Extracting Topic-related Opinions and their Targets in NTCIR-7

Youngho Kim, Seongchan Kim, Sung-Hyon Myaeng
Information and Communications University
119, Moonji-ro, Yuseong-gu, Daejeon, 305-714, Korea
{yhkim, seongchan, myaeng}@icu.ac.kr

Abstract

In recent years, there have been many interests in opinion resources such as online news, blogs, and forums. With this tendency, many opinion-related applications are proposed to quench an opinion-seeking desire of users. However, selecting publicly interesting opinions is important to improve the effectiveness of such opinion applications, by focusing more on important opinions. To achieve this goal, we propose an opinion mining system which extracts topic-related opinions (at sentence level) and identifies their targets. Our system can be characterized with probabilistic divergence based keyword extraction, language model based topic relevance determination with web-snippets expansion, and heuristic feature based target identification. Experimental results show that our approach is promising.

Keywords: *Opinion Extraction, Opinion Target, Topic Relevance, NTCIR, Opinion Analysis, Opinion Mining.*

1. Introduction

Opinion mining is concerned with discovering opinions and their related features such as sentiment polarity (positive or negative) and opinion holder, by dealing with computational traits of opinionated and subjective text [1]. Nowadays by the growth of popularity in opinion resources such as online news, forums, blogs and reviews, users actively seek opinions for their interest. In order to facilitate such opinion-seeking behavior, numerous methods and related applications have been developed in the past [2,3,4]. However, most existing research has focused on any opinions although some are not interesting to users. As Jindal [5] addressed, there are many spam opinions on the Web, and such wastes can burden both opinion analysis system and the users.

Consequently, we need to be able to measure how interesting and important an opinion might be in order to build a more effective opinion analysis system. While knowing personal interest would be desirable, automatically capturing such information is very difficult and problematic because of privacy issues. However, public interest and importance are less difficult to recognize. That is, identifying opinions that the majority of people would be interested in is feasible. Thus we focus on finding publicly interesting and important opinions in this paper.

Newspapers usually deal with topics and opinions many people are interested in and concerned about. So, an opinion analysis on news articles have been highly desirable [3,6,8]. However, previous systems have predominantly considered opinion extraction, neglecting which opinions are truly relevant to topics people are interested in. Therefore, if we can determine the topic relevance of each opinion, finding interesting opinions would be a relatively easy task. In addition, identifying the target of each opinion would be very helpful. Some particular entities such as products, events, celebrities, and locations, can generally simulate user interests. As an example, in movie reviews, people are likely to read reviews about popular movies such as blockbuster. People tend to buy a popular product and search for reviews about it. In news articles, people are more interested in sensational events such as presidential election and Iraqi war. Such interesting entities are normally well-known to the public, and research for automatic identification of targets has been done in many other application areas (e.g., topic detection, text-mining). However, the task of automatically identifying opinion targets remains as a challenging issue to extract popular opinions. With this missing element in opinion mining, we are motivated to participate in NTCIR-7 MOAT [7] that involves sentence-level opinion extraction, its topic-relevance determination, and its target identification.

In this paper, we focus on extraction of topic-related opinions and their relevance targets. In a given newspaper collection, we first extract sentence level opinions by using an opinion keyword spotting approach [8]. Since statistically most opinions contain opinion keywords such as “say”, “criticize”, and “good” as clue words [6,8], the performance of statistical extraction is sensitive to what clues are used. Previously, most classifiers in their feature spaces have used a clue lexicon. However, we believe that it would be important to carefully select powerful clues instead of trying to add more and more clues to the lexicon. In addition, considering negative keywords (i.e., rarely appearing in opinions, but frequently appearing in non-opinionated sentences) would enhance the performance. Based on this intuition, we generate our feature keywords to include not only important opinion clues but also non-opinionated keywords. We extract such discriminative keywords by measuring statistical distance between two models: opinion model and non-opinion model. Next, our system determines the topic relevance of each opinion. Since we view an opinion as an opinionated sentence, we need to determine the relatedness of such sentence to the document topic to capture related opinions. We adopt a query-likelihood model [11] complemented by query expansion using web-search snippets (some sentences are short and sparseness of topic words is problematic). With the topic-relevance judgment, we developed a heuristic approach for target identification. A statistical learning framework is employed for this task. Finally, we can extract topic-related opinions and their targets, which would play a pivotal role in finding interesting opinions.

