NLPR in TREC 2007 Blog Track

LIU Kang, WANG Gen, HAN Xianpei, ZHAO Jun
National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing, China, 100080

Abstract

This paper describes the opinion retrieval system for TREC 2007 blog track. This paper focuses on two components of the system. One component is important content block detection component which is used to extract blog contents and get rid of noises in blog pages. Another component is opinion retrieval component which is used to give each sentence an opinion score and combine it with topic score based on SVR. The evaluation proves the validity of our algorithm in the task.

1 Introduction

The Blog track had two tasks in the TREC 2007. We participate in the opinion retrieval task, which is the same as the task of TREC 2006. This task is to identify and rank opinionated blog posts for the given topic. There were 50 topics with Title, Description and Narrative terms.

Our system divides the opinion retrieval task into four main steps.

The first step is important contents block detection, which is used to clean noises in blog pages. Our system segments pages into several blocks with different sizes based on layout. Then we extract some special features and use SVM classifier to detect important content from blog pages and discard unimportant block that was filled with noises. This step can remove noises and retain title, post and comments in blog pages. In the second step, we build index using Indri toolkits [2] and create query according to the "topic" fields provided by NIST. In the third step, we have two components, one is retrieval model including sentences retrieval model and document retrieval model. Using document retrieval, we can obtain topic relevant score for each document. Another component is opinion scoring model. In this component, we employ NB regression method to obtain opinion scores for each sentence output from the sentence retrieval model. In the last step, we treat the answer of TREC 2006 Blog track as training corpus and employ SVR to train a fusion model. By using this model, our system can integrate the topic relevant score and opinion score together. Finally, we re-sort the result of document retrieval. Top 1000 documents for each query are submitted.

Figure 1 gives the frame of our system. We will introduce each step in detail in the following sections. Section 2 will introduce the important content block detection method and the query formulation process. In section 3, we will describe the retrieval module including document retrieval, sentence retrieval and fusion method between topic score and opinion score. Section 4 will analyze the results of the six submitted runs and give a conclusion.

