UIC at TREC 2007 Blog Track

Wei Zhang
Department of Computer Science
University of Illinois at Chicago
wzhang@cs.uic.edu

Clement Yu
Department of Computer Science
University of Illinois at Chicago
yu@cs.uic.edu

ABSTRACT

In TREC 2007 Blog Track, we developed a three-step algorithm for the opinion retrieval task. An information retrieval step retrieves the query-relevant documents. A following opinion identification step identifies the opinionative texts in these documents. A ranking step identifies the query-related opinions in the documents and ranks them by calculating their opinion similarity scores. For the polarity task, our strategy is to find the positive and negative documents respectively, and then find the mixed opinionative documents in the intersection of the positive and negative document sets. We implemented our opinion retrieval algorithm in two special cases, one to retrieve the positive documents, and the other to retrieve the negative documents. A judging function labeled a subset of the documents, which were in the intersection of the positive and negative documents, as the mixed opinionative documents. We studied two parameters in our opinion retrieval algorithm, each of which had two values to compare. This resulted in four submitted opinion retrieval runs and their corresponding polarity runs.

1. INTRODUCTION

The opinion retrieval task was introduced in the TREC 2006 Blog Track [6]. In opinion retrieval, a relevant document must have query-related opinions, regardless of the orientation of the opinions. Our opinion retrieval algorithm is a classification-based algorithm. It is developed based on our TREC 2006 Blog system [11]. We consider the opinion retrieval as a three-step procedure. The first step is an information retrieval (IR) step that retrieves the documents relevant to the query topics. We apply concept (phrase) identification, query expansion, phrase similarity calculation and document filtering techniques to improve the retrieval effectiveness. The document filtering is a new component in our 2007 system. The second step is an opinion identification step that finds the opinionative texts in the documents. This is a text classification process. The chi-square test is applied to the training data to select the features, which are used to build a support vector machine (SVM) classifier. This classifier tests all the sentences of a document. Each sentence receives either a subjective label or an objective label. We say a document is opinionative if it has at least one subjective sentence. This year we use a single query-independent classifier to replace all the query-dependent classifiers in our 2006 system. The final step is the ranking step. It locates the query-relevant opinions in the opinionative documents, and uses them to calculate the documents' opinion similarity scores. In this step, we designed new opinion similarity functions.

The polarity task is a new task in 2007. It asks a system to identify the orientation (polarity) of the opinions in a retrieved opinionative document. The label could be positive, negative and mixed. Our strategy is that, after our opinion retrieval system retrieves the opinionative documents, we identify the documents having query-related positive and negative opinions respectively. The documents having mixed opinions should be a proper subset of the intersection of these two sets. We construct two special opinion retrieval systems, one to retrieve the positive documents, and the other one to retrieve the negative documents. A judging function determines if a document retrieved by both the positive and the negative systems can be labeled as mixed.

The paper is organized as follows. Section 2 describes the IR module of our opinion retrieval system. Section 3 describes the opinion identification module. The ranking module is described in Section 4. The polarity classification system is described in Section 5. Section 6 explains our submitted runs. Conclusions are given in Section 7.

2. TOPIC RETRIEVAL

The overall structure of our opinion retrieval system is shown in Figure 1, where the topic retrieval module has the components of the concept (phrase) identification, query expansion, concept similarity based retrieval and document filter. All components other than the filter component had been used in our 2006 system. The new filter component acts as a post-processing procedure to remove potential spam documents.