For evaluation, our system was run on English news articles from Korea Times, Xinhua News, Hong Kong Standard, and the Straits Times [7]. We obtained an encouraging result for the goals.

The rest of our paper is organized as follows. In Section 2, we investigate related works. We present our system and used methodologies in Section 3 and the evaluation including results and related discussions in Section 4. Finally, we conclude in Section 5.

2. Related Work

In this paper, we focus on the tasks of topic-related opinion extraction and its target identification. So, we investigated previous studies into three categories: opinion extraction, topic relevance judgment, and opinion holder identification. Since we could not find any of published works on target identification, we explored opinion holder identification as a related work for the target identification.

Firstly, to extract opinions, most of previous research has utilized statistical classifiers such as Naïve Bayes, Support Vector Machine and Maximum Entropy models. Wiebe et al. [18] have proposed methods for discriminating between opinionated and factual text at the document, sentence, and phrase levels, as focusing on automated opinion detection. They described a sentence-level Naïve Bayes classifier using as features the presence of particular syntactic classes[1], punctuation, and sentence position. Yu and Hatzivassiloglou [8] have worked the task of separating opinions from facts at both the document and sentence level on Trec 8,9,11 collections by using statistical classifier. Pang [1] employed a machine-learning method, which applies text-categorization techniques to identify the subjective portions of the document. As discussed in [1], in statistical approach to opinion extraction, the role of clue keywords is critical.

Study for topic relevance is finding relevant sentences to a particular topic in a document, and this work newly proposed in NTCIR-6 [19]. Evans [16] improved the effectiveness of the topic relevance task by using a standard vector space model with tf*idf weights and Rocchio blind relevance feedback. Li et al. [17] investigated the effectiveness of the inner product between the topic's feature vector and the sentence's feature vector in the relevance judgment.

In opinion holder identification, Kim [12] aimed to improve the opinion holder identification by identifying opinion sources given an in a sentence. Choi et al. [13] used Conditional Random Fields (CRFs) featured by lexical, syntactic, and semantic words, and extraction patterns are combined into the CRFs.

3. Opinion Mining System

In NTCIR-7, we were given a news collection that includes 17 topics and 167 related documents [7]. Let $C = \{T_1, T_2, \dots , T_{17}\}$ be a set of topics in a collection. Each topic T includes related documents, i.e., $T = \{D_1, D_2, \dots, D_n\}$. We define an opinion as an opinionated sentence, i.e., we classify a sentence into opinionated or factual (i.e., non-opinionated). Next, we determine the relevance of each extracted opinion to the given topic T. With this relevance determination, for each opinion, we find the target *Obj*. As a result, each sentence is associated with a triplet: $S = <opinionated, relevance, Obj>$.

3.1 Opinion Extraction

As addressed in Section 1, previous approaches generally use opinion clue words only. As Yu [8] reported, statistical classifiers (e.g., Maximum Entropy, Support Vector Machine) featured by the presence of opinion clues (i.e., keyword spotting approach) are effective. However, we conjecture that not only using such clue words but also considering their importance would enhance the classification performance. We use a probabilistic

[1] pronouns, adjectives, cardinal number, modal verbs, adverbs

model to measure such importance; we regard the importance as keyword's probability drawn from a model. Also, negative examples (i.e., non-opinionated) can reinforce the classification capability. We model opinions and non-opinions, simultaneously. Since general opinion clues are independent from topicality, we postulate that a non-opinion model would be similar to the topic model as opposed to the opinion model. Thus, we regard keywords that can be drawn from either of opinion and topic models as important keywords useful for the classification as they would serve as highly discriminative factors.

