ISCAS at Multilingual Opinion Analysis Task

Yunping Huang, Yulin Wang, Le Sun
Institute of Software, Chinese Academy of Sciences
P.O.Box 8718, Beijing, 100190, P.R.China
yunping07@iscas.ac.cn

Abstract

The paper presents our work in the multilingual opinion analysis task in NTCIR7 in Simplified Chinese. In detecting opinionated sentences, an EM algorithm was proposed to extract the sentiment words based on the sentimental dictionary, and then an iterative algorithm was used to estimate the score of the sentiment words and the sentences. In detecting relevant sentences, we solve this problem by analogizing the task to the traditional information retrieval task. The difficulty lies in that some sentence is relevant to the topic even if there are no key words hit in it. In this situation, we use a pseudo feedback and query extension method to refine the result. The evaluation results and the result analysis will also be presented.

Keywords: *NTCIR, Opinion Analysis.*

1. Introduction

The processing of opinion information has been widely discussed these days. There are a large amount of written subjective texts referring to the quality of commercial products or the political elections, and the desire to extract and analyze the opinions expressed by the consumers and the voters is becoming high. People are concerned about opinion existed in these data, which makes opinion analysis a quite attracting and active research domain recently.

There are five subtasks defined at sentence level in MOAT task: opinionated sentences, relevant sentences, opinion polarities, opinion holders, opinion targets. We participated in two subtasks: detecting the opinion sentences, and detecting the relevant sentences.

In the subtask of detecting opinion sentences, we use a lexicon based algorithm, which considers the sentiment words in the sentiment dictionary as the seed sentiment words, and uses the statistical method to expand the sentiment words. At last, we compute the opinion score of a sentence through the sentiment words. The proposed method can automatically expand the sentiment words, estimate sentiment score of the word and detect the opinionated sentence iteratively. The experimental results are satisfactory.

In the subtask of detecting relevant sentences, certain topics and relevant documents are given. Each document contains tens to hundreds of sentences. What we need to do is to judge if one sentence in a document is relevant to the given topics. We find that this task is very similar to the traditional information retrieval task. In this situation, the topics are peer to the search queries, and sentences are peer to the whole corpus retrievable. If we can convert the topics to queries, and sentences to corpus, we can convert this relevant task to at retrieval task. But there is another problem. Some sentences that don't contain any key words from the topics, but they indeed are relevant to the topics. This is because the topics are short and can't cover all the aspects that the sentences are talking about. We borrowed the implicit feedback and query extension method to refine the topics to improve the recall rate.

The remainder of the paper is organized as follows: firstly, we will present the related work in section 2. Then our methods to detect opinionate sentences and detect relevant sentences will be described separately in section 3 and 4. Finally, conclusions and future work will be presented in section 5.

2. Related Work

Opinion analysis can be also called subjectivity analysis. It is the task of identifying subjective words, expressions ([7]), and sentences ([2], [10]), or documents ([6], [8]). Early work focused mainly on determining whether sentences or documents contain opinion information. Some work used the machine learning methods to classify the sentences as subjective or objective, Wiebe and her colleagues [9] used a corpus tagged at the sentence level for subjectivity to train a Naive Bayes classifier using syntactic classes, punctuation, and sentence position as features. More recently, Pang and Lee [6], Yu and Hatzivassiloglou[12] used the training corpus to develop a classifier. However, machine learning method need training corpus, and training corpus was lacked because of the diversity of topical domain. Rule based method can be used to detect opinionated sentences. [3] used the lexical clues to extract initial candidate opinionated sentences.

There are also many simple empirical algorithms which detect opinionated sentences based on the sentiment lexicon. [11] created a very lenient classifier for detecting opinionated sentences using the sentiment lexicons, a sentence was considered as opinionated if it contain at least one sentiment word. [4] used the sentiment score to compute the sentences' sentiment score.

3. Detecting Opinionated Sentences

Discriminating subjective sentences from objective ones can be considered as a binary classification problem. Many researchers had adopted supervised machine-learning method for it. However, it needs a variety of lexical and contextual features. These machine-learning methods seem too complicated for our Chinese task, so we used a simple empirical algorithms based on the sentiment lexicons to detect opinionated sentences. For each sentence, we used the following formula to obtain opinion score of a sentence.

$$S_P = \sum_j S_{W_j} \qquad (1)$$

Where S_P and S_{W_j} are sentiment score of sentence P and sentiment score of word Wj. If the opinion score is greater than the predefined threshold value, we consider it opinionated sentence.

The key problem is how to extract the sentiment words, and how to estimate the sentiment score. The following will describe our method in detail.

