Extracting Comparative Entities and Predicates from Texts Using Comparative Type Classification

Seon Yang
Department of Computer Engineering,
Dong-A University,
Busan, Korea
seony.yang@gmail.com

Youngjoong Ko
Department of Computer Engineering,
Dong-A University,
Busan, Korea
yjko@dau.ac.kr

Abstract

The automatic extraction of comparative information is an important text mining problem and an area of increasing interest. In this paper, we study how to build a Korean comparison mining system. Our work is composed of two consecutive tasks: 1) classifying comparative sentences into different types and 2) mining comparative entities and predicates. We perform various experiments to find relevant features and learning techniques. As a result, we achieve outstanding performance enough for practical use.

1 Introduction

Almost every day, people are faced with a situation that they must decide upon one thing or the other. To make better decisions, they probably attempt to compare entities that they are interesting in. These days, many web search engines are helping people look for their interesting entities. It is clear that getting information from a large amount of web data retrieved by the search engines is a much better and easier way than the traditional survey methods. However, it is also clear that directly reading each document is not a perfect solution. If people only have access to a small amount of data, they may get a biased point of view. On the other hand, investigating large amounts of data is a time-consuming job. Therefore, a *comparison mining* system, which can automatically provide a summary of comparisons between two (or more) entities from a large quantity of web documents, would be very useful in many areas such as marketing.

We divide our work into two tasks to effectively build a comparison mining system. The first task is related to a sentence classification problem and the second is related to an information extraction problem.

Task 1. Classifying comparative sentences into one non-comparative class and seven comparative classes (or types); *1) Equality, 2) Similarity, 3) Difference, 4) Greater or lesser, 5) Superlative, 6) Pseudo,* and *7) Implicit* comparisons. The purpose of this task is to efficiently perform the following task.

Task 2. Mining comparative entities and predicates taking into account the characteristics of each type. For example, from the sentence "*Stock-X is worth more than stock-Y.*" belonging to "*4) Greater or lesser*" type, we extract "*stock-X*" as a subject entity (SE), "*stock-Y*" as an object entity (OE), and "*worth*" as a comparative predicate (PR).

These tasks are not easy or simple problems as described below.

Classifying comparative sentences (Task 1): For the first task, we extract comparative sentences from text documents and then classify the extracted comparative sentences into seven

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1636–1644,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

comparative types. Our basic idea is a keyword search. Since Ha (1999a) categorized dozens of Korean comparative keywords, we easily build an initial keyword set as follows:

▪ K_{ling} = {"같 ([gat]: same)", "보다 ([bo-da]: than)", "가장 ([ga-jang]: most)", …}

In addition, we easily match each of these keywords to a particular type anchored to Ha's research, e.g., "같 ([gat]: same)" to "*1) Equality*", "보다 ([bo-da]: than)" to "*4) Greater or lesser*". However, any method that depends on just these linguistic-based keywords has obvious limitations as follows:

1) K_{ling} is insufficient to cover all of the actual comparison expressions.
2) There are many non-comparative sentences that contain some elements of K_{ling}.
3) There is no one-to-one relationship between keyword types and sentence types.

Mining comparative entities and predicates (Task 2): Our basic idea for the second task is selecting candidates first and finding answers from the candidates later. We regard each of noun words as a candidate for SE/OE, and each of adjective (or verb) words as a candidate for PR. However, this candidate detection has serious problems as follows:

4) There are many actual SEs, OEs, and PRs that consist of multiple words.
5) There are many sentences with no OE, especially among superlative sentences. It means that the ellipsis is frequently occurred in superlative sentences.

We focus on solving the above five problems. We perform various experiments to find relevant features and proper machine learning techniques. The final experimental results in 5-fold cross validation show the overall accuracy of 88.59% for the first task and the overall accuracy of 86.81% for the second task.

The remainder of the paper is organized as follows. Section 2 briefly introduces related work. Section 3 and Section 4 describe our first task and second task in detail, respectively. Section 5 reports our experimental results and finally Section 6 concludes.

