Opinion Analysis for NTCIR8 at POSTECH

Hun-Young Jung Jungi Kim Jong-Hyeok Lee

Knowledge and Langauge Engineering Laboratory
Department of Computer Science and Engineering
Pohang University of Science and Technology
San 31, Hyoja-Dong, Nam-Gu, Pohang, Republic of Korea
{blesshy, yangpa, jhlee}@postech.ac.kr

ABSTRACT
We describe an opinion analysis system developed for a Multilingual Opinion Analysis Task at NTCIR8. Given a topic and relevant newspaper articles, our system determines whether a sentence in the articles has an opinion. If so, we then extract the holder of the opinion. In the opinion judgment task, we constructed a phrase-level opinion expression extractor from sentence-level annotated corpus. In opinion holder extraction task, we used the probability that the word is appeared in the opinion holder and a dependency relationship between the word and the verb of the sentence.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: linguistic processing

General Terms
Experimentation

Keywords
Opinion Analysis, Multilingual Opinion Analysis Task, MOAT, NTCIR

1. INTRODUCTION
The Multilingual Opinion Analysis Task (MOAT) at NTCIR is a task to judge the opinion of a sentence and extracting its properties, which relate to the opinion of the sentence, such as polarity, relevance to a topic, holder and target from a set of newspaper articles in English, Chinese and Japanese. The MOAT at NTCIR8 adds a task called answerness. The Answerness judgment task decides whether a sentence has information which can be an answer to a given question. The opinion judgment task, relevance judgment task and answerness judgment task are sentence level tasks and opinion holder extraction, target extraction and polarity judgment are clause level task [6].

Among the tasks defined for NTCIR8, we focused on the opinion judgment task and the opinion holder extraction task. Whereas our previous work at NTCIR7 focused on the judgment of opinion sentences using term weighting, our work at NTCIR8 focused on extracting the opinion expression and the opinion holder.

In the opinion judgment task, we converted this task into a task to extract the opinion expression task. If a sentence has an opinion expression, the sentence is opinionated. Because our training data is annotated at the sentence-level, we constructed the phrase-level data automatically using Nam's approach [5]. Furthermore we made an opinion expression extractor using a Conditional Random Field (CRF) model.

In the opinion holder extraction task, we calculated a score for a given word to estimate how often each word appears in the phrase that represents the opinion holder. Also, we extracted information about the relationship between each word and verb of the sentence by the dependency parse tree. With those two segments of information, we made the opinion holder extractor using CRF model.

The remainder of the paper is organized as follows; firstly, we will describe our motivation in section 2. Our system for the extraction of the opinion expression and holder will be described in section 3. Finally, the results will be presented in section 4 and the conclusion is in section 5.

2. Related Work and Motivations
In this section, we describe our motivation based on a number of related works that made an opinion expression extractor and opinion holder extractor.

First, information on the opinion expression is useful to extract the opinion holder. Choi et. al [1] made an opinion holder extraction system using a CRF model. They used information on lexicons, grammatical roles and sentiment lexicons. Even before applying a Integer Linear Programming Approach, the performance was of a high level.

Second, a word-level context independent score does not provide enough information to find the opinion expression. So, a phrase-level opinion judgment is needed. In Wilson's paper, they showed an example to explain why a phrase-level opinion analysis is needed [9].

- The polluters are suddenly going to become reasonable.
- They are polluters.

In the first sentence, word 'polluters' is not a subjective word but an objective word, while the same word in the second sentence is a subjective word.

Third, the phrase that represents the opinion holder is closely related to the verb. Our approach in NTCIR7 for opinion holder extraction had one simple rule: find the subject of the main verb and check if the subject has a named entity or a word in a manually constructed lexicon. Kim et. al [4] made 6 heuristic rules to extract an opinion holder; four of the 6 rules are related to a certain special verb. These rules provide the best performance in NTCIR6 and were also used in Seki's paper [7].

Finally, some nouns cannot exist in the opinion holder while some nouns frequently appear. In the NTCIR7 training corpus, there are some words appear in the holder phrase very frequently, such as spokesman, sociologist, psychologist and dealer. Words such as su-30, soil and sever do not appear in the holder phrases.

3. The Proposed System

We consider the extraction problem as a sequential tagging problem. To extract the opinion expression and opinion holder, we tagged each word using the CRF model. We made two sequential tagging models: one for opinion expression, the other for the opinion holder.