3.1. Extracting Sentiment Words

Since our method was based on the sentimental lexicons, the first problem is how to collect as many sentiment words as possible, for this, we use several existed emotion dictionaries to extract seed sentiment words. The first emotion dictionary is provided by Hownet[1], and then we enlarge the vocabulary by consulting tong2yi2ci2lin of HIT. The other emotion dictionary we used is the emotion dictionary[1] offered by NTCIR downloaded from its website.

After these operations, we can get a relatively big emotion dictionary. However, this can not cover all sentiment words. Many sentiment words will not be included by these dictionaries, and there are also many topic-related sentiment words, thus, we need to use statistical method to extract sentiment word. Turney[8] used the statistical method to extract the sentiment words.

We assume that the word usually occurred around sentiment word will have a big possibility to be sentiment word. Based on this assumption, we proposed an EM algorithm to extract the sentiment words.

E-step:

$$Z_{wj} = \frac{p^{(n)}(w \mid \theta_j)}{\sum_{j'=1}^{k} p^{(n)}(w \mid \theta_{j'})} \qquad (2)$$

$$Z_{wB} = \frac{\lambda_B p(w \mid \theta_B)}{\lambda_B p(w \mid \theta_B) + (1-\lambda_B)\sum_{j=1}^{k} \pi_{d,j}^{(n)} p^{(n)}(w \mid \theta_j)}$$

M-Step:

$$p^{(n+1)}(w \mid \theta_j) = \frac{c(w,\theta_j)(1-Z_{wB})Z_{wj}}{\sum_{w' \in V} c(w',\theta_j)(1-Z_{w'B})Z_{w'j}} \qquad (3)$$

Where Z_{wj}, Z_{wB} are the hidden variables, indicating the word w being generated by θ_j, θ_B respectively. θ_j is the j-th seed sentiment word in sentiment dictionary; θ_B is the background model, and $p(w \mid \theta_j)$ indicates the possibility that θ_j trigger word w. $p(w \mid \theta_B)$ is fixed, and its value can be computed as following.

$$p(w \mid \theta_B) = \frac{c(w)}{\sum_{w' \in V} c(w')} \qquad (4)$$

Where c(w) is the frequency of word w.

After the EM algorithm converges, the following formula is used to compute the score of word w, and we select the biggest M word as sentiment words. The most frequent words and the word which just contains one character are removed.

$$Opin(w) = \sum_j p(w \mid \theta_j) \qquad (5)$$

3.2. An Iterative Algorithm

After extracting the sentiment words, the other key problem is how to estimate the score of the sentiment word. Since the dictionary does not define the strength of the words, we need to compute the sentiment score of the words. Ku et, al. [5] use the sentiment scores of the composing characters to compute the score of a sentiment word, and they use the dictionary to estimate the score of the sentiment score of the character. In our system, the following formula is used to estimate the strength of the sentiment word.

$$S_{W_j} = \frac{OpinSF(W_j)}{SF(W_j)} \qquad (6)$$

Where $OpinSF(W_j)$ is number of opinion sentences which contain word Wj; $SF(W_j)$ is the sentence frequency of word Wj.

However, $OpinSF(W_j)$ is unknown, so we propose an iterative algorithm to estimate it. First, Rule 1 is used to select the initial opinionated sentences.

Rule 1: if one sentence has "claim verb" and at least one sentiment word, this sentence is selected as the seed opinionated sentence.

The "claim verbs" we used is from HowNet[1]. There are 38 claim verbs shown in Table 1.

After that, an initial opinionated sentence set is obtained, the score of the sentiment word can be computed, and it can be used to calculate the score of the sentence using formula 1. After that, the new score can

[1] http://nlg18.csie.ntu.edu.tw:8080/opinion/pub1.html

be used to detect the opinion sentences, which can be used to estimate the sentiment score of the word more accurately. This process can be continued until convergence (the number of new selected opinioned sentences is less than the threshold). The iterative algorithm is shown in Figure 1.

Table 1. Claim Words of Simplified Chinese

察觉	感受到	预感	论
触目	见到	自觉	认定
耳闻	见得	抱定	认为
发	觉	当	认准
发觉	觉得	道	想
发现	看得出来	感到	相信
风闻	窥见	感觉	以为
感	领教	觉得	主张
感觉	听说	看	
感觉到	痛感	看待	

Input: sentence set of one topic
Output: opinionated sentence set O.
1. Insert all sentences into unlabelled set U, use the rule to annotate the opinionated sentences, insert these sentences into O, and remove these sentences from U.
2. Estimate the sentiment score of the sentiment word using formula 6.
3. For each sentence p in U
Re-estimate the sentiment score using formula 1 through the updated opinion score of sentiment word
If the sentence is opinionated, insert it into O, and remove it from U.
4. Go to step 2 and iterate until convergence.