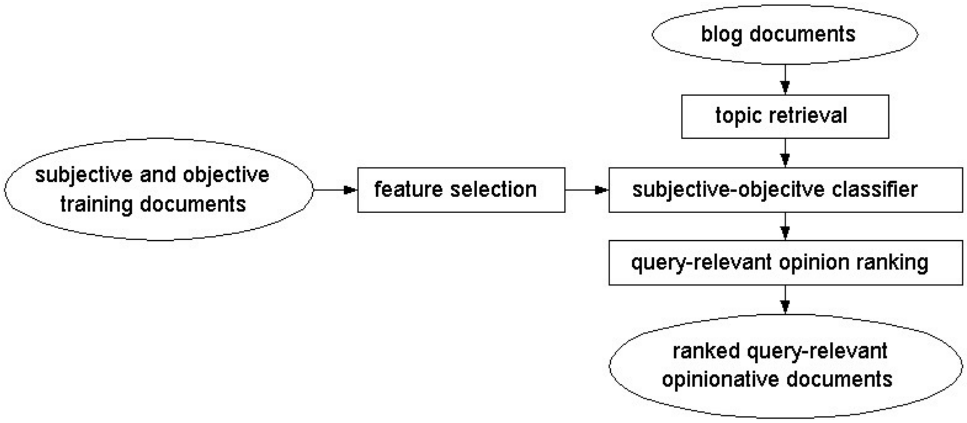
blog documents
topic retrieval
subjective and objective training documents
feature selection
subjective-objecitve classifier
query-relevant opinion ranking
ranked query-relevant opinionative documents

Figure 1. The structure of the opinion retrieval system.

2.1 Concept Identification

We define the concepts in a query as the phrases or single words that denote entities. A query may contain multiple concepts. For example, a query "opera software or opera browser" contains three concepts "opera software", "opera browser" and the query itself. We identify the concepts in a query to feed them to our document search engine, as it needs to calculate the concept similarity. We defined four types of concepts: proper nouns, dictionary phrases, simple phrases and complex phrases. The proper nouns are the noun phrases referring to people, places, events, organizations, or other particular things. A dictionary phrase is a phrase that has an entry in a dictionary. Proper noun can be considered as a special type of dictionary phrase. A simple phrase is a 2-word noun phrase, which is grammatically valid but does not have a dictionary entry, e.g. "small car". A complex phrase is similar to simple phrase but has 3 or more words. We developed an algorithm that combines several tools to identify the concepts in a query. We use Minipar [9], WordNet [8], and Wikipedia [3] for proper noun and dictionary phrase identification. Collins Parser is used to find the simple phrase and complex phrase. Web search engine (Google) is used to provide statistical information for phrase verification and selection purpose. The details of the algorithm can be found in [12].

2.2 Query Expansion

Query expansion aims to add a certain number of query-relevant terms to the original query, in order to improve retrieval effectiveness. We adopt three query expansion methods. The first method utilizes the online dictionary Wikipedia to find an entry page for a concept in a query. If such entry exists, the title of the entry page is expanded as synonym of the concept. The synonyms are treated the same as the original query terms in the retrieval. In addition, the content words in the entry page are ranked by their in-page frequencies. The top k terms are returned as potential expanded terms. The local pseudo feedback [1] is our second expansion method. The original query is used to retrieve n ranked documents, where the top k terms that are highly correlated to the query are returned as the expanded words. The assumption is that the top ranked retrieved documents should be relevant to the query, and the terms that co-occur with the query terms should be related to the query. The third query expansion method is Web-based. It is similar to the second method but using the Web as the document collection. The query is submitted to a Web search engine, such as Google, which returns a ranked list of documents. In the top m documents, the top k terms that are highly correlated to the query terms are returned. All the expanded terms, from different methods, of all the concepts of a query are put together. If an expanded term is returned by two or more methods, its weight is the sum of the individual weights from the different methods.

The weights of the expanded terms are normalized to values in (0, 0.7]. The original query terms and concepts all have weights of 1, while the expanded terms should have lower weights. The 20 top weighted expanded terms are chosen as the final expanded terms for a query.

2.3 Concept-Based Topic Retrieval

After concepts identification and query expansion, an original query will be expanded with a list of concepts and their synonyms (if exists) and a list of expanded words. In our topic retrieval module, the query-document similarity consists of two parts: the concept similarity and the term similarity (*concept-sim, term-sim*). The *concept-sim* is computed based on the identified concepts in common between the query and the document. The *term-sim* is the usual term similarity between the document and the query using the *Okapi* formula [7]. Each query term that appears in the document contributes to the term similarity, irrespective of whether it occurs in a concept or not. The *concept-sim* has a higher priority than the *term-sim*, since we emphasize that the concept is more important than individual terms. Consider, for a given query, two documents d_1 and d_2 having similarities (x_1, y_1) and (x_2, y_2), respectively. d_1 will be ranked higher than *d2* if either (1) $x_1 > x_2$, or (2) $x_1 = x_2$ and $y_1 > y_2$. Note that if $x_i>0$, then the individual terms which contribute to *concept-sim* will ensure that $y_i>0$. The calculation of *concept-sim* is described in [5].