2 Related Work

Linguistic researchers focus on defining the syntax and semantics of comparative constructs. Ha (1999a; 1999b) classified the structures of Korean comparative sentences into several classes and arranged comparison-bearing words from a linguistic perspective. Since he summarized the modern Korean comparative studies, his research helps us have a linguistic point of view. We also refer to Jeong (2000) and Oh (2004). Jeong classified adjective superlatives using certain measures, and Oh discussed the gradability of comparatives.

In computer engineering, we found five previous studies related to comparison mining. Jindal and Liu (2006a; 2006b) studied to mine comparative relations from English text documents. They used comparative and superlative POS tags, and some additional keywords. Their methods applied Class Sequential Rules and Label Sequential Rules. Yang and Ko (2009; 2011) studied to extract comparative sentences in Korean text documents. Li et al. (2010) studied to mine comparable entities from English comparative questions that users posted online. They focused on finding a set of comparable entities given a user's input entity.

Opinion mining is also related to our work because many comparative sentences also contain the speaker's opinion/sentiment. Lee et al. (2008) surveyed various techniques that have been developed for the key tasks of opinion mining. Kim and Hovy (2006) introduced a methodology for analyzing judgment opinion. Riloff and Wiebe (2003) presented a bootstrapping process that learns linguistically rich extraction patterns for subjective expressions.

In this study, three learning techniques are employed: the maximum entropy method (MEM) as a representative probabilistic model, the support vector machine (SVM) as a kernel model, and transformation-based learning (TBL) as a rule-based model. Berger et al. (1996) presented a Maximum Entropy Approach to natural language processing. Joachims (1998) introduced SVM for text classification. Various TBL studies have been performed. Brill (1992; 1995) first introduced TBL and presented a case study on part-of-speech tagging. Ramshaw and Marcus (1995) applied TBL for locating chunks in tagged texts. Black and Vasilakopoulos (2002) used a modified TBL technique for Named Entity Recognition.

3 Classifying Comparative Sentences (Task 1)

We first classify the sentences into *comparatives* and *non-comparatives* by extracting only comparatives from text documents. Then we classify the comparatives into seven types.

3.1 Extracting comparative sentences from text documents

Our strategy is to first detect *Comparative Sentence candidates* (CS-candidates), and then eliminate non-comparative sentences from the candidates. As mentioned in the introduction section, we easily construct a linguistic-based keyword set, K_{ling}. However, we observe that K_{ling} is not enough to capture all the actual comparison expressions. Hence, we build a comparison lexicon as follows:

- *Comparison Lexicon* = K_{ling} U {Additional keywords that are frequently used for actual comparative expressions}

This lexicon is composed of three parts. The first part includes the elements of K_{ling} and their synonyms. The second part consists of idioms. For example, an idiom "X 가 먼저 웃었다 [X-ga meon-jeo u-seot-da]" commonly means "The winner is X" while it literally means "X laughed first". The last part consists of long-distance-words sequences, e.g., "<X 는 [X-neun], 지만 [ji-man], Y 는 [Y-neun], 다 [da]>". This sequence means that the sentence is formed as < S(X) + V + but + S(Y) + V > in English (S: subject phrase; V: verb phrase; X, Y: proper nouns). We could regard a word, "지만 ([ji-man]: but)," as a single keyword. However, this word also captures numerous non-comparative sentences. Namely, the precision value can fall too much due to this word. By using long-distance-words sequences instead of single keywords, we can keep the precision value from dropping seriously low.

The comparison lexicon finally has a total of 177 elements. We call each element "CK" hereafter. Note that our lexicon does not include comparative/superlative POS tags. Unlike English, there is no Korean comparative/superlative POS tag from POS tagger commonly. Our lexicon covers 95.96% of the comparative sentences in our corpus. It means that we successfully defined a comparison lexicon for CS-candidate detection. However, the lexicon shows a relatively low precision of 68.39%. While detecting CS-candidates, the lexicon also captures many non-comparative sentences, e.g., following Ex1:

- Ex1. "내일은 주식이 오를 것 같다." ([nai-il-eun ju-sik-i o-reul-geot gat-da]: I think stock price will rise tomorrow.)

This sentence is a non-comparative sentence even though it contains a CK, "같[gat]." This CK generally means "*same*," but it often expresses "*conjecture*." Since it is an adjective in both cases, it is difficult to distinguish the difference.

