PKUTM at TREC 2010 Blog Track

Liqiang Guo[1], Fangzhou Zhai[2], Yan Shao[2], Xiaojun Wan[1]
Institute of Computer Science and Technology, Peking University
[1]{guoliqiang,wanxiaojun}@icst.pku.edu.cn
[2]{allsamenames,shaoyan}@pku.edu.cn

ABSTRACT

This paper describes the PKUTM participation in the TREC 2010 Blog Track. We only concentrated on the Faceted Blog Distillation Task this year. Our system adopts a two-stage approach for this task. In the first stage, our system makes use of an IR platform - indri to obtain the top N ad-hoc topic-relevant blog posts for each query. In the second stage, different models are designed to identify the facet inclination. The experimental results show the effectiveness of our approach.

1. Introduction

In this paper, we describe the participation of PKUTM in the TREC 2010 Blog Track. The Blog track explores the information seeking behavior in the blogosphere，and it is first introduced in TREC 2006 [8], with a main pilot search task, namely the opinion-finding task. This year there are also two tasks: Faceted Blog Distillation Task and Top Stories Identification Task. The PKUTM group only concerned the Faceted Blog Distillation Task. The PKUTM system is based on the *indri* [6] framework and it makes use of a two-stage approach. The system first retrieval the top N topic-relevant blog posts and then analyzes them for the three facet inclination identification sub-tasks, respectively. For the opinion/factual facet, our system uses two different ranking strategies and a novel opinion retrieval model. For the personal/official facet, the facet is predicted based on the proportion of pro-nouns, the presence of named entities and offensive words. For the In-depth/shallow facet, the facet is considered closely related to the proportion of the regular words according to the word-building rules.

2. Collection and Preprocessing

The TREC blog08 collection consisting of permalinks, feeds and blog homepages is again used in TREC 2010. We used only the permalinks in the Faceted Blog Distillation Task. The permalinks encoded by HTML contain relevant content and many irrelevant contents such as HTML tags, advertisements, site descriptions and menus. For the Faceted Blog Distillation Task, the irrelevant contents are noises. Thus, we have to extract the relevant content from the permalinks. A simple but effective algorithm is proposed to get the relevant content. We first assume the content that invariably appears in each post of a certain feed is irrelevant [3].Then, we regard most of the hyperlinks as another irrelevant content, for example the advertisements. However, not all of these hyperlinks are irrelevant. The algorithm proposed by us can identify the two kinds of irrelevant contents above effectively.

For example, let p_i be any blog post, and let p_j be the blog post whose blog feed is the same as p_i.

$$
\begin{aligned}
&Noise(p_i)\\
&= (Content(p_i) \cap Content(p_i))\\
&+AdHyperlink(p_i)
\end{aligned}
$$

where $Noise(p_i)$ denotes the irrelevant content of p_i, $Content(p_i)$ and $Content(p_j)$ denote the content of p_i and p_j respectively, and $AdHyperlink(p_i)$ denotes the irrelevant hyperlinks of p_i.The idea that identifies the irrelevant hyperlinks is similar to [11]. Finally,

$Noise(p_i)$ and HTML tags are filtered out from p_i.

In addition, we find that the comments can lower the accuracy in the opinion/factual and official/personal facet inclination identification sub-tasks. Due to the lacking of common method to remove comments over different web sites, we simply make use of the first part of the blog post instead of the whole in these two sub-tasks.

3. Faceted Blog Distillation Task

In this section, we describe our approaches for the Faceted Blog Distillation Task in detail.

3.1. Topical blog distillation sub-task

Firstly, we have to obtain the top *N* ad-hoc topic-relevant blog posts. In our system, we set *N* as 10000.