In the opinion expression extraction, we made a phrase-level corpus of opinion expressions to establish the CRF model via training. To make the phrase-level corpus, we made a sentiment lexicon automatically from our sentence-level corpus. Also, we annotated words in subjective sentences of sentence-level corpus as opinion expressions if those words appeared in the sentiment lexicon. This annotation constructs the phrase-level opinion expression corpus.

In opinion holder extraction, we estimated the probability that a word appeared in a phrase of the opinion holder. Also we extracted a path from each word to the verb of a sentence from a dependency tree of the sentence. We used the probability of the word and the path in the dependency tree as a feature when training the CRF model.

We used the Stanford parser [1] to extract the dependency information and the CRF++ [2] to construct the CRF model. In the parameter setting process, we used NTCIR7 training data as the training corpus and NTCIR7 formal data as the development corpus. However when we made a formal run at NTCIR8, we merged the NTCIR7 training and formal data and used it as a training corpus.

3.1 Opinion Expression Extraction

Nam et. al [5] suggested a learning method to construct a phrase-level opinion extraction system from a sentence-level annotated corpus. They constructed a sentiment lexicon from movie reviews which are a sentence-level annotated corpus, and tagged some words in a positive or negative sentence as a phrase of opinion, if the words is positive or negative word in the sentiment lexicon. In that paper, their movie reviews is the corpus for polarity and the review scores were from 1 to 10. So, we cannot directly apply their approach to construct the lexicon because our corpus has just two labels for each sentence: subjective or objective. Therefore we used Turney's method [8] to construct the sentiment lexicon.

3.1.1 Construct Subjectivity Lexicon

Turney's method is based on Pointwise Mutual Information (PMI). Using PMI, the polarity score *PS(w)* is defined as

$$PS(w) = PMI(w,pos) - PMI(w,neg) = \log_2 \frac{P(w,pos)/P(pos)}{P(w,neg)/P(neg)} = \log_2 \frac{P(w,pos)}{P(w,neg)}$$

where *w* is a given word, *pos* is set of positive sentences and *neg* is set of negative sentences.

According to this method, we defined the subjectivity score *S(w)* as

$$S(w) = \log_2 \frac{P(w,subjective)}{P(w,objective)}$$

If *S(w)* is larger than a given threshold, the word *w* is judged as a subjective word.

3.1.2 Features

In the training CRF model, we used features from Wilson's paper [9]. Table 1 is a list of the feature for opinion expression extractor.

For a word feature, we used not only the word token but also the Parts of Speech (POS) of the previous word, the word itself and the next word. 'Prior subjectivity' is a subjectivity score in Senti-WordNet.

For the general modification features, we used two binary features and two continuous features. 'Preceded by an adjective' and 'preceded by an adverb' is a binary feature. 'Preceded by an adjective' is true if a word is a noun preceded by an adjective. 'Preceded by an adverb' is true if a word is preceded by an adverb. 'Modifies' is a subjective score of the parent of a word. 'Modified by' is a the maximum subjective score of the children of a word.

To get a subjective score, we used the Senti-WordNet [3] score [2]. Senti-WordNet gives a polarity score for a given word. We considered a higher score between a positive and a negative as the subjectivity score of a word.

Table 1. Features for opinion expression extraction

Feature Groups	Features
Word Feature	word token
	word part of speech(POS) tag [-2,2]
	prior subjectivity
General Modification Features	preceded by an adjective
	preceded by an adverb
	modifies
	modified by

3.2 Opinion Holder Extraction

In NTCIR7, our system for opinion holder extraction is a heuristic rule based approach. The system finds the subject phrase of the main verb and selects the phrase if the phrase is a named entity or has some special noun [3]. To identify a special noun, we made a

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

[2] http://crfpp.sourceforge.net/

[3] http://sentiwordnet.isti.cnr.it/

lexicon manually. To overcome the limitation in the coverage of the heuristic rule and the manually constructed lexicon, we put the heuristic rule and the lexicon into the CRF model.

Instead of a manually constructed lexicon, we calculated the probability that a word appeared in the phrase of the opinion holder. To replace the heuristic rule to find the subject of a sentence, we extracted the path from the word to the verb in the dependency tree.

In this paper, we did not consider anaphoric resolution or quoted sentences, though these are factors that could effect the opinion holder extraction.

