# PRIS at TREC 2010 Blog Track: Faceted Blog Distillation

Si Li, Yan Li, Jiayue Zhang, Jingyi Guan, Xueji Sun,
Weiran Xu, Guang Chen, Jun Guo
School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications
Beijing, P.R. China, 100876
ls198cf@gmail.com, buptly@yahoo.com.cn, jyz0706@gmail.com,
gjy8722.student@sina.com, sunxueji1986@yahoo.com.cn

## Abstract

This paper presents the system adopted for the Faceted Blog Distillation task by PRIS team. The PRIS system is submitted by Pattern Recognition and Intelligent System Lab at Beijing University of Posts and Telecommunications. And a two-stage strategy is involved for this task. First, an adaptable Voting Model is carried out for blog distillation. Then, different models are designed to judge the facets and ranking.

## 1 Introduction

We participate in the Faceted Blog Distillation task at TREC 2010 Blog Track. And the task is the same as the Faceted Blog Distillation task in 2009. And three kinds of facets are used. They are 'opinionated' vs. 'factual', 'personal' vs. 'official' and 'in-depth' vs. 'shallow'.

The PRIS system adopts a two-stage strategy in the faceted blog distillation. The first step is baseline blog distillation. This step only consists in ranking blogs which are relevant to the topic. An adaptable voting model with Posts Average algorithm (PA) is designed for blog distillation. In the second stage, different models are used for identifying different facets. For 'opinionated' vs. 'factual' facets, the opinion lexicon and the factual lexicon are adopted for sentiment analysis to make a distinction between these two facets. Then, an improved in-depth analysis model based on the *L-Qtf* (Length-Query term frequency) coefficient is carried out for 'in-depth' vs. 'shallow' facets. Meanwhile, a personal lexicon and an official lexicon are generated by Information Gain (IG).

In Section 2, we introduce the blog distillation algorithm and facets models respectively. In Section 3, the evaluation of the faceted blog distillation system is presented. Finally in Section 4, conclusions and comments on the future work are given.

## 2 The Faceted Blog Distillation System

### 2.1 Blog distillation

The aim of the blog distillation is to identify blogs which have a recurring interest in the query topic area. In our system, we use the adaptable Voting Model [1] for blog distillation. In this model, blogs are ranked by learning the ranking of posts with respect to the query. If a blog has many associated posts highly ranked in the ranking of posts, these are seen as votes and the blog will be ranked higher than another blog with less or lower ranked posts. In the simplest technique, called Votes, blogs are ranked by the aggregation of their posts ranked in response to a query. In particular, the retrieval score for a blog *B* with respect to a query *Q*, denoted *Score*(*B,Q*) is:

$$Score_{Votes}(B,Q) = \left\| R(Q) \cap posts(B) \right\| \qquad (1)$$

where *R*(*Q*) is the underlying ranking of blog posts, and *posts*(*B*) is the set of posts belonging to blog *B*. Note that each post is associated to exactly one blog.

Our system ranks each blog by the sum of the relevance score of all the retrieved posts of the blog, and strengthens the highly scored posts by applying the root function (strong votes evidence):

$$Score_{rootSum}(B,Q) = \sum_{p \in R(Q) \cap posts(B)} \sqrt{score(p,Q)} \qquad (2)$$

However, an issue using such a technique is that the productive bloggers may gain an unfair advantage in the ranking. This is because the more a blogger writes, the more likely a query term will appear at random in a blog post (e.g., many blog posts contain links to other recent posts, with the title of each post identical to the link anchor), and hence the blog will receive extra erroneous votes. To this end, we adapt a normalization technique, called Posts Average algorithm (PA), with regard to the number of posts of blog. The normalized score of a blog is adapted as follows:

$$Score_{Norm}(B,Q) = \frac{\sum_{p \in R(Q) \cap posts(B)} \sqrt{score(p,Q)}}{\left| posts(B) \right|} \qquad (3)$$

Where |*posts* (*B*)| denotes the total number of posts of blog *B*.

Moreover, query expansion is added to our system to improve the retrieval accuracy. From the aspect of topic understanding, the Learning Query Expansion (LQE) model based on semi-machine learning method is designed as we have done at the Blog Track 2009 [2].