2.4 Document Filtering

In this year, we add a filtering component to the topic retrieval module. It acts as a post-processing step to remove potential spam documents, which could get high similarity scores. We adopt three simple filtering rules. The first rule removes any document that contains a sentence of 300-or-more words, as one type of the spams simply puts a large amount of words in a document, so it is retrievable by many queries. We chose the threshold of "300 words" intuitively without a thorough study. The second rule removes any document that contains at least two of the three words of "nude", "naked" and "sex", while the total number of these words should be no less than 10. We hope that it could help us remove those offensive spams. Again, the feature words and the threshold "10" were chosen intuitively. The third rule removes documents written in foreign languages. We count the frequencies of some common English words and foreign words. If the English word frequency is smaller than a threshold, and the foreign word frequency is greater than the threshold, we consider the document as written in the foreign language, and then discard it. These three rules are set without using complicated algorithms. Clearly they can be refined later on.

3. OPINION IDENTIFICATION

In this step, we try to detect all the opinions in a document, which is returned from the topic retrieval. The opinions can be either related or irrelevant to the query. The next ranking component will connect the opinions to a query. In general, both query-dependent and query-independent training data are collected. Subjective and objective features are selected. A single query-independent classifier is built by using these features, and applied to the documents, to label the opinionative contents.

3.1 Feature Selection by Using Query-Relevant Training Data

A subset of the features comes from the subjective and objective training data related to the queries. For each of the 50 TREC 2007 Blog queries, the query-related subjective training data is collected from review Web sites and general opinionative Web pages. A query is searched in Rateitall.com. Once the entry is found, the reviews are collected. The reviews from other sibling nodes of the entry node are also collected in order to get enough amount of training data. The site epinions.com is added as a new data source to collect query-related reviews too. A small set of "opinion indication phrases", such as "I think", "I don't think", "I like" and "I don't like", are used together with the query to collect opinionative Web pages. Each such phrase is submitted to a search engine with the query. The top ranked documents are collected as query-related review documents. To obtain the objective training data, we submit a query to Wikipedia. If there is an entry page, the whole page is collected as the objective data. The query's sibling nodes from Rateitall.com are also searched in Wikipedia to collect more objective data. The details of this training data collecting procedure can be found in [11].

After the 50 queries' subjective and the objective training data are ready, all the subjective data are placed in a set, and all the objective data are placed in another set. We adopt the Chi-square test [2] as the feature selection method to find the unigram and bigram features from these two sets. The chi-square value of a qualified feature must be no less than the threshold of 5.02, which corresponds to the significance level of 0.025. We assume a feature like this is class dependent, and thus can be used as a feature in the SVM classifier.

3.2 Feature Selection by Using Query-Independent Training Data

We have found that using more features improves the opinion retrieval effectiveness [10]. In order to collect more good features, we collect all the leaf nodes and their reviews from rateitall.com to construct a very large query-independent subjective data set (over 10 thousand topics). The rateitall.com has all the topics organized in a tree structure. The leaves are the specific topics, such as "Chicago Bulls". The non-leaf nodes refer to the more general categories, such as "basketball team", "basketball", "sports", etc. Upon collecting the reviews, we also record the scores of these reviews. For example, score 5 means the most positive review, while 0 or 1 mean the most negative review. For each topic on the leaves, epinoin.com reviews with their positive/negative labels are also collected if available. Possible Wikipedia entries for each of these topics are also collected. All the subjective reviews with their scores are put into a set. All the objective reviews are put into another set.