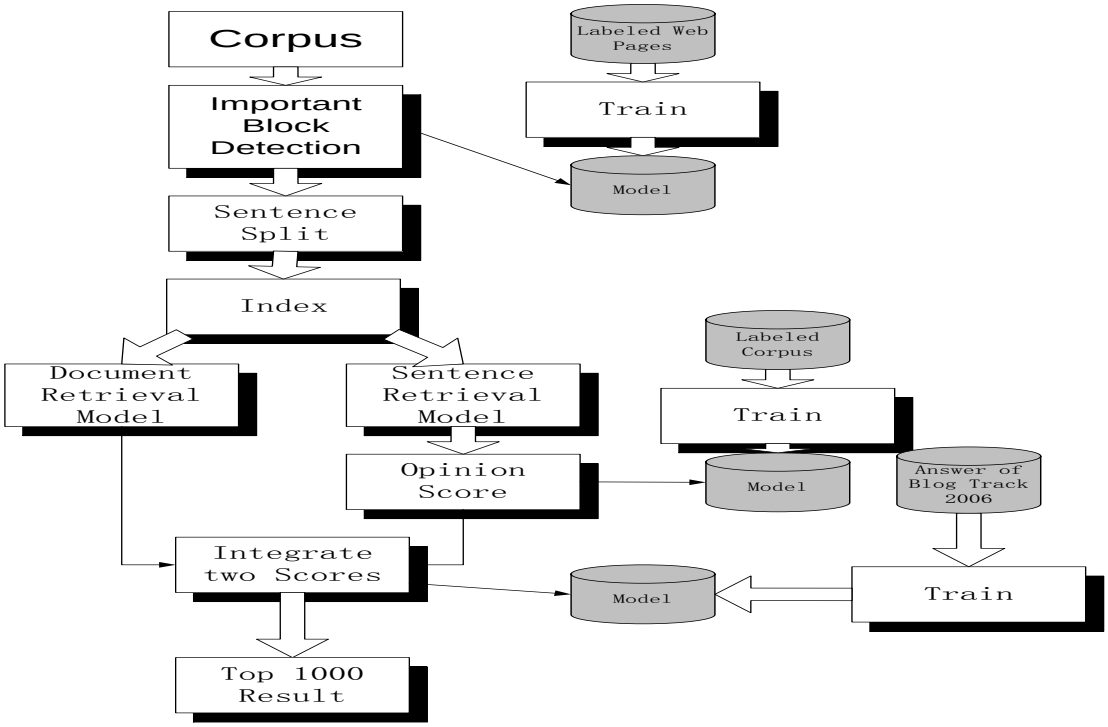

Figure 1. Overall View of Our System

2 Data Preprocessing

Before indexing, we must preprocess the corpus. This step includes data cleaning, sentence splitting, transforming pages into regular TREC text format and query processing. In data cleaning step, we propose a novel denoising method named as ***important content block detection***. It is based on page segmentation. In the following, we will dive a description to the method.

2.1 Problem of Blog Page Cleaning

Data cleaning is a very important preprocessing step because the corpus comes from online webpages with HLTM format, which contains many noises. These noises will reduce the quality of the retrieval result. However, blog pages are different to other pages (navigational pages, sell pages, etc) in aspect of page layout. When we need to find some opinion about an entity, we often focus on blog's post, title, and its comments. Other parts in blog pages, such as links to previous posts, advertise, navigational links, some web special flags, can be treated as noise or unimportant contents. The presence of these parts may reduce retrieval quality. So our cleaning task is to extract important contents of blog pages and reduce pages noises.

However, the layout of blog pages is too simple compared to navigational pages and sell pages. Important contents in blog pages are centralized in isolated blocks. From Figure 2, we can see that the important content of a blog page, like title, post, comment, is set in the yellow region. Many links titled by 'Previous Posts' are centralized in the right side of pages. So our method for extracting important contents in blog pages has two steps. In the first step, we segment one page into several visual isolated blocks. It made blog's contents and noises to be integrated into isolated blocks respectively. In the second step, we use SVM classifier based on layout features and language features to identity important content blocks from the output of step one.

2.2 Important Content Block Detection

2.2.1 Blog Page Segmentation

The first step in important content block detection method is page segmentation. The segmentation method used in our system is based on VIPS [1][3]. There are several methods which are based on DOM tree [4][5][6]. These methods use DOM tree parsed from HTML file to find important tags and use these important tags to segment pages. But pages have many visual structures, especially in blog pages, see Figure 2. DOM tree can reveal only tag structure rather than these visual layouts information. VIPS can make full use of the visual features of pages such as font, color and size. Figure 2 gives a segmentation result for a blog page. We can see that the whole page is segmented into several individual blocks. In each block, content is almost unified. After segmentation, a block tree is built for each page. Each block except leaf node has a parent block node and several child block nodes. From each node in this block tree, we can extract some features including layout features and language features. By using SVM classifier and some heuristic rule, we can identify whether each node being content or noise. When we use VIPS method, we can set the value of segmentation level (pDoc). In our systems, we set this value was 8.