3.2.1 Construct Lexicon for Opinion Holder

To estimate the probability that a word appears in a holder phrase, we defined the score *H(w)* for a given corpus as:

$$H(w) = \max_{doc \in corpus}\{H(w,doc)\}.$$

Also we defined the opinion holder score for a given document *H(w,doc)* as:

$$\begin{aligned} H(w,doc) &= P(holder \mid w,doc) \\ &= \frac{P(w \mid holder,doc)P(holder \mid doc)}{P(w \mid doc)} \\ &= \frac{\dfrac{count(w \mid holder,doc)}{size(holder \mid doc)} \times \dfrac{size(holder \mid doc)}{size(doc)}}{count(w \mid doc)/size(doc)} \\ &= \frac{count(w \mid holder,doc)}{count(w \mid doc)} \end{aligned}$$

where the *holder* means a bag of words from phrases which represent the opinion holder in a given document. Because a word is used as an opinion holder in some document while the word is not used in any other document, we took the maximum score of *H(w,doc)*.

There are many words that appear in the test data but did not appear in the training data. We cannot calculate the *H(w)* of these words. To solve this unseen word problem, we expanded the lexicon using word similarity with WordNet[4]. There are many measures available to calculate word similarity. Among the possible similarity measures, we used the path-similarity method, because this measure is based on the hypernym/hyponym relationship and is in the range from 0 to 1. For the unseen word *'us'*, we calculate the holder score as:

$$w = \operatorname*{argmax}_{w}\{similarity(us,w)\}$$

$$H(us) = similarity(us,w) \times H(w)$$

During implementation, we used the nltk package [5]for python to calculate the word similarity.

3.2.2 Dependency Information

To replace the heuristic rule of our previous approach, we used the path between each word and the verb in the dependency tree. From the dependency tree, we collected verbs from among the ancestors of words, and extracted the path from the word to the verb in the tree. To solve the data sparseness problem, we used information on the main verb and the nearest verb independently. We extracted the first and the last relationship of the dependency path and used those relations as features to establish the CRF model via training.

3.2.3 Features

We used several features. Choi et. al [1] made a successful CRF-based opinion holder extraction system. We added those features. Table 2 is a list of features for opinion holder extraction.

3.2.3.1 Word Feature

A word token is a lemmatized form of a word. We used the POS of the window size [-2, +2]. There are two binary features: starting with the upper case and has an upper case. 'Starting with the upper case' is true if the first capital of the word is upper case. 'Has the upper case' is true if any capital of a word is uppercase. Also, we used *H(w)*.

3.2.3.2 Grammatical Feature

For each word, we took the grammatical role of the word. Also we took the grammatical role of the word's chunk and the previous word's chunk.

3.2.3.3 Opinion Expression Feature

The 'Parent chunk having an opinion expression' is true if the parent chunk of a word has any opinion word. We used information on whether a word and its previous word are in an opinion expression or not.

Choi et. al [1] used a dictionary and manually annotated training data to clarify whether a word is in an opinion expression or not. In this paper, we used the result of our opinion expression extraction.

3.2.3.4 Dependency Feature

We took two verbs among the ancestors of words in the dependency tree; these two verbs can be same. Also we used its word token and POS. We extracted the dependency path to each verb, and used the whole path, the first relationship of the path and the last relationship of the path. For example, if 'nn, pobj, prep, pobj, prep, nsubjpass' is the whole path, the first relationship of the path is 'nn' and the last relationship of the path is 'nsubjpass'.

Table 2. Feature for opinion holder extraction

Feature Groups	Features
Word Feature	word token
	POS tag [-2,+2]
	starting with the upper case
	has the upper case
	H(w)
Grammatical Feture	grammatical role of word
	grammatical role of chunk of word
Opinion Expression Feature	parent chunk having an opinion expression

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

[5] http://www.nltk.org/

		word is included in opinion expression [-1,+1]
Dependency Feature	Nearest Verb	word token
		POS tag
		dependency path
		first relationship of the path
		last relationship of the path
	Main Verb	word token
		POS tag
		dependency path
		first relationship of path
		last relationship of path

4. Results

In this section, we report our result in NTCIR8 MOAT. In the first run, we judged sentences as subjective if the sentence contains any opinion expression. In the second and third runs, we judged sentences as subjective if the sentences contain more than or equal to two or three subjective words.