We design a method to only extract subjective features from this training set, because our concern is that we may not have enough subjective features. We extract all the positive reviews (scores 4 or 5) and all the negative reviews (score 0, 1 or 2) from the large training set respectively. These two sets are used in the Chi-square test. The features should be either positive-oriented or negative oriented, and they all should be the subjective features. This time, the Chi-square threshold is set to 10, which corresponds to the significance level of 0.0016.

3.3 The SVM Opinion Classifier

The subjective features from Section 3.1 and 3.2 are merged into a subjective feature set. The objective features from Section 3.1 form the objective feature set. We build a single query-independent opinion classifier by using these features. All the subjective training data in Section 3.1, the positive/negative data in Section 3.2, and the objective training data are converted to the vector representation of the features. Then we use the support vector machine (SVM) [4] learning program to train a classifier by using the vector data. We only build one classifier this year, because we have found that the query-independent and the query-dependent classifiers usually perform at the same level [10]. Clearly the former simplifies the system structure. When using the classifier, a document is split into a list of sentences. Each sentence is converted to a vector of the features. The classifier takes the sentence vector as the input, and outputs a label (subjective or objective) and an associated score. A subjective sentence gets a positive score while an objective sentence gets a negative score. The score represents the confidence level of the classifier to this answer. Larger absolute score (toward infinity) means higher confidence. A score close to 0 means low confidence. We define that a document is subjective (opinionative) if it has at least one sentence labeled as subjective.

4. OPINIONATIVE DOCUMENT RANKING

The opinions in the documents have been identified. But we need to find those that are actually related to the queries. This is done by using the text window method developed in our 2006 system. When there is a subjective sentence, we get two sentences prior to it and two sentences following it to form a 5-sentence window. We then search the original query terms and the expanded query terms within this window. If certain restrictions are met, this subjective sentence is labeled as a *relevant opinionative sentence* (ROS). Otherwise it is discarded. A document having at least one ROS is said to be a *relevant opinionative document* (ROD) of the query topic. A ranked list of RODs is the output of our opinion retrieval system.

We adopt the new query-document opinion similarity functions from [10]. Let Q denote a query. Let D denote a the ROD set of Q. Let d be a document in D. Let ROS(d, Q) denote the ROS set in d for the query Q. Let s denote a sentence in ROS(d, Q). Let Sim(d, Q) denote the topic retrieval similarity score from Section 2. Let OSim(d, Q) denote the query-document opinion similarity. We have the following basic opinion similarity functions:

$$OSim_{ir}(d,Q) = Sim(d,Q) \quad (4.1)$$

$$OSim_{stcs}(d,Q) = \sum_{s \in ROS(d,Q)} score(s) \quad (4.2)$$

$$OSim_{stcc}(d,Q) = |ROS(d,Q)| \quad (4.3)$$

Function (4.1) and (4.2) had been used in our 2006 system. Function (4.1) uses the topic retrieval score directly as the opinion similarity score. It is based on the assumption that the blog tends to contain the opinions because of its nature. Higher topic relevancy should result in higher chance of the existence of the query-relevant opinions. Function 4.2 uses the sum of the scores of the ROSs of d as d's opinion similarity. These scores come from the SVM classifier. Function (4.3) uses the size of the ROS as d's opinion similarity. Function (4.3) and (4.2) are similar to each other but emphasize on different aspects.

Our final opinion similarity functions are two combined functions. Function (4.4) is a linear combination of function (4.1) and (4.2). Function (4.5) is a linear combination of function (4.1) and (4.3). They represent the idea that both the objective topic information and the subjective opinion should contribute to the opinion similarity score. In both formula (4.4) and (4.5), the two opinion similarities are normalized to be in the range of [0, 1] before the summation. We set both the coefficients of *a* and *b* to 0.5, since experimental results shows that the best scores usually appear when *a* and *b* are in the range of [0.3, 0.7].

$$OSim_{ir\_stcs} = a \times OSim_{ir} + (1-a) \times OSim_{stcs} \quad (4.4)$$

$$OSim_{ir\_stcc} = b \times OSim_{ir} + (1-b) \times OSim_{stcc} \quad (4.5)$$

5. POLARITY CLASSIFICATION

The polarity classification is the new task in this year's Blog Track. The polarity task asks a system to label every retrieved subjective document as positive, negative or mixed. We consider this as a text classification problem, in which there are two base categories, i.e. the positive documents and the negative documents, while we consider the mixed opinions a union of the two base categories. So our strategy is to identify the positive and the negative query-relevant opinions in a document respectively, and then any document having both kinds of opinions should be checked if the opinions are in a mixed status. Figure 2 shows the structure of our polarity classification system.