Figure 1. Iterative Algorithm to Detect Opinionated Sentences

3.3. Evaluation results and result analysis

The evaluation results are presented in Table 2 and 3. Seen from the table, our system received average scores, and the precision and recall value are not quite high. The possible reason is our method may bring noise, for example, when using the rule 1, the "claim verb" may be ambiguous. The other possible reason is that there are many parameters to be tuned, and we failed to tune it to be optimal.

Table 2: Opinion Analysis Results of Simplified Chinese(Lenient)

Group	RunID	Opinionated (Lenient)		
		P	R	F
BUPT	1	0.604	0.3991	0.4807
ICLPKU	1	0.4803	0.8004	0.6003
ICLPKU	2	0.4487	0.7983	0.5745
NEUNLP	1	0.4721	0.7116	0.5676
NLCL	1	0.4425	0.3991	0.4197
NLCL	2	0.4822	0.3686	0.4178
NLCL	3	0.4316	0.6988	0.5336
NLPR	1	0.5822	0.7753	0.665
NLPR	2	0.588	0.4842	0.5311
NLPR	3	0.4551	0.5725	0.5071
NLPR	4	0.5769	0.5639	0.5703
NTU	1	0.5939	0.6089	0.6013
NTU	2	0.5956	0.6067	0.6011
NTU	3	0.5956	0.6067	0.6011
TTRD	1	0.412	0.9636	0.5772
TTRD	2	0.4456	0.756	0.5607
WIA	1	0.5862	0.8208	0.6839
ISCAS	1	0.4649	0.7442	0.5723

Table 3: Opinion Analysis Results of Simplified Chinese(Strict)

Group	RunID	Opinionated (Strict)		
		P	R	F
BUPT	1	0.6312	0.4421	0.52
ICLPKU	1	0.4486	0.8207	0.5801
ICLPKU	2	0.3984	0.8252	0.5373
NEUNLP	1	0.4358	0.7339	0.5469
NLCL	1	0.3857	0.402	0.3937
NLCL	2	0.4425	0.3898	0.4144
NLCL	3	0.3667	0.706	0.4827
NLPR	1	0.6096	0.892	0.724
NLPR	2	0.6129	0.5501	0.5798
NLPR	3	0.4197	0.637	0.506
NLPR	4	0.5973	0.6459	0.6207
NTU	1	0.6314	0.7517	0.6863
NTU	2	0.6343	0.7494	0.6871
NTU	3	0.6343	0.7494	0.6871
TTRD	1	0.3481	0.9699	0.5124
TTRD	2	0.3958	0.755	0.5193
WIA	1	0.6098	0.8964	0.7259
ISCAS	1	0.4271	0.8118	0.5597

4. Detecting Opinionated Sentences

In Relevance subtask of MOAT, we judge if one sentence in a document is relevant to a given topic. We find that this task is very similar to the traditional information retrieval task. But there is another problem. Some sentences that don't contain any key words from the topics, but they indeed are relevant to the topics. We use the pseudo feedback and query expansion method to refine the topics to improve the recall rate.

4.1 Relevance Judgment Method

First, we segment the topics and filter stop words out of topics. One is dealing with long Chinese phrases. While retaining the long Chinese phrases, we segment these phrases into smaller words, such as "美国经济" into "美国" ,"经济" , and "暴力事件" into "暴力" , "事件" . The last precession is to restore the original abbreviation expressions, "中俄" to "中国 俄国" for example.

After first stage retrieval, we get a relatively small relevant result set. Although the result is of high precision, the recall rate is low. We need to extend the content that topics cover. Since we consider the topic as a query, we use pseudo feedback for query expansion. So we select words from the result from the first retrieval. After this, the topics contain more content than before.

4.2 Evaluation results and result analysis

The evaluation results are presented in Table 4 and 5. Seen from the table, we find that for each participant the recall rate is relatively lower than precision. By using pseudo feedback and query expansion, we got a relatively higher recall rate. This demonstrates that the traditional information retrieval method is appropriate for the relevant problem, and pseudo feedback and query expansion are very useful in improving the recall rate.