3.1.1. Query Expansion

The topics of TREC2010 contain five fields ,namely *'num', 'query', 'desc', 'facet'* and *'narr'*. We consider that *'desc'* and *'narr'* are helpful to retrieve the topic-relevant blog posts. We design a simple but effective algorithm to extract the useful words from the two fields which can be used to expand the query. Query expansion effectively deals with the word mismatch problem caused by the short queries. Since queries for the Faceted Blog Distillation Task are usually short, we expect that query expansion could play an important role for improving the performance of topic-relevant retrieval For example, below is one sentence of *Topic 1103* in TREC 2009.

> *I want to find blogs about farm subsidies in the United States.*

We regard the words of *'farm', 'subsidies', 'United'* and *'States'* as useful information, and the remaining words such as *'want' 'find'* are useless for retrieving topic-relevant blog posts. It is easy to summarize this conclusion that the nouns of a sentence are probably useful words. So our algorithm extracts all the nouns and noun phrases of the sentences. Finally, some stop words were removed from them. The *Stanford*[1] *Parser* and *Tregex*[2] are used to get the nouns and noun phrases from the parser trees.

3.1.2. Baseline

In the *baseline* stage, we submitted two baselines as follows:

1. **PKUTMB1** is an automatic '*query-only*' run which is compulsive in TREC 2010.In this run, participants are allowed to use only the *'query'* field of the *topic*. Since *indir* [6] support structure query language, an example query of this run for *Topic 1103* is as follows:

```
<query>
#weight(1.0 farm 1.0 subsidies
2.0 #1(farm subsidies)
1.5 #uw5(farm subsidies) )
</query>
```

2. **PKUTMB2** is also an automatic run. The query of this run consists of, apart from the *'query'* field of the *topic*, the expansive words which are given by algorithm 3.1.1. For instance, the query for *Topic 1103* is as follows.

```
<query>
#weight(1.0 farm 1.0 subsidies
2.0 #1(farm subsidies)
1.5 #uw5(farm subsidies)
0.8#uw5(united states)
0.5 #combine(united states farmers
government farmers products ))
</query>
```

Since the ranking unit of the Faceted Blog Distillation Task is blog feed, we need to obtain the topic-relevant score of each blog feed. The feed's topic-relevant scores of the above two baselines are both calculated as follows:

[1] http://nlp.stanford.edu/software/lex-parser.shtml
[2] http://nlp.stanford.edu/software/tregex.shtml

$$Score_R(Feed) = (Max_{p \in Feed^{Top}}(Score(p))) \times \frac{|Feed^{Top}|}{|Feed|}$$

Where $Score_R(Feed)$ is the topic-relevant score of *Feed*, *Score (p)* is the *indri's* retrieval score of blog post *p* belonging to *Feed*, and $|Feed^{Top}|$ and $|Feed|$ are the numbers of corresponding blog posts in the Top *N* collection and the whole blog08 collection, respectively.

3.2. Facet inclination identification sub-task

In this section, we introduce our models of opinion/factual, personal/official and In-depth/shallow facet inclination identification sub-tasks respectively. In this second stage, we applied our facet models on these blog posts retrieved in the first stage.

3.2.1. Opinionated vs. Factual Model

As we aim to find the blog feeds which are not only interested in a given topic, but also make opinionated expressions on this topic, we adopt two different ranking strategies - Average Strategy and Maximum Strategy, and a novel Opinion Retrieval Model to solve this problem. These approaches are all based on the presence of sentiment words.

1. Sentiment Lexicon

For the opinion facet identification sub-task, we constructed our own sentiment lexicon based on the following lexicons.

SetntiWordNet

SentiWordNet [2] is a lexical resource for opinion mining. *SentiWordNet* assigns to each synset of *WordNet* three sentiment scores: positivity, negativity, objectivity. We can get the opinion score of each synset by summing the positivity score and negativity score. For one word , if any opinion score of the synsets that this word belongs to is not smaller than *o.6* ,we add it to our own sentiment lexicon.