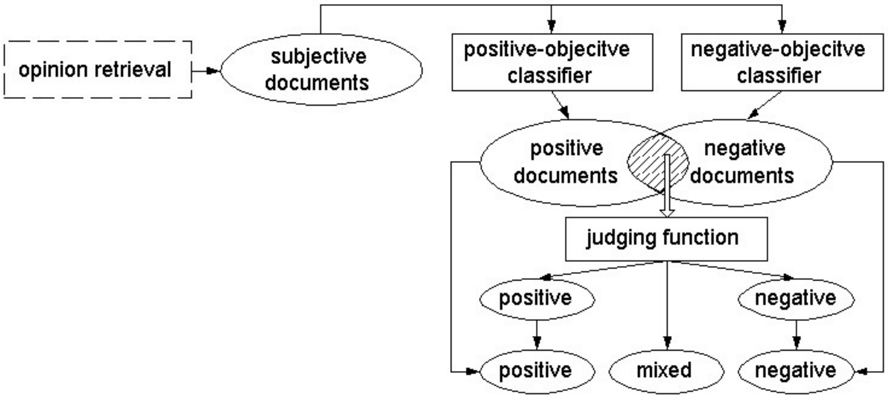

Figure 2. The structure of the polarity classification system.

5.1 Positive/Negative Opinion Identification

In order to find the positive opinions and the negative opinions respectively in the retrieved documents, we build two opinion retrieval systems. So that one system is able to identify the positive opinions, while the other one can identify the negative opinions. These two systems can be considered as special versions of our general opinion retrieval system. We use the positive opinion retrieval system for illustration. As mentioned in Section 3.2, we have collected the scored reviews for over 10 thousand topics. We pick the reviews having scores of 4 or 5 to form a “positive training set”, because we think that a score of 4 or 5 usually indicates the strong positive orientation in a review. We also pick their corresponding objective documents from the objective data set to form the “objective training set”. These two sets are used for the Chi-square feature selection to get the positive and the objective features, which are then used to build a positive-objective SVM classifier. This is the same procedure that we have followed to build the opinion retrieval system. The only difference is that the positive training data replaces the general subjective training data. This system can identify the query-relevant positive opinions in the documents. For the negative opinion retrieval system, we simply changed the positive training set to a negative training set, which contained the reviews with scores of 0 or 1. After these two systems are ready, we feed the documents retrieved from our opinion retrieval system, which by default are considered as subjective, to them respectively. If a document is labeled as objective by both of the positive and the negative systems, we set it to positive as a default value.

5.2 Mixed Opinion Identification

We do not simply take for granted that, if a document is labeled as both positive and negative, it should be a mixed opinionative document. This definition does not take the strength of the opinions into consideration. For example, given two documents of d_1 and d_2, both contain the opinions about the query “NASA”. 40% of the opinions in d_1 are positive and the other 60% are negative, while d_2 has 97% positive opinions and 3% negative opinions. A reasonable decision should be that d_1 is mixed and d_2 is positive, because the two kinds of opinions are about even in d_1, but the positive contents are overwhelming in d_2. Based on this understanding, we design a set of rules to test the documents that are labeled as both positive and negative, and to give them the final polarity labels. Firstly, if both positive and negative opinions are strong in the document, the document should be mixed. Otherwise, if one type of the opinions is strong, the document is labeled to that type. Then the rest of the documents in the intersection are remained as mixed.