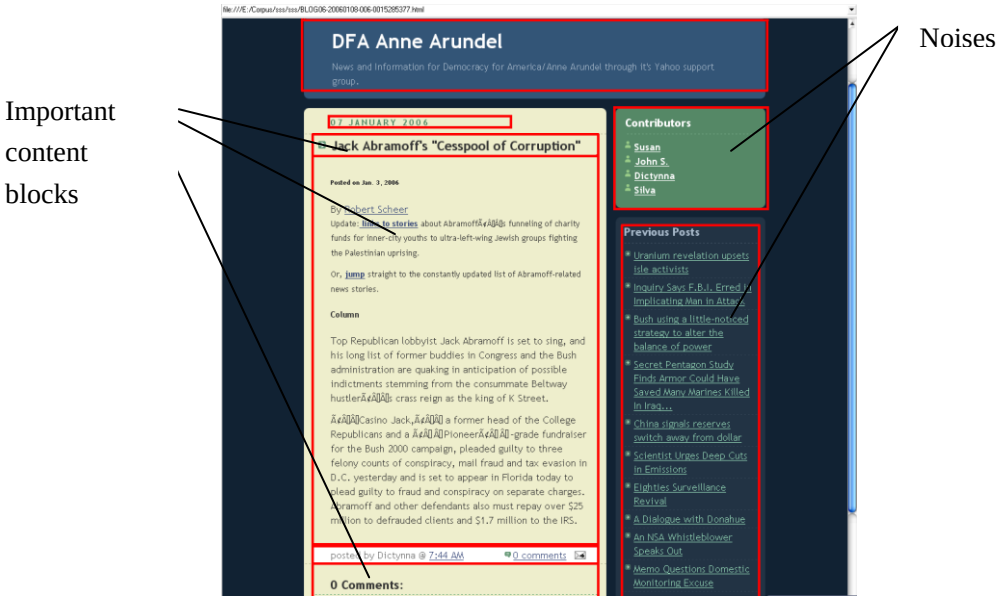

Figure 2. Segmentation of a sample blog page

2.2.2 Important Content Block Detection and the Experiments

This task is treated as a two-class classification problem. One class is important content blocks and the other is noise blocks. Noise blocks should be discarded and important content blocks should be retained. We extract some layout features for SVM classifier including spatial features, tag features, content features and so on. We also use language features based on unigram model. The important features are listed in Table 1. To the block tree which is built after segmentation, we begin from leaf node to root, using SVM classifier and heuristic rule to decide whether each node is content or noise.

We randomly extract 500 pages from blog corpus and labeled them manually. Then we get 1836 noise blocks and 3576 important content blocks. For testing the validation of our method, we select 70% labeled pages to train model and 30% for testing. Our aim is to get tradeoff results between precision and recall. We aim at getting better precision results on the condition of higher recall value of important content block. So the main contents of blogs will be retained for indexing. Important contents of blogs will not be lost after data cleaning. The experimental results are listed in Table 2.

We also select 5 sites from testing data to test the robust of our method, because Blog pages in different site have the different layout. Table 3 also lists experiments results for each site.

Block Width/Page Width	Block Height/Page Height
Block Center Y/Page Height	Block Center X/Page Width
pDoc value in VIPS	Text number in one block
Link number in one block	Link Number/Token Number
Image number in one block	Number of Sub-block
Block depth in Block Tree	Block Tree Depth
<FORM> tag number in one block	<P> tag number in one block

Table 1. Some selected layout features

Results	Only layout features	Combined features
Precision	0.8031	0.85
Recall of Important Content Block	0.913	0.908

Table 2. Results of Important Content Block Detection

Site	Precision	Recall of Important Content Block
Northfield.org	0.9950	0.9890
mdcbowen.org	0.9490	0.9930
gutRumbles.com	0.9650	0.99
asktaxmoms.com	0.9280	0.8860
scotsman.com	0.7170	0.9950

Table 3. Results of Important Content Block Detection for Different Site

From Table 2, we can see that our method based on combined features can extract important content block with high accuracy. The method combined with two kinds of features has higher precision than the method using only layout features. The results for different sites testified our method's robustness. We have high recall of detecting important content block for different layout pages. That means that important contents are mainly retained after data cleaning. Because of the diversity of blog pages, the precision of some pages is not high like site 'scotsman.com'. But our first target is to ensure high recall of important content block class. So this case can be ignored and our method can get effective results.

2.3 Format Transform

After data cleaning for blog page files, we need conduct sentence splitting and format transforming. We only retain texts in content blocks and removed images, tags, scripts and stylesheets from content blocks. Then we transform each blog pages into the following format. means paragraph and <s> means sentences. <title> means the title of the blog.

<DOC>
<DOCNO> file number </DOCNO>
<TEXT>
<title>blah blah blah</title>

<s>blah blah blah</s>
<s>blah blah blah</s>
…..

….
</TEXT>
</DOC>

2.4 Query Formulation

The TREC Blog track provided 50 topic including Title, Description and Narrative terms. We use Title, Title and Description to build queries respectively. The tagger in [8] is used to identify POS for each term. We only use all token in Title and nouns and adjectives in Description to create query. The word 'opinion' in the Description terms is removed. We do not use any dictionary and corpus for query expansion. The reason is that our focus is mainly put on the data cleaning and fusion between topic score and opinion score. For retrieval component, because document retrieval and sentence retrieval are conducted respectively, our system need create two kinds of queries. Different queries will be created based on different formats. A sample of queries format is listed in the following. Because sentence retrieval is for searching topic relevant sentences, it would be processed in '<s>' field