HowNet

HowNet[1] is a knowledge database of the Chinese language, and some of the words in the dictionary have positive or negative properties. We use the English translation of those sentiment words provided by *HowNet*. There are 1001 negative sentiment words and 769 positive sentiment words. Since the *HowNet* words do not have opinion scores, we simply assign 0.8 to each word as its opinion score. Besides, there is an opinion operator lexicon in *HowNet*. Following [9], we consider that operator words such as *'advocate'*, *'believe'* are import clues for the sentences which contain the author's opinion. We simply assign *1.0* to each operator word as its opinion score

OpinionFinder's Subjectivity Lexicon

The *Subjectivity Lexicon* [10] is compiled from manually annotated corpus *MPQA* which contains a wide variety of news articles. The words in the *Subjectivity Lexicon* have been labeled with part of speech tags as well as either strong subjective or weak subjective tags depending on reliability of the subjective nature of the word. We use only the strong subjective words in this task. The words in the *Subjectivity Lexicon* do not have opinion scores. Since this lexicon is constructed manually, we consider this lexicon is more reliable. So we assign *1.0* to each word as its opinion score.

Indicator

Following [9],we regard opinion indicator words such as '*would*', '*should*', as another significant clues for the author's opinion. We chose 9 indicators (e.g.' *would', 'could', 'pity', 'should', 'might', 'maybe', ' but', ' in fact', 'consequently'*) which can get higher precision on the data of TREC 2009. We simply assign *1.0* to each indicator as its opinion score.

Finally, we remove 1326 sentiment words of *SetntiWordNet, HowNet* and *OpinionFinder's Subjectivity Lexicon* which get lower precision on TREC 2009 data.

2. Opinion Scoring

For one blog post, the opinion score is computed as follows:

$$Score_{opin}(post) = w_{tf}(post, t) \times w_{op}(t)$$

Where $Score_{opin}(post)$ stands for the opinion score of a blog post, $w_{tf}(post, t)$ denotes the term frequency of opinion word t in the blog post, and $w_{op}(t)$ corresponds to the opinion score of word *t*.

Similar to 3.1.2, we also need to calculate the opinion/factual score of a blog feed. Two different strategies are used for computing the blog feed's opinion/factual score.

Average Strategy (AS)

Under this strategy, the opinion score of a blog feed is calculated as follows:

$$Score_{opin}(Feed) = \frac{\sum_{post \in Feed^{Top}} Score_{opin}(post)}{|Feed^{Top}|}$$

We simply compute the blog feed's factual score through the following equation.

$$Score_{fact}\,(Feed) = -Score_{opin}(Feed)$$

Maximum Strategy (MS)

Under this strategy, the opinion score of a blog feed is calculated as follows:

$$Score_{Opin}(Feed) = Max_{post \in Feed^{Top}}(post)$$

This essential idea comes from the IR domain where the most topic-relevant topic of a document is regarded as the topic that this document talks about. Thus，we regard the maximum opinion score of the posts as this feed's opinion score.

We also compute the blog feed's factual score through the following equation like the *Average Strategy*.

$$Score_{fact}(Feed) = -Score_{opin}(Feed)$$

Finally, we need to combine the blog feed's topic-relevant score and opinion/factual score to generate this feed's final ranking score. This ranking score should consider the topic-relevance and the opinion/factual facet inclination. It can be calculated as follows:

$$Score(Feed) \\ = Score_R(Feed)^{\mu} \times Score_{opin/fact}(Feed)^{1-\mu}$$

where μ is the parameter.