To effectively filter out non-comparative sentences from CS-candidates, we use the sequences of "continuous POS tags within a radius of 3 words from each CK" as features. Each word in the sequence is replaced with its POS tag in order to reflect various expressions. However, as CKs play the most important role, they are represented as a combination of their lexicalization and POS tag, e.g., "같/pa[1]." Finally, the feature has the form of "X → y" ("X" means a sequence and "y" means a class; y_1: comparative, y_2: non-comparative). For instance, "<pv etm nbn 같/pa ef sf[2] >→ y_2" is one of the features from Ex1 sentence. Finally, we achieved an f1-score of 90.23% using SVM.

3.2 Classifying comparative sentences into seven types

As we extract comparative sentences successfully, the next step is to classify the comparatives into different types. We define seven comparative types and then employ TBL for comparative sentence classification.

We first define six broad comparative types based on modern Korean linguistics: *1) Equality, 2) Similarity, 3) Difference, 4) Greater or lesser, 5) Superlative, 6) Pseudo* comparisons. The first five types can be understood intuitively, whereas

[1] The POS tag "*pa*" means "the stem of an adjective".
[2] The labels such as "*pv", "etm"* are Korean POS Tags.

the sixth type needs more explanation. "*6) Pseudo*" comparison includes comparative sentences that compare two (or more) properties of one entity such as "Smartphone-X is a computer rather than a phone." This type of sentence is often classified into "*4) Greater or lesser.*" However, since this paper focuses on comparisons between different entities, we separate "*6) Pseudo*" type from "*4) Greater or lesser*" type.

The seventh type is "*7) Implicit*" comparison. It is added with the goal of covering literally "implicit" comparisons. For example, the sentence "Shopping Mall X guarantees no fee full refund, but Shopping Mall Y requires refund-fee" does not directly compare two shopping malls. It implicitly gives a hint that X is more beneficial to use than Y. It can be considered as a non-comparative sentence from a linguistic point of view. However, we conclude that this kind of sentence is as important as the other explicit comparisons from an engineering point of view.

After defining the seven comparative types, we simply match each sentences to a particular type based on the CK types; e.g., a sentence which contains the word "가장 ([ga-jang]: most)" is matched to "*Superlative*" type. However, a method that uses just the CK information has a serious problem. For example, although we easily match the CK "보다 ([bo-da]: than)" to "*Greater or lesser*" without doubt, we observe that the type of CK itself does not guarantee the correct type of the sentence as we can see in the following three sentences:

- Ex2. "X 의 품질은 Y **보다** 좋지도 나쁘지도 않다." ([X-eui pum-jil-eun Y-bo-da jo-chi-do na-ppeu-ji-do an-ta]: The quality of X is neither better nor worse than that of Y.) → It can be interpreted as "The quality of X is **similar** to that of Y." (*Similarity*)
- Ex3. "X 가 Y **보다** 품질이 좋다." ([X-ga Y-bo-da pum-jil-I jo-ta]: The quality of X is better than that of Y.) → It is consistent with the CK type (*Greater or lesser*)
- Ex4. "X 는 다른 어떤 카메라**보다** 품질이 좋다." ([X-neun da-reun eo-tteon ka-me-ra-bo-da pum-jil-i jo-ta]: X is better than any other cameras in quality.) → It can be interpreted as "X is the **best** camera in quality." (*Superlative*)

If we only rely on the CK type, we should label the above three sentences as "*Greater or lesser*". However, each of these three sentences belongs to a different type. This fact addresses that many CKs could have an ambiguity problem just like the CK of "보다 ([bo-da]: than)."

To solve this ambiguity problem, we employ TBL. We first roughly annotate the type of sentences using the type of CK itself. After this initial annotating, TBL generates a set of error-driven transformation rules, and then a scoring function ranks the rules. We define our scoring function as Equation (1):

$$\boldsymbol{Score(r_i) = C_i - E_i} \qquad (1)$$

Here, r_i is the $\boldsymbol{i}$-th transformation rule, $\boldsymbol{C_i}$ is the number of corrected sentences after r_i is applied, and $\boldsymbol{E_i}$ is the number of the opposite case. The ranking process is executed iteratively. The iterations stop when the scoring function reaches a certain threshold. We finally set up the threshold value as 1 after tuning. This means that we use only the rules whose score is 2 or more.