Table 4: The evaluation results for relevance subtask(Lenient)

Group	RunID	Relevance (Lenient)		
		P	R	F
ICLPKU	1	0.9775	0.6559	0.785
ICLPKU	2	0.9775	0.6559	0.785
NLCL	1	0.963	0.3258	0.4869
NLCL	2	0.9752	0.2799	0.4349
NLCL	3	0.9714	0.585	0.7302
NTU	1	0.9656	0.7693	0.8564
NTU	2	0.9796	0.5798	0.7284
NTU	3	0.9767	0.5796	0.7275
TTRD	1	0.9507	0.6981	0.8051
TTRD	2	0.968	0.7363	0.8364
WIA	1	0.994	0.5032	0.6682
ISCAS	1	0.9703	0.9288	0.9491

Table 5: The evaluation results for relevance subtask (Strict)

Group	RunID	Relevance (Strict)		
		P	R	F
ICLPKU	1	0.9845	0.6743	0.8004
ICLPKU	2	0.9845	0.6743	0.8004
NLCL	1	0.9736	0.3326	0.4959
NLCL	2	0.9848	0.2846	0.4415
NLCL	3	0.9827	0.5897	0.7371
NTU	1	0.9748	0.7859	0.8702
NTU	2	0.9878	0.5969	0.7441
NTU	3	0.9866	0.5943	0.7418
TTRD	1	0.9631	0.7006	0.8112
TTRD	2	0.9759	0.7487	0.8474
WIA	1	0.9969	0.524	0.687
ISCAS	1	0.9828	0.9369	0.9593

5. Conclusions and Future Work

This paper describes our system in detail for NTCIR 7 MOAT track. Our System has two main components. The first is opinionated sentence detecting component, and the second is relevant sentence detecting component.

In the subtask of detecting opinionated sentences, our system just receive average scores. The precision and recall scores are not quite high. After speculation, we find that our method has many limitations: 1). When using the EM algorithm to extract sentiment words, the effect is influenced by the initial seed sentiment words, so maybe some preprocesses are needed. 2). In the iterative algorithm, there are also many problems. The initial rule may bring noise. The “claim verb” may be ambiguous, especially for the word that just contain a character, so the rule may be changed as “if one sentence has “claim verb” as the main verb and other sentiment word, this sentence is selected as the seed opinionated sentence”. We realize that if the WSD (word sense disambiguation) can be incorporated, the system can be improved. 3). When estimating the score of sentiment word, it is sensitive to the scale of the collection, and just using the collection to estimate the score may be not very suitable, so it is necessary to combined it with other method. For example, if we have a dictionary with strength score, we can incorporate it as a prior value.

In future work, we will reinforce our method through overcoming the above limitation, and we will test sensitivity of the performance to the introduced parameters, and then develop some methods to automatically tune the parameters.

6. Acknowledge

This research is supported by the National Natural Science Foundation Program of China under grants 60773027 and the National High Technology Research and Development Program of China (863 Program) 2006AA010108.

References

[1]. Z. Dong, Q. Dong. HowNet, http://www.keenage.com/zhiwang/e_zhiwang.html.
[2]. V. Hatzivassiloglou, K.R. McKeown, Predicting the semantic orientation of adjectives. Proceedings of the 35th Annual Meeting of the ACL andthe 8th Conference of the European Chapter of the ACL (pp. 174-181). New Brunswick, NJ: ACL, 1997.
[3]. Y. Kim, S.H. Myaeng, Opinion Analysis based on Lexical Clues and their Expansion. In Proceedings of NTCIR-6, 2007.
[4]. L.W. Ku, Y.S. Lo and H.H Chen, Using Polarity Scores of Words for Sentence-level Opinion Extraction. In Proceedings of NTCIR-6, 2007.
[5]. L.W. Ku, Y.T. Liang, and H.H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. AAAI-CAAW'06, 2006.
[6]. B. Pang, L. Lee and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the 2002 Conference on EMNLP, pages79-86. 2002.
[7]. E. Riloff, J. Wiebe. Learning extraction patterns for subjective expressions. Proceedings of the 2003 Conference on EMNLP, pages 105-112, 2003.
[8]. Peter Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proc. of the ACL, 2002.
[9]. J.Wiebe, R. Bruce, and T. O’Hara. Development and use of a gold-standard data set for subjectivity classifications.

In Proceedings of the 37th Association of Computational Linguistics, pages 246–253, 1999.

[10]. J. Wiebe and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In CICLing-2005.

[11]. Y. Wu, D.W Oard. NTCIR-6 at Maryland: Chinese Opinion Analysis Pilot Task. In Proceedings of NTCIR-6, 2007.

[12]. H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In Proceedings of the 2003 conference on Empirical methods in natural language processing, volume 10, pages 129–136, Morristown, NJ, USA, 2003. Association for Computational Linguistics.