4.1 Opinion Extraction

Our best performance, based on the F-value, was the third run which judged a sentence if the sentence has more than two opinionated tagged words. When we trained our model in the NTCIR7 training corpus and tested it on the NTCIR7 formal data, we found that a slightly low threshold when constructing sentment lexical has a better performance, if we judged sentences which have any opinion expression. However, with the same threshold and training system with both NTCIR7 training and formal data, this system makes more false-positive errors, when judging opinions in objective sentences, than true-negative errors. This approach resulted in a high recall but low precision. Due to this problem, the third run performed best.

4.2 Opinion Holder Extraction

Because of the low precision and high recall of the opinion extraction, the precision of the opinion holder extraction was very low. Most of the errors ware from annotating the opinion holder for an objective sentence. Also, the software, used in this paper, was based on Simplified Chinese, while our target was Traditional Chinese. We think that converting Traditional Chinese into Simplified Chinese in automatic fashion may cause a problem.

Table 3. Performance of opinion classification in English

Run	Precision	Recall	F-value
KLELAB-1	16.82	95.37	28.6
KLELAB-2	17.9	82	29.39
KLELAB-3	19.68	68	30.53

Table 4. Performance of opinion classification in Traditional Chinese

Run	Precision	Recall	F-value
KLELAB-1	41.98	94.94	58.22
KLELAB-3	44.51	87.92	59.1

Table 5. Performance of opinion holder extraction in English (Only for opinionated sentences)

Run	Precision	Recall	F-value
KLELAB-1	39.7	35.6	37.5
KLELAB-2	41.1	31.7	35.8
KLELAB-3	43.4	27.8	33.9

Table 6. Performance of opinion holder extraction in English (For all sentences)

Run	Precision	Recall	F-value
KLELAB-1	6.8	35.6	11.4
KLELAB-2	7.4	31.7	12.0
KLELAB-3	8.6	27.8	13.2

Table 7. Performance of opinion holder extraction in Traditional Chinese (strict)

Run	Total Evaluated	Correct	Partially Correct	Incorrect	Score
KLELAB -1	3137	923	8	2205	29.6
KLELAB -2	2905	756	10	2139	26.2

5. Conclusion

In comparison with our previous approach at NTCIR7, the opinionated judgment system has a lower precision but a higher recall. During parameter setting stage of the system, a low threshold for the sentiment lexicon performed better, but resulted in a low precision in formal run. We applied the same system for English and Traditional Chinese and we expected a similar performance because of the language independent approach. However, the performance was different; The performance of the opinion holder in Traditional Chinese was especially bad. In further research, we will analysis these unexpected results.

6. ACKNOWLEDGMENTS

This work was supported in part by the BK 21 Project in 2009.

7. REFERENCES

[1] Yejin Choi, Claire Cardie, Ellen Riloff and Siddharth Patwardhan. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In Proceedings of Human Language Technology Conference

and Conference on Empirical Methods in Natural Language Processing (Vancouver, October 2005). 355-362.

[2] A Esuli and F Sebastiani .SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. Proceedings of LREC, 2006. 417-422.

[3] Jungi Kim, Hun-Young Jung, Sang-Hyeob Nam, Yeha Lee, and Jong-Hyeok Lee. 2008. English Opinion Analysis for NTCIR7 at POSTECH. In Proceedings of NTCIR-7 Workshop Meeting (Tokyo, Japan, December 16–19, 2008). 241-246.

[4] Youngho Kim and Sung-Hyon Myaeng. Opinion Analysis based on Lexical Clues and their Expansion. In Proceedings of NTCIR-6 Workshop Meeting (, Tokyo, Japan, May 15-18, 2007). 308-315

[5] Sang-Hyob Nam, Seung-Hoon Na, Jungi Kim, Yeha Lee, and Jong-Hyeok Lee. Partially Supervised Phrase-Level Sentiment Classification. In proceedings of ICCPOL 2009. 225–235.

[6] Yohei Seki, Lun-Wei Ku and Le Sun. Overview of Multilingual Opinion Analysis Task at NTCIR-8. Proceedings of NTCIR-8 Workshop Meeting

[7] Yohei Seki, Noriko Kando, and Masaki Aono. Multilingual opinion holder identification using author and authority viewpoints. Information Processing and Management. Vol 45. (2009). 189–199.

[8] Peter D. Turney and Michael L. Littman. 2002. Unsupervised learning of semantic orientation from a hundredbillion-word corpus. Technical report, National Research Council Canada

[9] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. 2009 Association for Computational Linguistics. Val 35(3). 399-433.