4 Mining Comparative Entities and Predicates (Task 2)

This section explains how to extract comparative entities and predicates. Our strategy is to first detect *Comparative Element candidates* (CE-candidates), and then choose the answer among the candidates.

In this paper, we only present the results of two types: "*Greater or lesser*" and "*Superlative*." As we will see in the experiment section, these two types cover 65.8% of whole comparative sentences. We are still studying the other five types and plan to report their results soon.

4.1 Comparative elements

We extract three kinds of comparative elements in this paper: SE, OE and PR

- Ex5. "X 파이가 Y 파이보다 싸고 맛있다." ([X-pa-i-ga Y-pa-i-bo-da ssa-go mas-it-da]: Pie X is cheaper and more delicious than Pie Y.)
- Ex6. "대선 후보들 중 Z 가 가장 믿음직하다." ([dai-seon hu-bo-deul jung Z-ga ga-jang mit-eum-jik-ha-da]: "Z is the most trustworthy among the presidential candidates.")

In Ex5 sentence, “X 파이 (Pie X)” is a SE, “Y 파이 (Pie Y)” is an OE, and “싸고 맛있다 (cheaper and more delicious)” is a PR. In Ex6 sentence, “Z” is a SE, “대선 후보들 (the presidential candidates)” is an OE, and “믿음직하다 (trustworthy)” is a PR.

Note that comparative elements are not limited to just one word. For example, “싸고 맛있다 (cheaper and more delicious)” and “대선 후보들 (the presidential candidates)” are composed of multiple words. After investigating numerous actual comparison expressions, we conclude that SEs, OEs, and PRs should not be limited to a single word. It can miss a considerable amount of important information to restrict comparative elements to only one word. Hence, we define as follows:

▪ *Comparative elements (SE, OE, and PR) are composed of one or more consecutive words.*

It should also be noted that a number of superlative sentences are expressed without OE. In our corpus, the percentage of the *Superlative* sentences without any OE is close to 70%. Hence, we define as follows:

▪ *OEs can be omitted in the Superlative sentences.*

4.2 Detecting CE-candidates

As comparative elements are allowed to have multiple words, we need some preprocessing steps for easy detection of CE-candidates. We thus apply some simplification processes. Through the simplification processes, we represent potential SEs/OEs as one “N” and potential PRs as one “P”. The following process is one of the simplification processes for making “N”

- Change each noun (or each noun compound) to a symbol “N”.

And, the following two example processes are for “P”.

- Change “pa (adjective)” and “pv (verb)” to a symbol “P”.
- Change “P + ecc (a suffix whose meaning is “and”) + P” to one “P”, e.g., “cheaper and more delicious” is tagged as one “P”.

In addition to the above examples, several processes are performed. We regard all the “N”s as CE-candidates for SE/OE and all the “P”s as CE-candidates for PR. It is possible that a more analytic method is used instead of this simplification task, e.g., by a syntactic parser. We leave this to our future work.

4.3 Finding final answers

We now generate features. The patterns that consist of POS tags, CKs, and “P”/“N” sequences within a radius of 4 POS tags from each “N” or “P” are considered as features.

Original sentence	“X 파이가 Y 파이보다 싸고 맛있다.” (Pie X is cheaper and more delicious than Pie Y.)
After POS tagging	X 파이/nq + 가/jcs + Y 파이/nq + 보다/jca + 싸/pa + 고/ecc + 맛있/pa + 다/ef +./sf
After simplification process	X 파이/N**(SE)** + 가/jcs + Y 파이/N**(OE)** + 보다/jca + 싸고맛있다/P**(PR)** + ./sf
Patterns for SE	<**N(SE)**, jcs, N, 보다/jca,P>, …, <**N(SE)**, jcs>
Patterns for OE	<N, jcs, **N(OE)**, 보다/jca,P, sf>, …, <**N(OE)**, 보다/jca >
Patterns for PR	<N, jcs, N, 보다/jca,**P(PR)**, sf>, …, <**P(PR)**, sf>

Table 1: Feature examples for mining comparative elements

Table 1 lists some examples. Since the CKs play an important role, they are represented as a combination of their lexicalization and POS tag. After feature generation, we calculate each probability value of all CE-candidates using SVM. For example, if a sentence has three “P”s, one “P” with the highest probability value is selected as the answer PR.