3. Opinion Retrieval Model (ORM)

In this section, we propose a novel opinion retrieval model. Following [12], the score of a blog post reflects not only the topic-relevance, but also the opinion/factual facet, and it can be formulated as follows:

$$Score(post|opin,Q) \\ \propto Score(post, opin, Q) \\ = p(post)p(Q|post)P(opin|Q, post)$$

$$Score(post|fact,Q) \\ \propto Score(post, fact, Q) \\ = p(post)p(Q|post)P(fact|Q, post) \\ = p(post)p(Q|post)(1 - P(opin|Q, post))$$

We can see two components in the above formula: $p(post)p(Q|post)$ which considers the topic-relevant degree and $P(opin|Q, post)$ which deals with its opinionated degree. Since the first component can be calculated through the classic language model, we only need to compute $P(opin|Q, post)$, and it can be calculated as follows:

$$P(opin|Q, post) = \sum\nolimits_{s \in Opin} co(s, Q|w)_f$$

Where *s* is any sentiment word in the above sentiment lexicon, $co(s, Q|w)_f$ is the frequency of the sentiment word *s* which is co-occurred with any query word of Q within a window of *W*.

Similar to 3.1.2, the blog feed's ranking score can be obtained through the following formula:

$$Score(Feed) \\ = (Max_{post \in Feed^{Top}}(Score(post|opin/fact, Q))) \\ \times \frac{|Feed^{Top}|}{|Feed|}$$

3.2.2. Personal vs. Official Model

For the personal/official task, we select three features to identify the personal/official facet: the existence of offensive words, the proportion of personal pronouns, and the maximum named entity proportion. Firstly, we believe that a blog where strongly offensive word appears is less likely to be an official one. For those feeds, the respective feeds are multiplied with a very large penalty multiplier; therefore they are less likely to appear in the top of the result list. Secondly, official blogs tend to use plural forms of personal pronouns, such as '*we*', '*our*' to refer to the organization, while personal blogs tend to use single forms of personal pronouns, for example '*I*'. We calculate the proportion of the two kinds of personal pronouns above as a feature. Finally, the most obvious feature of an official blog is the frequent appearance of one same named entity. Similar method has been previously used in [4]. We use *Stanford NER*[3] to tag the named entities and then sorted the proportion of all named entities, and then select the maximum one as another feature. We use a lower bound and an upper bound to the proportion value, a proportion *p* is set to the nearer bound if it exceeds the interval *(min-proportion, max-proportion)*.

Then, the facet score of a blog feed is formulated as follows:

$$C = \sum\nolimits_{post \in Feed^{Top}} Content(post)$$

$$Score_{official}(Feed) = \frac{1}{|Feed^{Top}|} \times NEC(C)^{x} PCX(C)^{y} PCP(C)^{1-x-y} \times Penalty$$

$$Score_{personal}(feed) = \frac{1}{|Feed^{Top}|} \times NEC(C)^{-x} PCX(C)^{-y} PCP(C)^{-(1-x-y)} \times \ Penalty$$

where *NEC* is the maximum named entity proportion, *PCS* is the proportion of singular forms of first personal pronouns, *PCP* is the proportion of plural forms of first personal pronouns, *x*, *y* are the parameters, and *Penalty* is directly proportional to the average numbers of strongly offensive words in each blog. We point out that smaller *penalty* is applied when calculate the personal facet. Finally, we used a similar method to combine the facet score and the topic-relevant score with in 3.2.1.

3.2.3. In-depth vs. Shallow Model

To establish the In-depth/shallow facet analysis model, we consider the proportion of regularly built words and the proportion of long words as our features to identify the in-depth/facet inclination. According to the word-building rules, most words used to describe simple stuffs and activities in daily life are short and irregular. To describe profound and abstract things, we often use those words built according to some special rules, such as those words ending with '*tion*', '*ous*' or '*ly*'. If a blog has a high proportion of this kind of words, it is very possible that this blog expresses '*in-depth*' topics rather than making simple descriptions. So we regard the proportion of these words as an important feature.

On the other hand, it is easy to make a hypothesis that longer words carry deeper and more complicated meanings. So we calculate the words which contain more than 8 letters and got the proportion as our second feature.

The in-depth score of one blog post is formulated as follows:

$$Score_{In-depth}(post) = PA \times \mu + PL \times (1 - \mu)$$

where *PA* represents the proportion of regularly built words, *PL* stands for the proportion of long words, and μ is the parameter.