- #combine(lactose gas)
- #combine(#combine(lactose gas) #combine(#2(lactose gas) symptoms remedies)
- #combine[s] (#weight(0.7 #2(lactose gas) 0.3#combine(lactose gas))
- #combine[s](#combine(lactose gas) #combine(#2(lactose gas) symptoms remedies)

3 Index and Retrieval

3.1 Document Retrieval and Sentence Retrieval

We use Indri toolkit to build two kinds of index. One index is built based on the output of important content block detecting. The file format is set be 'trectext'. Another index is built based on original blog corpus, file format is 'trecweb'. We do not use stemmer and stoplist when indexing.

After document retrieval, we can get a topic relevant score for each document. On the other side, we can obtain topic relevant sentences through sentence retrieval. We set the number of returned documents being 1000. The target of sentences retrieval is to get enough relevant sentences. So the number of returned sentences is set 10000. For the sentence retrieval component, we use two kinds of methods. One is based on passage retrieval in Indri. Each document texts are dynamically split into fixed length sentences (sentences length was set 100). Another method is based on sentence splitting toolkit. Each sentence length is not unified and sentence retrieval is processed on <s> domain.

3.2 Opinion Score

In the step, Our system could give each sentence returned by sentences retrieval a opinion score by using Naïve Bayes regression method. We use resources clawed from web to train a NB regression model. Then we can obtain the opinion score for each relevant sentence by using it.

Training corpus was lacked because of the diversity of topical domain, though Pang [9] and Bing Liu [10] could provide limited subjective corpus respectively. The Corpus of Pang is movie reviews containing 5000 subjective and 5000 objective sentences. The corpus of Bing Liu is product reviews containing 4258 subjective sentences. We need more resources that cover more broad range of topic and can provide amount of subjective texts. [7] is a review site. Reviews can cover a lot of fields including restaurant, film, sport and other events. So we download reviews on this site [7] and treated all of them as subjective texts. We also use TREC-AP news corpus as objective texts. Altogether, we obtain 85,914 subjective sentences and 362,936 objective sentences. We use unigram as the feature types, and train Naïve Bayes classifier to score each sentence.

When training, we use stemming tool and get opinion score of each word in training data. For each retrieval sentence, we use the following formula to obtain opinion score.

$$Score(Sentence) = \sum_{Word \in Sentence} Score(Word)$$

3.3 Fusion

After document retrieval and sentence retrieval, our system obtains topic relevant score for each document and opinion score for each topic relevant sentences. The fusion component integrates two kinds of scores together. In last year, several groups employed many variations of the weighted sum formula. The weight of each part was set by experience or estimated by train data.

In our systems, we use SVR to integrate the topic relevant score and the opinion score. The training corpus comes from TREC 2006 blog track answer. We extract some special features for SVR. The features are listed in the Table 4. Because there are not one topic relevant sentence corresponding to a retrieval document, we employ average, sum, highest opinion score of sentences in retrieval document as feature for SVR.

In answer of TREC 2006 blog track, if one document is topic relevant but not expressing any opinion, it will obtain 1 point. If one document is topic relevant and expresses opinion on entity, this document will obtain 2 point, which means that this document contains subjective texts about denoted entity. The value of each document in answer text will be used for training. If the document output from the document retrieval process does not contain any sentence that was topic relevant for the query, the document will obtain θ point. In our system, θ equals 0.8.

The rank in the document retrieval result	The score in the document retrieval result
Average opinion score of retrieval sentence in one retrieval document	Sum opinion score of retrieval sentence in one retrieval document
Highest opinion score of retrieval sentence in one retrieval document	

Table 4. Feature List for Fusion Component

4 Submission and Results

TREC 2007 blog track contains 50 topics. For each submission, MAP, R-precision, bpref and P@10 were four main evaluation methods. We submit six automatic runs. In five runs, we use important content block detection component and one run not. We also employ different sentence split methods in six runs. In three runs, we use Title terms to build query. In other three runs, we use Title and Description terms to create query. Table 5 is a description of the six runs in detail. The evaluation results of the six runs are given in Table 6