We trained LQE model based on CRFs with the manual Blog track 2008 queries which were expanded based on the human common sense and comprehension. After the classifier was trained, it was applied to the whole Blog track 2010's queries for query expansion which contains both expansion words and their weightings with Indri query language. One of the final query examples is as the following:

```
<query>
    <number>1151</number>
    <text> information   warfare   #5(information warfare).(title)
          #weight(1.0 #combine(information warfare) 0.8 attacks 0.8 cybersecurity
          1.0 cyberwarfare 0.8 information 0.1 warfare )
    </text>
</query>
```

## 2.2 Opinionated vs. Factual Model

This model contains three stages. Firstly, an existing opinion lexicon proposed in [3] is involved in our system. For the factual lexicon, it is generated automatically in this step. Secondly, the opinionated lexicon and the newly generated factual lexicon are utilized to calculate the opinion score and factual score respectively. Finally, a ranking scheme is used to generate the final ranking of opinionated and factual blogs.

### 2.2.1 Generating a factual lexicon

The factual lexicon is generated automatically based on Information Gain (IG) and Mutual Information (MI). For each term *t* in blog posts, its IG weight is calculated as follows.

$$IG(t) = p(t)[p(O \mid t)\log\frac{p(O \mid t)}{p(O)} + p(F \mid t)\log\frac{p(F \mid t)}{p(F)}] + p(\bar{t})[p(O \mid \bar{t})\log\frac{p(O \mid \bar{t})}{p(O)} + p(F \mid \bar{t})\log\frac{p(F \mid \bar{t})}{p(F)}] \quad (4)$$

*O* and *F* denote the opinionated and factual blogs respectively.

It is assumed that the number of opinionated and factual blogs in training collection is *A* and *B* respectively. Then,

$$p(t) = \frac{df(t \mid A,B)}{A+B} \quad (5)$$

$$p(O) = \frac{A}{A+B} \quad (6)$$

$$p(O \mid t) = \frac{df(t \mid A)}{df(t \mid A,B)} \quad (7)$$

$$p(O \mid \bar{t}) = \frac{A - df(t \mid A)}{(A+B) - df(t \mid A,B)} \quad (8)$$

$p(\bar{t})$, $p(F)$, $p(F|t)$ and $p(F|\bar{t})$ can be easily deduced according to the equations above. Terms whose IG values are above the threshold, we have previously set, are selected as candidates of the lexicon.

For the candidates produced above, we further compute term weight according to a document-frequency based on the version of the Mutual Information metric [4].

$$MI(t)_{\mathrm{fa}} = p(t,F)\log\frac{p(t,F)}{p(t)p(F)} + p(\bar{t},O)\log\frac{p(\bar{t},O)}{p(\bar{t})p(O)} \quad (9)$$

$$p(t,O) = \frac{df(t \mid A)}{A+B} \quad (10)$$

$p(t,F)$, $p(\bar{t},F)$ and $p(\bar{t},O)$ can be easily deduced according to the equations above. Another threshold is used to generate the final factual lexicon.

### 2.2.2 Computing opinion score and factual score

Given a query, for each term *t* in the opinion lexicon or factual lexicon, we first compute a tf-idf weight $w_{tfidf}(t)$ in the relevant document collection provided by the baseline blog distillation task. Simultaneously, we use a BoI term weighting model [5] to compute the query weight $w_{bol}(q)$ in the collection. Then we add the two weight to get the opinion score $score_{op}$ or factual score $score_{fa}$.

### 2.2.3 Ranking

First we get the relevance score *Score(B,Q)* in baseline for each blog. And then we use *Score(B,Q)*×*score*$_{op}$ as the final score for ranking opinionated blogs and *Score(B,Q)*×*score*$_{fa}$ as the final score for factual blogs.

## 2.3 In-depth vs. Shallow Model

In the in-depth facet stage, the improved in-depth analysis model is adopted. The facet of a blog is judged based on all the posts in it. In common sense, an in-depth post expresses author's opinion on the given topic in detail with a long length in ideal situation. For minimizing the impact of spam contents, the length with average length is considered as a feature of the in-depth degree. But only using the length feature isn't sufficient, to confirm the relevance degree, considering the query term frequency in the post is also necessary. The posts' length and the query term frequency are combined as the following *L-Qtf* coefficient [2]:

$$L - Qtf = \sum_{t \in Q \cap D} \frac{1 + \ln(1 + \ln(tf))}{(1 - s) + s\frac{dl}{avdl}} \times qtf \tag{11}$$

where *tf* and *qtf* represent the query term frequency in the post and in the query respectively. The *tf* and *qtf* are calculated after stemming. *dl* is the post length and *avdl* is the average-length of the whole relevant posts for the topic. *s* is a parameter which is set as 0.2 in our experiments. The *L-Qtf* coefficient is a kind of pivoted weighting coefficient [6] [7].