Given a query Q, let avgNs be the average negative score defined in formula 4.2 of the negative documents. Let avgNc be the average negative score defined in formula 4.3 of the negative documents. Similarly we define the avgPs and avgPc for the positive documents. Let d be a retrieved opinionative document for Q. let NSimstcs(d) and NSimstcc(d) be d’s two negative scores according to formula 4.2 and 4.3 respectively. The two positive scores of d are defined respectively as NSimstcs(d) and NSimstcc(d). The first rule is

$$\begin{aligned} \mathit{Polarity\_label}(d) = \mathit{mixed}, \text{ if } & (\mathit{PSim}_{stcs}(d) > \mathit{avgPs} \times k) \text{ and} \\ & (\mathit{PSim}_{stcc}(d) > \mathit{avgPc} \times k) \text{ and} \\ & (\mathit{NSim}_{stcs}(d) > \mathit{avgNs} \times k) \text{ and} \\ & (\mathit{NSim}_{stcc}(d) > \mathit{avgNc} \times k) \end{aligned} \quad (5.1)$$

Rule 5.1 has the highest priority among all the rules. Rule 5.1 says that if in both the positive and the negative set, both d’s two kinds of opinion scores are well above the average, then d should be a document with mixed opinions. The coefficient *k* is set to 1.7 empirically.

$$\mathit{Polarity\_label}(d) = \begin{cases} \mathit{positive}, \text{ if } (\mathit{PSim}_{stcs}(d) / \mathit{NSim}_{stcs}(d) > \mathit{avgPs} / \mathit{avgNs}) \text{ and} \\ \qquad (\mathit{PSim}_{stcc}(d) / \mathit{NSim}_{stcc}(d) > \mathit{avgPc} / \mathit{avgNc}) \text{ and} \\ \qquad (\mathit{PSim}_{stcs}(d) > \mathit{avgPs} \text{ or } \mathit{PSim}_{stcc}(d) > \mathit{avgPc}) \\ \\ \mathit{negative}, \text{ if } (\mathit{NSim}_{stcs}(d) / \mathit{PSim}_{stcs}(d) > \mathit{avgNs} / \mathit{avgPs}) \text{ and} \\ \qquad (\mathit{NSim}_{stcc}(d) / \mathit{PSim}_{stcc}(d) > \mathit{avgNc} / \mathit{avgPc}) \text{ and} \\ \qquad (\mathit{NSim}_{stcs}(d) > \mathit{avgNs} \text{ or } \mathit{NSim}_{stcc}(d) > \mathit{avgNc}) \end{cases} \quad (5.2)$$

Rule 5.2 has a priority lower than that of the rule 5.1 but higher priority than others. Rule 5.2 says that in order to be a positive document, the ratio of the document’s positive score to its negative score should be above the ratio average, and the document’s positive scores should be no less than the average positive scores. In order to be a negative document, the ratio of the document’s negative score to its positive score should be above the ratio average, and the document’s negative scores should be no less than the average negative scores.

$$
Polarity\_label(d) = \begin{cases} positive, \text{ if } (PSim_{stcs}(d) > NSim_{stcs}(d)) \text{ and } & (5.3) \\ \qquad (PSim_{stcc}(d) > NSim_{stcc}(d)) & \\ negative, \text{ if } (NSim_{stcs}(d) > PSim_{stcs}(d)) \text{ and } & \\ \qquad (NSim_{stcc}(d) > PSim_{stcc}(d)) & \\ mixed, \text{otherwise} & \end{cases}
$$

Rule 5.3 has a priority lower than that of the rule 5.2. Rule 5.3 says that in order to be a positive document, its positive scores should be greater than its negative scores, while a negative document should have greater negative scores. Otherwise the opinions are mixed.