Run	Description
NLPRPST	Important Content Block Detection + Title + Sentence Splitting + θ=0.8
NLPRPT	Important Content Block Detection + Title + Fixed Sentence Splitting + θ=0.8
NLPRPTONLY	Important Content Block Detection + Title + No Opinion Score Component
NLPRPTD1	Important Content Block Detection + Title_Description + Fixed Sentence Splitting + θ=1
NLPRPTD2	Important Content Block Detection + Title_Description + Fixed Sentence Splitting + θ=0.8
NLPRTD	Title_Description + Fixed Sentence Splitting + θ=0.8

Table 5. Description of Our Runs

Runs	MAP	R-precision	bpref	P@10
NLPRPST	0.2542	**0.3168**	0.2945	**0.4620**
NLPRPT	0.2476	0.2973	**0.3018**	0.4340
NLPRPTONLY	0.2506	**0.3166**	0.2917	0.4520
NLPRPTD1	0.2532	0.3036	0.2929	0.43
NLPRPTD2	**0.2587**	0.3088	0.2956	0.4560
NLPRTD	0.2462	0.2935	0.2723	**0.472**

Table 6. Result of Our Runs

From the experimental results shown in Table 6, we can get the following conclusion.

- Comparing the results of NLPRTD and NLPRPTD2, we can see that our data preprocessing component can get rid of noises effectively. After important content block detection, MAP value can be improved. We can also see that the MAP values of all runs with important

content block detection are higher than those of the runs without that component. So it proves the validity of the "important content block" method.

- Comparing the results of NLPRPTONLY and NLPRPST, we can see that all evaluation values Are improved slightly after opinion scoring and fusion. It means that our opinion score and fusion method is effective.
- Comparing the results of NLPRPT and NLPRPST, we can see that performance with opinion score based on fixed length sentences is worse than that based on sentence splitting method. The performance of NLPRPT is even worse than NLPRPTONLY, i.e. the run without opinion component. But the result which we test in TREC 2006 block track was reverse.
- Comparing the results of NLPRPTD1 and NLPRPTD2, we can see that performance with θ=0.8 is better than θ=1. That is in our expectation.
- We also find that query built based on Title and Description can not performance much better than Title only. It is out of our expectation. In last year block track, the performance of the query based on Title, Description is much better than Title only. The reason may be that nouns and adjectives in Description term bring some noises into query.

5 Conclusion

This paper describes our opinion retrieval system in detail for TREC 2007 blog track. Our system has two important components. The first important component is important content block detection. In this component, our system uses VIPS method to segment blog pages and used SVM classifier to extract important content in blog pages. Layout features and language features are employed. In the second important component, our system employs NB regression to obtain opinion score for each relevant sentence. And it uses SVR to integrate topic score and opinion score. Our six submitted runs performed not the best, but better than the median of all the submitted runs. The evaluation of our six runs shows the validity of two important components in our system.

6 Acknowledge

This work is supported by the National Sciences Foundation of China under grant No. 60673042 and National Sciences Foundation of Beijing under grant No. 4052027, 40734043.

7 Reference

[1] Deng Cai, Shipeng Yu, Ji-Rong Wen and WeiYing Ma. VIPS: A Vision based Page Segmentation Algorithm. MSR-TR-2003-79. 2003.

[2] http://www.lemurproject.org

[3] Ruihua Song, Haifeng Liu, Ji-Rong Wen, Wei-Ying Ma. Learning Block Importance Models for Web Pages. 13th International WWW Conference, 2005.

[4] Suhit Gupta, Gail Kaiser, David Neistadt, Peter Grimm. DOM-based Content Extraction of HTML Documents. The Twelfth International World Wide Web Conference, 2003.

[5] Lan Yi, Bing Liu, Xiaoli Li. Eliminating Noisy Information in Web Pages for Data Mining.

The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.

[6] Lan Yi, Bing Liu. Web page cleaning for web mining through feature weighting. In proceedings of Eighteenth International Joint Conference on Artificial Intelligence, 2003.

[7] http://www.yelp.com

[8] http://www.cs.jhu.edu/~brill/

[9] http://www.cs.conell.edu/People/pabo/movie-review-data/

[10] http://www.cs.uic.edu/~liub/FBS/FBS.html