Based on the whole posts of the topic-relevant blogs given by the blog distillation, the posts are ranked according to the in-depth coefficient. In the ranking list, the top 45% of topic-relevant posts are considered as the in-depth, while the last 45% posts are the shallow. *indepth(post*$_i$*,Q)* and *shallow(post*$_i$*,Q)* represent the post whether it is in in-depth or shallow. *indepth(post*$_i$*,Q)* is the total number of the in-depth posts. If *post*$_i$ is in the top 45% of ranking list, *indepth(post*$_i$*,Q)* is 1, and 0 otherwise. Similarly, *shallow(post*$_i$*,Q)* is the total number of the shallow posts. The in-depth degree (*Score*) of each blog is calculated according to the relationship between the in-depth posts and shallow posts as the following equation.

$$S_i = Score(b\log_x, Q) = \frac{\sum_{i=1}^{n} indepth(post_i, Q) - \sum_{i=1}^{n} shallow(post_i, Q)}{n} \tag{12}$$

$$indepth(post_i, Q) = \begin{cases} 1 & post_i \text{ is a indepth poster} \\ 0 & other \end{cases} \tag{13}$$

$$shallow(post_i, Q) = \begin{cases} 1 & post_i \text{ is a shallow poster} \\ 0 & other \end{cases} \tag{14}$$

The larger the *Score* is, the deeper the feed is. Otherwise, the shallower the feed is.

According to the experiment, the ranking according to the *L-Qtf* is more effective in the in-depth facet, while the shallow facet is more dependent both on *L-Qtf* and the length of the post.

In the in-depth facet task, the feed should be judged not only the topic relevance but also the facets. By considering these two points, the combination model is adopted.

$$S_j = \begin{cases} Score(b\log_x, Q) \times Score_{Norm}(B,Q) & Score(b\log_x, Q) \geq 0 \\ 1 - Score(b\log_x, Q) \times Score_{Norm}(B,Q) & Score(b\log_x, Q) < 0 \end{cases} \tag{15}$$

$S_j$ is the final confidence value of the blog B. *Score(blog$_x$,Q)* is the facet result as Eq.(12). *Score$_{Norm}$(B,Q)* is got from the result of Blog distillation. The combination model use multiplication to consider both topic relevance and facets result. According to the experiment, the combine model with multiplication is more effective than the model with addition, as Eq. (16) [2].

$$S_j = \mu \times Score(b\log_x, Q) + (1-\mu) \times Score_{Norm}(B,Q) \tag{16}$$

μ is a weighting parameter that distributes in the interval [0, 1] and balances the scores of facet level and similarity.

## 2.4 Personal vs. Official Model

For the personal vs. official facet, we get Information Gain (IG) values of the terms. After that, we extract the terms with higher IG values to build lexicons with considering the factor of sentiment at the same time. Then the lexicons are used to score and rank the related blogs respectively.

### 2.4.1 Calculating IG

We calculate IG values of the terms using the TREC Blogs08 collection.

$$\begin{aligned} IG(t) = {} & p(t)[\, p(c_1 \mid t) \log \frac{p(c_1 \mid t)}{p(c_1)} + p(c_2 \mid t) \log \frac{p(c_2 \mid t)}{p(c_2)}] \\ & + p(\bar{t})[\, p(c_1 \mid \bar{t}) \log \frac{p(c_1 \mid \bar{t})}{p(c_1)} + p(c_2 \mid \bar{t}) \log \frac{p(c_2 \mid \bar{t})}{p(c_2)}] \end{aligned} \tag{17}$$

where $t$ is the term we want to process. $c_1$ is the personal facet, and $c_2$ is the official facet. The essence of IG is that the term with larger IG value can distinguish the two classes. Then, we select the terms which IG values are above certain threshold. Considering the factor of sentiment, we also pick out the sentiment terms to improve the results.

### 2.4.2 Building lexicons

In the procedure of building the lexicons [8], the Mutual Information metric is split into two parts to gain personal and official facet weights.

$$personal(t) = p(t, personal)\log\frac{p(t, personal)}{p(t)p(personal)} + p(\bar{t}, official)\log\frac{p(\bar{t}, official)}{p(\bar{t})p(official)} \tag{18}$$

where *p(t, personal)*, *p(t)* and *p(personal)* are defined as follows:

$$p(t, personal) = df(t, personal) / |R| \tag{19}$$

$$p(t) = df(t, R) / |R| \tag{20}$$

$$p(personal) = df(personal) / |R| \tag{21}$$

where *df(t, personal)* is the number of personal documents containing the term *t*. *R* is the number of relevant documents in the collection, including personal and official ones.