6. RESULTS OF THE SUBMITTED RUNS

In this year's main task, we want to test two parameters in our opinion retrieval system. This first one is the weights of the expanded query terms in the topic retrieval module. The term similarity score defined in Section 2.3 consists the Okapi scores of the original query terms and the weighted Okapi scores of the expanded terms. We set the default value range of the weight of the expanded terms to $(0,\ 0.7]$. For this weight range parameter, we test various values for the right boundary, e.g. $(0,\ k\times 0.7]$ where k=1, 0.75, 0.5 or 0.25. Finally we decide to apply k=1 and k=0.75 respectively. A smaller k value means that the expanded query terms are less important. The second parameter to be tested is the opinion similarity function. We have defined five such functions in formula 4.1 to 4.5. The combined functions of 4.4 and 4.5 always help the system get higher scores than the three simple functions. But we are not sure which one of 4.4 and 4.5 is better. So for this position, we use both of them.

Two parameters, each of which has two possible values, produce four different system configurations. These four systems are built by using the same training data, so that the results of the opinion retrieval main task would only be affected by these two parameters. Table 1 shows the scores of our four submitted runs to the main task. The uic75c configuration has the highest overall scores. But the score difference between any two runs is small.

Run	Weights of the expanded terms	Opinion similarity function	Overall MAP	Overall R-precision
uic1c	$(0,\ 0.7]$	$\text{OSim}_{\text{ir\_stcc}}$	0.4341	0.4529
uic1s	$(0,\ 0.7]$	$\text{OSim}_{\text{ir\_stcs}}$	0.4255	0.4522
uic75c	$(0,\ 0.75\times 0.7]$	$\text{OSim}_{\text{ir\_stcc}}$	0.4341	0.4538
uic75s	$(0,\ 0.75\times 0.7]$	$\text{OSim}_{\text{ir\_stcs}}$	0.4241	0.4521

Table 1. The opinion retrieval scores of the runs from UIC

We only had one configuration for our polarity classification system. So the documents in each of the four opinion retrieval runs were processed by the same system. Table 2 shows the overall scores of the four polarity classification runs. Again the run associated with the uic75c run has the best score among the four runs.

Run	Corresponding opinion retrieval run	Overall R-precision	Correct@10	Correct@20
uic1cpnm	uic1c	0.2284	0.3720	0.3530
uic1spnm	uic1s	0.2266	0.3580	0.3350
uic75cpnm	uic75c	0.2295	0.3700	0.3530
uic75spnm	uic75s	0.2281	0.3660	0.3350

Table 2. The overall polarity classification scores

7. CONCLUSIONS

In the opinion retrieval task of the TREC 2007 Blog Track, we develop a three-step algorithm to retrieve documents that have opinioned content about a query topic. The system has the new features such as the new way of using the training data, the single opinion classifier and the combined opinion similarity functions. For the polarity classification task, we adopted a “split-and-merge” strategy to distinguish the three kinds of opinions. Both the two opinion classifiers and the judging function in our polarity system need further study to make the polarity system perform at the same level as the opinion retrieval system.

REFERENCES

[1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto: *Modern Information Retrieval*, Addison-Wesley, 1999.

[2] Chernoff H, Lehmann E.L. The use of maximum likelihood estimates in $\chi 2$ tests for goodness-of-fit. The Annals of Mathematical Statistics 1954; 25:579-586.

[3] http://en.wikipedia.org

[4] T. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of the European Conference on Machine Learning (ECML), Springer, 1998.

[5] Shuang Liu, Fang Liu, Clement Yu, and Weiyi Meng. An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. In Proceedings of the 27th Annual International ACM SIGIR Conference. 2004.

[6] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, Ian Soboroff. Overview of the TREC-2006 Blog Track. In proceedings of the 15th TREC. 2006.

[7] S. Robertson, S. Walker Okapi/Keenbow at TREC-8, 1999.

[8] http://wordnet.princeton.edu/

[9] http://www.cs.ualberta.ca/~lindek/minipar.htm

[10] Wei Zhang and Clement Yu. Opinion Retrieval from Blogs. In proceedings of the 16th CIKM. 2007.

[11] Wei Zhang and Clement Yu. UIC at TREC 2006 Blog Track. In proceedings of the 15th TREC. 2006.

[12] Wei Zhang, Shuang Liu, Clement Yu, Chaojing Sun, Fang Liu and Weiyi Meng. Recognition and Classification of Noun Phrases in Queries for Effective Retrieval. In proceedings of the 16th CIKM. 2007.