Specifically, we define *n* topic language models as $\theta_{T_1}, \theta_{T_2}, ..., \theta_{T_n}$ ($T_i \in C$) and an opinion language model as θ_{opi}. As an opinion clue generally consists of a unigram word (e.g., "insist", "harmful"), our language model follows unigram word distribution. Each topic language model is estimated using the set of vocabulary (V_T) that comes from given topic as:

$$p(w|\theta_T) = \frac{\sum_{d \in T} freq(w:d)}{\sum_{w' \in V_T} \sum_{d \in T} freq(w':d)}$$

We use the relative frequency of each unigram within the document $d \in T$ as a unigram probability. The opinion language model is defined in the same manner as in the topic model. So, a topic keyword (which frequently occurs in topic documents) would be generally assigned a high probability.

Based on those models, we extract highly discriminative keywords in both opinion and topic models, by measuring a probability gap between two models; less discriminative words are ambiguous words, i.e., a word identically appears in both two models, and such word includes a small probability difference between two models. To extract keywords with a large probability distance, KL-Divergence is utilized:

$$D_{w_i}(\theta_{opi} \| \theta_T) = p(w_i|\theta_{opi}) \log \frac{p(w_i|\theta_{opi})}{p(w_i|\theta_T)}$$

$$\text{where } w_i \in V_T$$

Originally KL-Divergence measures the overall differences of two models, but in our task, we calculate each word's difference between opinion language model and topic language model. Also, the set of topic vocabulary V_T is a superset of vocabulary of opinion model. However, the divergence is asymmetric and an infinity problem occurs when opinion model does not contain a given word (i.e., $\lim_{x \to 0} \log x = -\infty$). Thus, we use total divergence to the average (mean) proposed in [8] as a supplementary measure. An average divergence of each word w_i is given by:

$$A_{w_i}(\theta_{opi} \| \theta_T) = D_{w_i}\left(\theta_{opi} \middle\| \frac{\theta_{opi} + \theta_T}{2}\right) + D_{w_i}\left(\theta_T \middle\| \frac{\theta_{opi} + \theta_T}{2}\right)$$

$$= \left\{ \begin{array}{l} p(w_i|\theta_{opi}) \log \dfrac{2p(w_i|\theta_{opi})}{p(w_i|\theta_T) + p(w_i|\theta_{opi})} \\ + p(w_i|\theta_T) \log \dfrac{2p(w_i|\theta_T)}{p(w_i|\theta_T) + p(w_i|\theta_{opi})} \end{array} \right\}$$

As proved in [9], the average divergence is smoothed from the zero probability and symmetric. Based on this, we develop the keyword extraction algorithm as shown in Figure 1.

```
function extractKeyword
Input:
    $\{\theta_{T_1}, \theta_{T_2}, ..., \theta_{T_n}\}$: the set of topic language model
    $\{V_{T_1}, V_{T_2}, ..., V_{T_n}\}$: the set of vocabulary for each topic model
    $\theta_{opi}$: opinion language model
    $V_{opi}$: opinion vocabulary set

Output:
    $K = \{w_1, w_1, ..., w_m\}$: extracted keyword set

Body:
    1. $K = \varnothing$
    2. for each topic model $\theta_{T_i}$ do
        for each word $w_j \in V_{T_i}$ do
            $S \leftarrow A_{w_j}(\theta_{opi} \| \theta_{T_i})$
            if $S > \delta$ then
                $K \leftarrow w_j$
    3. return $K$
```

Figure 1. Keyword Extraction Function

As shown in Figure 1, for each word we measure the average divergence between opinion model and the topic model (which the word belongs to). If a word is assigned a high probability in opinion model and a low probability in topic model, the long distance (i.e., average divergence) would be measured for the word. We can consider such word as an opinion clue, and it is a highly discriminative keyword. In the reverse case (i.e., high in the topic model but low in the opinion model), there would be a long gap and it would involve a discriminative factor. For stop words (i.e., which would involve high probabilities in both models) and sparse words (i.e., low in both), the average divergence would be small, and it would be unnecessary word in our classification. Thus, by comparing the divergence to the threshold *δ*, we can extract keywords that are expected to contain a discriminative ability. Since the divergence value depends on the corpus nature (e.g., size), the threshold is empirically set. As a result, we can train our statistical learner featured by the keywords that are extracted as above. Also, to assign a weight to each keyword, we use the average divergence value as the feature value for the classifier.