Then, like the idea of 3.1.2, we also adopt two different strategies to calculate the in-depth/shallow facet score of the blog feed.

Average Strategy (AS)

$$Score_{In-depth}(Feed) = \frac{\sum_{post \in Feed^{Top}} Score_{In-depth}(post)}{|Feed^{Top}|}$$

$$Score_{shallow}(Feed) = -Score_{In-depth}(Feed)$$

Maximum Strategy (MS)

$$Score_{In-depth}(Feed) = \ Max_{post \in Feed^{Top}} Score_{In-depth}(post)$$

$$Score_{shallow}(Feed) = -Score_{In-depth}(Feed)$$

Finally, the idea of combining the topic-relevant score and in-depth/shallow score is also the same as in 3.2.1.

4. Result Analysis

In this section, we analyze the results of our approaches. Our approaches are evaluated on the new topics of TREC 2010. In the *baseline* sub-task, 46 new topics are

[3] http://nlp.stanford.edu/software/CRF-NER.shtml

evaluated and 31 new topics are evaluated in the *facet* inclination sub-tasks [7].

Table 1 provides the performance values of our own two baselines. We can see that the query expansion method applied on PKUTMB2 is effective. The little lower of *R-prec* value may due to that the method cannot improve the *precision* value which is the generic problem of most existing query expansion methods [5].

The opinion/factual results are shown in Table 2. We can see that for the opinion facet, the Average Strategy performs much better than the others. However, for the factual feat, the Maximum Strategy performs better, and our opinion/facet models are more suitable to the opinion sub-task than to the factual sub-task. We simply use the topic-relevant score of *stdbaseline* directly in this task. Maybe the different topic-relevant algorithms of *stdbaselines* result in the insignificant improvements over these *stdbaselines*. Besides, the insignificant improvements of ORM are possibly due to the limitation of the window size *W*.

Table 3 illustrates the personal/official results. In most cases, our result shows a higher MAP than the baselines, which proves the effectiveness of our method. However we can find obvious instability between different baselines. This also may be a result of the differences between topic-relevant algorithms. Those lower than the baseline results should be a consequence of randomization effect, which may come from the following aspects: the possibly inclusion of posts or lack of main blog text, the non-target effect of the algorithm, the limitation of standard tags, the simple multiply combination of parameters.

Table 4 provides the results of in-depth/shallow facet, which illustrates that our system works well on our own baseline. However, it seems not very effective on the *stdbaselines* like the above two facet sub-tasks. It reveals that our algorithm is effective but has some disadvantages as well. Besides, our system strongly related to several parameters and some of them needed to be changed in that case. The results prove that we don’t make reasonable change.

Tag	MAP	R-prec	bpref	P@10
PKUTMB1	0.2453	**0.2892**	0.2325	0.3304
PKUTMB2	**0.2537**	0.2882	**0.2403**	**0.3435**

Table 1: The performance of our baselines

Tag	Opinion MAP				Factual MAP			
	baseline	AS	MS	ORM	baseline	AS	MS	ORM
PKUTMB1	0.1761	**0.2807**	0.1804	0.1701	**0.2192**	0.1399	0.2148	0.1399
PKUTMB2	0.1619	**0.2758**	0.1740	0.1553	**0.2150**	0.1394	0.2124	0.1394
stdbaseline1	0.2598	**0.2608**	0.2603		0.2693	0.2705	**0.2761**	
stdbaseline2	0.1054	**0.1116**	0.1068		0.2068	**0.2081**	0.2069	
stdbaseline3	**0.0768**	0.0700	0.0723		0.1660	**0.2344**	0.1566	

Table 2: Opinion/Factual MAP results over five baselines.