The official facet weight is calculated analogously as follows:

$$official\,(t) = p(t, official\,)\log\frac{p(t, official\,)}{p(t)\,p(official\,)} + p(\bar{t}, personal\,)\log\frac{p(\bar{t}, personal\,)}{p(\bar{t})\,p(personal\,)} \qquad (22)$$

### 2.4.3 Scoring and Ranking

We use Vector Space Model (VSM) to score the blogs [9]. ($p(t_1|post)$, $p(t_2|post)$, $p(t_3|post)$ …$p(t_i|post)$) and ($personal(t_1)$, $personal(t_2)$, $personal(t_3)$ …$personal(t_i)$) can represent a post we want to judge and personal lexicon respectively. The score of the post belonging to personal facet is calculated as follows.

$$personal\_score\,(post\,) = \sum_{i=1}^{n} p(t_i \mid post\,)\,personal\,(t_i) \qquad (23)$$

Similarly, the official facet score is calculated as follows:

$$official\_score\,(post\,) = \sum_{i=1}^{n} p(t_i \mid post\,)\,official\,(t_i) \qquad (24)$$

Since a blog is comprised of many posts, its score of personal/official facet should be the addition of posts' personal/official score. Finally the score of a blog can be as follows.

$$score\,(b\log) = \frac{\sum_{i=1}^{n} personal\_score\,(post_i) - \sum_{i=1}^{n} official\_score\,(post_i)}{n} \qquad (25)$$

We rank this score in descending order. Then from the top to the bottom of the ranking list, the blogs' inclination of personal facet becomes weaker while the inclination of official facet becomes stronger. Finally, we get 100 personal and 100 official blogs with their ranking respectively.

# 3 Submission and Evaluation Results

We have done many experiments on this track. In this section, we present empirical evaluation results of our different versions. We employed four performance metrics: mean average precision (MAP), binary preference (bPref), rPrec and P@10.

## 3.1 Blog distillation

We submitted 2 runs. The difference between the 2 runs is query. The first run is without query expansion, the words in the title field are only used. The second run is expanded by LQE. The evaluation results of the 2 submitted runs are listed in Table1. Pris and Prisb denote the “query-only” run and the “query-expansion” run respectively. From these data, it proves that for the first value the LQE is effective while “query-only” run is effective for the second value.

## 3.2 Faceted blog distillation

There are many runs for this sub-task, listing in Table2 and Table3. Q0 stands for the “query-only” run and QE stands for the “query-expansion” run. Std represents the standard baseline 1 that we used in our system and it means our own baseline if there is no “Std” in the run-tag label. PrisQ01, PrisQE1, PrisStdQ02, and PrisStdQE1 use both the opinion lexicon and the factual lexicon and normalization is also adapted, while PrisQ02, PrisQE2, PrisStdQ0 and

PrisStdQE2 use the opinion lexicon and the normalization scheme but not the factual lexicon. The two lexicons are also utilized in PrisQ03, PrisStdQ03 and PrisStdQE3 but normalization is not adapted in them. For PrisQ04 and PrisStdQ04, only the opinion lexicon is used.

Table2 shows that PrisStdQ02 obtains the best result for MAP, PrisStdQE2 performs best at bPref. In addition, PrisStdQ02 and PrisStdQ04 get the same highest score for R-prec and P@10.

**Table1. Blog distillation results**

| | MAP | bPref | R-prec | P@10 |
|---|---|---|---|---|
| pris.none | 0.2355 | 0.2393 | 0.2981 | 0.3417 |
| prisb.none | 0.2210 | 0.2271 | 0.2885 | 0.3250 |
| pris.first | 0.1218 | 0.0973 | 0.1414 | 0.1542 |
| prisb.first | 0.1296 | 0.1056 | 0.1466 | 0.1292 |
| pris.second | 0.1668 | 0.1391 | 0.1731 | 0.1500 |
| prisb.second | 0.1625 | 0.1278 | 0.1519 | 0.1583 |