Since NTCIR-7 annotation data were not given yet, to extract the feature words and their weights, we utilized NTCIR-6 opinion analysis data [19] that has similar natures as another news collection.

3.2 Topic Relevance Determination

After opinion extraction, we can obtain the set of opinionated sentences from each topic documents. However, the set contains numerous topic-irrelevant opinions (i.e., which are not related to the overall topic), and those would not be of any interest or concern to users. Thus, in this section, we describe how to select topic relevant opinions among initially extracted opinions in each topic document.

As used in general information retrieval (IR), language modeling is a quite formal approach to find relevant text [11]. In our task, the query would be background topic information (explained in Section 4.1), and the retrieving target would be relevant sentences. However, as noticed in [11], an unsmoothed model works badly because of query term's sparseness in a target text. Thus, to alleviate this problem, we need to smooth the probability in our sentence language model. In a general IR process, smoothing assigns some probabilities for unseen words to avoid zero probabilities. However, in our task, since the length of target text (sentence) would be much shorter than that in general document IR and documents are initially relevant to the given topic, general smoothing method is not applicable. Thus, we need to expand given queries rather than smoothing sentence language models.

In query expansion, relevance feedback approaches are popular in IR, but our collection does not include a sufficiently large document pool to obtain enough relevant documents. One way to handle this situation is to use some external resources, like using a web-search engine to gather relevant query terms. We first collect top 100 snippets from a web-search result for each query and then generate snippet language models to extract highly relevant terms. In this section, we use an n-gram language model because unigrams do not capture meaningful dependency between words. By using an n-gram language model, we extract some high probabilities terms that would be highly relevant to the original query terms. We empirically set the threshold to expand original query. Thus, we obtain new expanded query terms, and our retrieve model is given by:

$$p(Q|S) \approx \sum_{\forall Q'} p(Q|Q',S)\, p(Q'|S)$$

$$\approx \sum_{\forall Q'} p(Q|S)\, p(Q'|S)$$

We briefly estimate the document ranking by the query likelihood $p(Q|S)$. Since the new query Q' is the result snippets of initially retrieved by using the original query Q, we simply assume that the new query is implicitly relevant to the original one; hence, the query likelihood is approximated as follows.

$$\log p(Q|S) \approx \log p(Q|S)\, p(Q'|S)$$

$$\approx \log \prod_{t \in Q} p(t|S) + \log \prod_{t' \in Q'} p(t'|S)$$

$$\approx \lambda \log \prod_{t \in Q} p(t|S) + (1-\lambda) \log \prod_{t' \in Q'} p(t'|S)$$

By using the log likelihood, we can divide the original query likelihood estimation into two parts: likelihood of original query term t and likelihood if a new expanded term t' . The weight bias λ is added to control the effect of newly expanded terms. Moreover, since the retrieval model is implemented based on n-gram language model (n can be 3 in our system), the term t would be unigram, bigram, or tri-gram (Q is divided into tri-, bi-, and uni-grams). The likelihood from each language model can be combined by:

$$p(Q|S) \approx \alpha \cdot p_{uni}(Q|S) + \beta \cdot p_{bi}(Q|S) + \gamma \cdot p_{tri}(Q|S)$$

The weight for each language model is generally set empirically, and in our task, $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.1$ because sparseness is problematic in tri-gram model. As a result, we rank sentences that belong to a given topic by its topic relevance, and the intersection between highly relevant and opinionated becomes topic-related opinions.

