 retrieved patents for each of query patent by this similarity score. Using the first 100 queries in Run 1 as a baseline, this boosting technique demonstrated improvement on some key metrics (i.e., MAP increased by 49%). See Table 7 for the results for each run.

Results

In the first table below, we examine the effects of the ratio of weights on the two indexes (Claims and TDAC) across four measures: mean average precision (MAP), binary preference (bpref), recall after 100 retrieved documents (recall-100), and normalized discounted cumulative gain (ndcg). Although we initially thought that a retrieval method with a heavier weighting on Claims (vs. TDAC) would perform better, our results demonstrate this is not the case. For Run 1 we used the TDAC:Claims ratio of 10:1, which provided slightly better results than the other ratios we examined.

In Table 7, we show key metrics from each of our submitted runs (Runs 1-3) and one post-TREC run (Run 4) on these same metrics. Run 4 used the list of retrieved patents from Run 1 and had a boosting technique applied to re-rank them, improving the observed results across all four examined metrics. There are two medians provided: one for submissions by all participants across all 1,000 queries, and another for submissions by all participants across

Table 6: Determining the Ratio of Weights Between Summary Functions

TDAC:Claims Index Ratio	MAP	bpref	recall-100	ndcg
1:100	0.0203	0.1494	0.0621	0.0886
1:50	0.0206	0.1588	0.0643	0.0956
1:10	0.0250	0.1630	0.0754	0.1034
1:5	0.0267	0.1658	0.0839	0.1071
1:2	0.0313	0.1802	0.1067	0.1214
1:1	0.0348	0.1880	0.1134	0.1304
2:1	0.0375	0.2207	0.1256	0.1451
5:1	0.0469	0.4028	0.1962	0.2159
10:1	0.0485	0.4207	0.1888	0.2245
50:1	0.0479	0.4161	0.1412	0.2226
100:1	0.0466	0.4033	0.1379	0.2118

Table 7: Run Results

Runs	Query Size	MAP	bpref	recall-100	ndcg
Run 1	1000	0.0683	0.4066	0.1851	0.2643
Run 1	100	0.0485	0.4207	0.1888	0.2245
Run 2	100	0.0049	0.1457	0.0368	0.0616
Run 3	100	0.0066	0.1092	0.0447	0.0542
Run 4	100	0.1017	0.4401	0.1924	0.2813
TREC 1000 query median	1000	0.0279	0.3614	0.0594	0.1639
TREC 100 query median	100	0.0229	0.3950	0.0654	0.1525

the first 100 queries (the smaller dataset). Both Run 1 and Run 4 were above the median for all four metrics evaluated. Runs 2 and 3 performed relatively poorly compared with Runs 1 and 4, and poorly compared with the medians scores for these four metrics.

Closing Remarks

With MAP scores under 0.10 and ndcg scores under 0.25, the ability to find possible prior art violators is understood to be non-trivial - we see there is room for improvement in chemical patent search. As with the Blog track, we have considered a number of different techniques but have not yet implemented them. Much of our processing was done after indexing, which allowed us to examine the effects of different techniques quickly, but was reliant upon our underlying indexing strategy. In the coming months, we plan to explore the role of classification codes in more detail and examine each component of our TDAC index independently to determine which patent elements best comprise an effective index. We enjoyed participating in this track and look forward to participating in future TREC Chemistry tracks.

References

[1] Harris, C. Foster, S. Arens, R. Srinivasan, P., On the Role of Classification in Patent Invalidity Searches, In Proceedings of the 2nd Interational Workshop on Patent Information Retrieval (PaIR'09), (2009)

[2] Ha-Thuc, V. Srinivasan, P., A Latent Dirichlet Framework for Relevance Modeling, In Proceedings of the 5th AIRS (LNCS), (2009)