Tag	Personal MAP		Official MAP	
	baseline	result	baseline	result
PKUTMB1	0.1470	**0.1636**	0.1820	**0.1930**
PKUTMB2	0.1441	**0.1901**	**0.1962**	0.1950
stdbaseline1	0.1377	**0.1575**	0.2439	**0.2507**
stdbaseline2	0.0755	**0.0856**	**0.1938**	0.1832
stdbaseline3	**0.0900**	0.0653	0.2014	**0.2127**

Table 3: Official/Personal MAP results over five baselines.

Tag	In-depth MAP			Shallow MAP		
	baseline	MS	AS	baseline	MS	AS
PKUTMB1	0.1644	**0.2407**	0.2398	0.1084	0.0874	**0.0973**
PKUTMB2	0.1533	**0.1733**	0.1729	**0.1005**	0.0966	0.0979
stdbaseline1	**0.2345**	0.0876	0.0876	**0.1038**	0.0416	0.0416
stdbaseline2	**0.1309**	0.0662	0.1066	**0.1259**	0.0528	0.0528
stdbaseline3	**0.0756**	0.0477	0.0477	**0.0923**	0.0372	0.0372

Table 4: In-depth/Shallow MAP results over five baselines.

5. Conclusion and Future Work

In this paper, we present the PKUTM system for the Faceted Blog Distillation Task. This task has been usually approached as a two-stage procedure consisting of *baseline* stage and identifying the facet inclination stage. In the *baseline* stage, an effective approach is proposed to extract useful words from the topics for query expansion which can improve the *recall* value. Regarding the facet inclination stage, several heuristic methods are used. The experimental results show these heuristic methods are effective. We also propose a novel opinion retrieval model for the opinion/factual facet inclination sub-task. Our system also has some weak points such as our facet models do not perform well over the *stdbaselines*. In the future, we will devote to explore models which are more robust.

6. Acknowledgements

This work was supported by NSFC (60873155), Beijing Nova Program (2008B03) and NCET (NCET-08-0006). We thank Lihong Zhao and Shanshan Huang for proofreading the paper.

References

[1] Z. Dong, HowNet. http://www.HowNet.org

[2] A. Esuli and F. Sebastiani. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings from International Conference on Language Resources and Evaluation* (*LREC*), 2006.

[3] Y. Lee, S-H. Na, J.K, SH. Nam, H-Y. Jung, J-H. Lee. KLE at TREC 2008 Blog Track: Blog Post and Feed Retrieval. In *Proceedings of TREC'08*, 2008

[4] S.Li, H. Gao, H.Sun, F. Chen, O.Feng, S.Gao,H.Zhang, X.Li, C.Tan, W.Xu, G.Chen, J.Guo.A Study of Faceted Blog Distillation-- PRIS at TREC 2009 Blog Track. In *Proceedings of TREC'09*, 2009

[5] C. Manning, P. Raghavan, and H. Schütze Introduction to Information Retrieval. Cambridge University Press, 2008.

[6] D. Metzler, T. Strohman, H. Turtle, and W. Croft. Indri at TREC 2004: Terabyte track. In *Proceeding of the 2004 Text Retrieval Conf*, 2004.

[7] I. Ounis, C. Macdonald, and I. Soboroff. Overview of the TREC-2010 Blog Track. In *Proceedings of TREC'10*, 2010.(to appear)

[8] I. Ounis, M. deRijke, C. Macdonald, G. Mishne, I. Soboroff.Overview of TREC-2006 Blog track. In *Proceedings of TREC'06*, 2006.

[9] C. Wang, T. Ma, L. Guo,X. Wan and J. Yang. PKUTM Experiments in NTCIR-8.In *Proceedings of NTCIR-8 Workshop Meeting*, 2010

[10] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of EMNLP'05*, 2005 .

[11] X. Xu,Y. Liu,H. Xu, X. Yu,L. Song, F. Guan, Z. Peng,X.Cheng.ICTNET at Blog Track TREC 2009.In *Proceedings of TREC'09*, 2009

[12] M. Zhang and X. Ye. A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval . In *Proceedings of SIGIR'08*, 2008