3.3 Opinion Target Identification

We now turn to the task of target identification. Identifying an opinion target is a quite a new task in opinion analysis. Also, to filter out less interesting opinions, this task is truly important as discussed in Section 1. Intuitively, we can think that the features of opinion targets would be similar to those of opinion holders; both would be a noun phrase named entity such as a person or company but would have different syntactic roles in a sentence. An opinion target is likely to be an object whereas an opinion holder is usually the subject of a sentence. Therefore, we can apply the previous approach used for opinion holder identification to this task. Kim and Hovy [12] used statistical classifier to find an opinion holder, which uses a syntactic path from an opinion clue to a candidate holder phrase. The path consists of constituent labels in a parse tree. In addition to utilizing this statistical learning framework with the syntactic features for our task, we add more heuristic features.

We first extract candidate phrases that can be an opinion target using a statistical parser [14]. From the analysis of sample data [7], most targets are a noun phrase. So, we generate a candidate list that contains atomic noun phrases in each opinion; in a parse tree, the first noun phrase from each leaf node can be an atomic noun phrase. Now we can classify each candidate into two categories: a target or not.

In order to train our target classifier, we develop syntactic features as follows. As discussed above, we assume that a syntactic path would be an important feature for target identification. We extract such a syntactic path as follows.

Procedure 1. Recognize the given candidate and opinion clues.

Procedure 2. Generate a syntactic label sequence from S of a parse tree to the candidate.

Procedure 3. Generate a syntactic label sequence from S to each opinion clue.

Procedure 4. Align two label sequences with common labels.

Procedure 5. Eliminate the common labels, except the last one.

We apply the above procedure to the example in Figure 2. We are given "the Yasukuni Shrine" as a target candidate and "views" "aggression" as opinion clues. We first extract the syntactic sequence to the candidate (i.e., S VP NP) and the sequence to the opinion clue (i.e., S VP VBZ). Next, we can align two sequences with the common labels to eliminate them except the last label, VP. As a result, the syntactic path, NP VP VBZ can be obtained. Also, to generalize the path, we normalize verb conjugation and the number of nouns (singular and plural) by WordNet[2], and adjacent syntactic labels are reduced to a single label (i.e., VP VP ➔ VP). Since too many distinctive paths could be harmful to train the classifier, the generalization is necessary. In addition, we generate syntactic dependency features, since opinion clues are dependent on a given target (e.g., "Beijing has opposed to visit Yasukuni Shrine").

Based on the part-of-speech of a clue word, we divide dependencies into 5 different categories, to recognize more diverse cases. In the first category, we can expect a case like "*criticism* of the government's action" (opinion clue is *italicized*), and Category 2 can cover a case like "victim of Russian's *negative* action." Also, a general case like "*attack* China" can be recognized by Category 3. Category 4 contains the example like "*positive* is reducing the cost", and Category 5 is assigned to the case "Japan's decision is *criticized*" or "Japan played a *positive* role" In order to recognize such dependencies, a dependency parser implementing a dependency grammar is utilized.

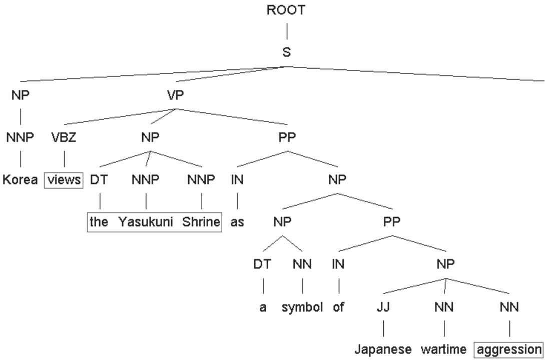

Figure 2. Syntactic Path Example

In addition to the syntactic features, we devised heuristic features. In sample data, we found that a topic term itself could be a target (e.g., "Yasukuni Shrine" can be a target in opinions relevant to the topic of "visiting Yasukuni Shrine"). We extract topic terms for each topic by using n-gram model in Section 3.2. Moreover, as discussed above, named entity such as "Japan" can be a target.