**Table 2. Faceted blog distillation results of the first value**

| | MAP | bPref | R-prec | P@10 |
|---|---|---|---|---|
| PrisQ01 | 0.0724 | 0.0690 | 0.0796 | 0.0958 |
| PrisQ02 | 0.0679 | 0.0584 | 0.0620 | 0.0792 |
| PrisQ03 | 0.0674 | 0.0635 | 0.0750 | 0.0917 |
| PrisQ04 | 0.0641 | 0.0568 | 0.0555 | 0.0750 |
| PrisQE1 | 0.0678 | 0.0675 | 0.0789 | 0.0833 |
| PrisQE2 | 0.0732 | 0.0692 | 0.0743 | 0.0750 |
| PrisStdQ0 | 0.1037 | 0.0879 | 0.1031 | 0.1083 |
| PrisStdQ02 | **0.1270** | 0.1254 | **0.1412** | **0.1167** |
| PrisStdQ03 | 0.1060 | 0.0894 | 0.1099 | 0.1125 |
| PrisStdQ04 | 0.1265 | 0.1264 | **0.1412** | **0.1167** |
| PrisStdQE1 | 0.0931 | 0.0789 | 0.0883 | 0.1042 |
| PrisStdQE2 | 0.1166 | **0.1279** | 0.1315 | **0.1167** |
| PrisStdQE3 | 0.1137 | 0.1258 | 0.1315 | 0.1125 |

**Table 3. Faceted blog distillation results of the second value**

| | MAP | bPref | R-prec | P@10 |
|---|---|---|---|---|
| PrisQ01 | 0.0714 | 0.0631 | 0.0621 | 0.0667 |
| PrisQ02 | 0.0451 | 0.0525 | 0.0623 | 0.0500 |
| PrisQ03 | 0.0442 | 0.0493 | 0.0585 | 0.0417 |
| PrisQ04 | 0.0448 | 0.0510 | 0.0623 | 0.0500 |
| PrisQE1 | 0.0791 | 0.0813 | 0.0926 | 0.0708 |
| PrisQE2 | 0.0801 | 0.0880 | 0.0856 | **0.0750** |
| PrisStdQ0 | 0.0716 | 0.0667 | 0.0642 | 0.0625 |
| PrisStdQ02 | **0.0956** | **0.0975** | 0.0999 | 0.0583 |
| PrisStdQ03 | 0.0706 | 0.0648 | 0.0642 | 0.0625 |
| PrisStdQ04 | 0.0689 | 0.0651 | 0.0642 | 0.0625 |
| PrisStdQE1 | 0.0947 | 0.0952 | **0.1066** | 0.0708 |
| PrisStdQE2 | 0.0432 | 0.0612 | 0.0583 | 0.0500 |
| PrisStdQE3 | 0.0432 | 0.0612 | 0.0583 | 0.0500 |

In Table3, the best performance on MAP, bPref, R-prec and P@10 is PrisStdQ02, PrisStdQ02, PrisStdQE1 and PrisQE2 respectively.

# 4 Conclusion

In this paper, we present a system for the faceted blog distillation. Compared Table 2 and Table 3 with Table 1, it can be concluded that most faceted models take negative feedback to the baseline. In the feature research, we will focus on exploring much more efficient faceted models for the faceted blog distillation.

# References

[1] C. Macdonald and I. Ounis. Voting for Candidates: Adapting data fusion techniques for an Expert Search task. In Proceedings of CIKM 2006, 2006.

[2] S. Li, H. J. Gao,et al..A Study of Faceted Blog Distillation--PRIS at TREC 2009 Blog Track, In proceeding of TREC 2009, 2010.

[3] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann, “Recognizing contextual polarity in phrase-level sentiment analysis,” in proceedings of HLT/EMNLP, 2005.

[4] C. D. Manning and H. Schtze, “Foundations of statistical natural language processing,” The MIT Press, June 1999.

[5] G. Amati, “Probabilistic models for information retrieval based on divergence from randomness,” PhD thesis, University of Glasgow, 2003.

[6] A. Singhal, C. Buckley, M. Mitra. Pivoted document length normalization. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. pp21-29,1996.

[7]http://zh.wikipedia.org/zh-cn/%E6%96%87%E6%9C%AC%E4%BF%A1%E6%81%AF%E6%A3%80%E7%B4%A2

[8] Mostafa Keikha,Mark Carman,Robert Gwadera,Shima Gerani,Ilya Markov,Giacomo Inches,Az Azrinudin Alidin and Fabio Crestani.University of Lugano at TREC 2009,2009.

[9] Chistopher D.Manning,Prabhaker Raghavan and Hinrich Schutze.Scoring,term weighting and the vector space model.An Introduction to Information Retrieval,pages 120-126,2008