Table 1. Syntactic Dependency Features

No.	Dependency
1	Noun | Adjective clue ➔ target
2	target ➔ Noun | Adjective clue
3	Verb clue ➔ target
4	VB clue ⇦ verb ➔target
5	target ⇦ verb ➔ VB clue

Overall, the feature space for the target classifier consists of (1) syntactic paths, (2) syntactic dependencies (Table 1), (3) whether containing topic words or not, and (4) named entity type, i.e., people, organization, or others.

4. Experimental Results

[2] Lexical database http://wordnet.princeton.edu/

4.1 NTCIR-7 MOAT data collection

The test collection from NTCIR-7 [7] consists of 14 topics (142 documents and 4,312 sentences) in English. The sample data containing three topics (25 documents and 399 sentences) are provided prior to the actual running of our system. In addition, the background information that contains the title and the descriptions is given to each topic. We analyzed the sample data to design our system in advance and then submitted the results for formal running (using NTCIR-7 data except the sample data).

4.2 Results and Discussions

First, we present the results from the testing of our system with the sample data (given in advance), and then the formal running result from the whole NTCIR-7 data. Two different gold-standards are given: lenient and strict. A lenient standard means at least two out of three annotators agreed whereas a strict standard means all 3 agreements agreed. To assess the correctness of target identification, we regard a target as a candidate if annotated text contains the candidate or both headwords (by parsing) are identical. Since semantic matching between two texts is somewhat difficult, we should set reasonable criteria. The overall gold-standard is specified in [7].

Table 2. Opinion Extraction Result in Sample Data

Features	*Precision*	*Recall*	*F-Measure*
All, Presence	0.3050 (0.0%)	0.8404 (0.0%)	0.4476 (0.0%)
$\delta = 0.01$, Presence	0.3317	0.7340	0.4570
$\delta = 0.01$, LM	0.3385	0.7021	0.4567
$\delta = 0.1$, Presence	0.3764	0.6809	0.4848
$\delta = 0.1$, LM	**0.3879 (+27.1%)**	**0.6809 (-19.0%)**	**0.4942 (+10.4%)**

In the opinion extraction task, as specified in Section 3.1 we trained LIBSVM[3] based on NTCIR-6 data that contains 28 topics (439 documents and 8,379 sentences), and applied the classifier to the given sample topics in the NTCIR-7 collection. We tested with the lenient standard, since strict standard in sample data contains too sparse data, not applicable for sufficiently designed testing.

As shown in Table 2, we ran the experiment on opinion extraction to test our divergence based keyword extraction (Figure 1). More specifically, the extraction scheme is tested in several ways: (1) using all the words in opinion sentences vs. only using extracted keywords, (2) extracting many keywords ($\delta = 0.01$) vs. extracting relatively less but effective keywords ($\delta = 0.1$), (3) presence of a keyword vs. language model weighting of a keyword. As shown in Table 2, using all words in opinionated sentences can extract the most opinions (i.e., the highest recall) whereas using higher threshold ($\delta = 0.1$) could not cover many opinionated sentences. However, our algorithm can enhance the precision by reducing feature keywords; about 27% precision enhancement was achieved. Also, using language model probabilities, instead of presence/absence criterion, can slightly increase the performance. However, while the precision has increased, the recall has fallen by about 19%; 0.8404 decreased to 0.6809. Nevertheless, the overall performance (F-Measure) is enhanced by 10.4%. We conclude that optimizing the number of keywords by setting the threshold helps.

Table 3. Topic Relevance Result in Sample Data

Retrieval Model	*R-Precision*
LM without expansion	0.8903
LM with expansion	0.9302

In topic relevance judgments, we tested our snippet-based expansion method (Section 3.2). Since each topic contains different number of extracted opinions, we measured average R-Precision to estimate the retrieval model performance. As shown in Table 3, the retrieval model expanded by the relevant snippets can improve the performance. However, since the sample data contains only 82 opinionated sentences, and only 10 of them are not relevant to their topics, the result seems to be somewhat exaggerated by the sparseness of non-relevant opinions.

The experiment on target identification validates effectiveness of each feature introduced in Section 3.3. In this experiment, we use 3-fold cross validation using the sample data, since this is a new task in NTCIR-7. As shown in Table 4, using only syntactic paths or syntactic dependencies could not make higher performance because the dependency's coverage is limited and the path is less accurate. However, using both features together dramatically enhances the performance in terms of their accuracy and coverage; many positive examples concurrently contain both syntactic paths and dependencies to opinion clues.

[3] Library for SVM http://www.csie.ntu.edu.tw/~cjlin/libsvm

Besides, named entity and topic words can show a little enhancement. However, our classifier (LIBSVM[3]) is weak for the examples that do not contain opinion clues, since we find the paths and dependencies based on the clues.

Table 4. Target Identification Result in Sample Data

Features	*Prec.*	*Recall*	*F-value*
Only SynPath	0.202	0.375	0.220
Only SynDep	0.235	0.191	0.211
SynPath+SynDep	0.732	0.402	0.519
SynPath+SynDep+NE+Tp	**0.810**	**0.448**	**0.577**

Based on the experiment on the sample data, we optimized our system, and submitted the results from the formal running data. As shown in Table 5, the overall performance is much deteriorated, which means that our system is much over-fitted to the sample data. Since we mostly use supervised learning, over-fitting could be harmful. Even though the sample data may inherit the overall nature of the whole data as part of the collection, there would be much difference with such things as unseen keywords, different topics, and different patterns (e.g., paths, dependencies). Moreover, the set of sample data is much smaller than the formal running data, which made such degradation.

Mostly, recall is higher than precision, since as shown in the sample results, the system is somewhat biased to the positive examples. For example, opinion clues are most common in both training set (NTCIR-6) and testing set (NTCIR-7), whereas topic keywords are not shared. So, our system presumptuously judges opinionated sentences if opinion keywords are spotted. However, non-opinionated sentences could contain such keywords, and most misclassified sentences contain topical keywords which are not commonly used in training examples (i.e., topics are different between training and testing examples). In topic relevance decision, the results are also much degraded because the performance in this task is influenced by the performance of the previous task (opinion extraction). Low opinion extraction accuracy means a small number of correct opinions to which the system is limited in finding relevant sentences.

For some topics like “Cosovo Civil war” (Topic No. 5), in addition, we could not obtain sufficient snippets since they are not popular in the web. Besides, synonymy problems are significant. Since some topic words could occur in opinion sentence with different appearances (i.e., synonymy), many errors come from this problem. We used the query expansion to compensate this problem, though. In the task of target identification, we observed rather different results (i.e., our system obtained higher precision than recall). Our system could find relatively correct targets, as long as extracted opinions are correct and types of patterns have been trained. However, the system could not cover many other typed targets, since the learned system would be rather weak for unseen patterns. To generalize this problem, we used NE and topic keywords, but those features are less important than other pattern features (i.e., syntactic path, syntactic dependency), as shown in Table 4. Thus, the sparseness of the sample data (training data) again caused this deterioration.

Table 5. Formal Running Results

Task	*Precision*	*Recall*	*F-value*
Opinion Extraction	0.2435 (0.0743)	0.3687 (0.3777)	0.2933 (0.1241)
Topic Relevance	0.2757 (0.0981)	0.3648 (0.3800)	0.3141 (0.1559)
Target Identification	0.3333 (0.3191)	0.1981 (0.1821)	0.2485 (0.2310)

- Strict cases in parentheses

5. Conclusions

This paper presents a probabilistic model based approach to opinion mining at NTCIR-7 MOAT. For opinion extraction at the sentence level, a probabilistic divergence based feature extraction is proposed. The main thrust of this method is the enhancement of statistical classifier by finding more effective features. Next, we determine the topic relevance of opinionated sentences based on a query likelihood model. To alleviate the expected sparseness problem (as discussed in Section 3.2), we utilize web-snippets. To identify the target of each opinion, we propose a statistical machine learning based approach that exploits syntactic features (syntactic path and dependency) and other heuristic features (topic words and named entity). Although we obtained rather lower performances from formal running (compared to the testing based on the data testing), we can enhance our system by adding some tolerance (e.g., slack variables, more sufficient data, more negative examples) to our classifiers. Thus, as a future study, we intend to manage how to add such tolerance without losing recall and coverage.

Acknowledgements

This work was (partially) supported by the 3rd Phase of Brain Korea 21 Project sponsored by the Ministry of Education and Human Resources Development, Korea. We are grateful to NII who organize the NTCIR workshop and the data.

References

[1] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval. Vol. 2: Issue 1-2, 2008.

[2] E. Riloff, J. Wiebe, and W. Phillips. Exploiting subjectivity classification to improve information extraction. In *Proceedings of 20th Conference on Artificial Intelligence* (*AAAI '05)*, 2005.

[3] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale seasntiment analysis for news and blogs. In *Proceedings of International Conference on Weblogs and Social Media* (*ICWSM '07*), 2007.

[4] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. Exploring sentiment summarization. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*, AAAI technical report SS-04-07, 2004

[5] N. Jindal and B. Liu. Opinion Spam and Analysis. In *Proceedings of Web Search and Data Mining* (*WSDM '08*), 2007.

[6] Y. Kim and S.-H. Myaeng. Opinion Analysis based on Lexical Clues and their Expansion. In *Proceedings of 6th NTCIR Evaluation Workshop Meeting*, Japan, 2007.

[7] Y. Seki. Overview of NTCIR-7 MOAT. In *Proceedings of 7th NTCIR Evaluation Workshop Meeting*, Japan, 2008.

[8] H. Yu and V. Hatzivassiloglou. Towards Answering Opinion Questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (*EMNLP '03*), 2003.

[9] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of 12th International Conference on World Wide Web (WWW '03)*. Budapest, Hungary, 2003.

[10] I. Dagan, L. Lee, and F. Pereira. Similarity-based methods for word sense disambiguation. In *35th Annual meeting of the ACL* (*ACL '03*), 1997.

[11] J. Ponte and W. Croft. A language modeling approach to information retrieval. In *Proceedings of* (*SIGIR '98*), 1998.

[12] S. Kim and E. Hovy. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, Sydney, 2006.

[13] Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. Identifying sources of opinions with conditional random fields and extraction pattern. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (*HLP/EMNLP '05*), 2005.

[14] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of 41st Annual Meeting of the Association for Computational Linguistics* (*ACL '03*), 2003.

[15] A. Esuli and F. Sebastiani. Determining the Semantic Orientation of Terms through Gloss Classification. In *Proceedings of 14th ACM International Conference on Information and Knowledge Management* (*CIKM '05*). 2005.

[16] D. Evans. A low-resources approach to Opinion Analysis: Machine Learning and Simple Approaches. In *Proceedings of the 6th NTCIR Workshop Meeting*, Japan, 2007.

[17] Y. Li, K. Bontcheva and H. Cunningham. Experiments of Opinion Analysis on the Corpora MPQA and NTCIR-6. In *Proceedings of 6th NTCIR Workshop Meeting*, Japan, 2007.

[18] J. Wiebe, R. Wilson, M. Bell, and M. Martin. Development and use of a gold standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, (*ACL '99*), 1999.

[19] Y. Seki, D. Evans, L. Ku, H. Chen, N. Kando, and C. Lin. Overview of Opinion Analysis Pilot Task at NTCIR-6. In *Proceedings of 6th NTCIR Evaluation Workshop*, Japan, 2007.