5 Experimental Evaluation

5.1 Experimental Settings

The experiments are conducted on 7,384 sentences collected from the web by three trained human labelers. Firstly, two labelers annotated the corpus. A Kappa value of 0.85 showed that it was safe to say that the two labelers agreed in their judgments.

Secondly, the third labeler annotated the conflicting part of the corpus. All three labelers discussed any conflict, and finally reached an agreement. Table 2 lists the distribution of the corpus.

Comparative Types		Sentence Portion
Non-comparative:		5,001 (67.7%)
Comparative:		2,383 (32.3%)
Total (Corpus)		7,384 (100%)
Among Comparative Sentences	*1) Equality*	3.6%
	2) Similarity	7.2%
	3) Difference	4.8%
	4) Greater or lesser	54.5%
	5) Superlative	11.3%
	6) Pseudo	1.3%
	7) Implicit	17.5%
	Total (Comparative)	100%

Table 2: Distribution of the corpus

5.2 Classifying comparative sentences

Our experimental results for Task 1 showed an f1-score of 90.23% in extracting comparative sentences from text documents and an accuracy of 81.67% in classifying the comparative sentences into seven comparative types.

The integrated results showed an accuracy of 88.59%. Non-comparative sentences were regarded as an eighth comparative type in this integrated result. It means that we classify entire sentences into eight types (seven comparative types and one non-comparative type).

5.2.1 Extracting comparative sentences.

Before evaluating our proposed method for comparative sentence extraction, we conducted four experiments with all of the lexical unigrams and bigrams using MEM and SVM. Among these four cases, SVM with lexical unigrams showed the highest performance, an f1-score of 79.49%. We regard this score as our baseline performance.

Next, we did experiments using all of the continuous lexical sequences and using all of the POS tags sequences within a radius of *n* words from each CK as features (*n*=1,2,3,4,5). Among these ten cases, "the POS tags sequences within a radius of 3" showed the best performance. Besides, as SVM showed the better performance than MEM in overall experiments, we employ SVM as our proposed learning technique. Table 3 summarizes the overall results.

Systems	Precision	Recall	F1-score
baseline	87.86	72.57	79.49
comparison lexicon only	68.39	95.96	79.87
comparison lexicon & SVM (proposed)	92.24	88.31	***90.23***

Table 3: Final results in comparative sentence extraction (%)

As given above, we successfully detected CS-candidates with considerably high recall by using the comparison lexicon. We also successfully filtered the candidates with high precision while still preserving high recall by applying machine learning technique. Finally, we could achieve an outstanding performance, an f1-score of 90.23%.

5.2.2 Classifying comparative sentences into seven types.

Like the previous comparative sentence extraction task, we also conducted experiments for type classification using the same features (continuous POS tags sequences within a radius of 3 words from each CK) and the same learning technique (SVM). Here, we achieved an accuracy of 73.64%. We regard this score as our baseline performance.

Next, we tested a completely different technique, the TBL method. TBL is well-known to be relatively strong in sparse problems. We observed that the performance of type classification can be influenced by very subtle differences in many cases. Hence, we think that an error-driven approach can perform well in comparative type classification. Experimental results showed that TBL actually performed better than SVM or MEM.

In the first step, we roughly annotated the type of a sentence using the type of the CK itself. Then, we generated error-driven transformation rules from the incorrectly annotated sentences. Transformation templates we defined are given in Table 4. Numerous transformation rules were generated on the basis of the templates. For example, "Change the type of the current sentence from "*Greater or lesser*" to "*Superlative*" if this sentence holds the CK of "보다 ([bo-da]: than)", and the second preceding word of the CK is tagged as *mm*” is a transformation rule generated by the third template.

Change the type of the current sentence from *x* to *y* if this sentence holds the CK of ***k***, and …
1. the preceding word of ***k*** is tagged ***z***.
2. the following word of ***k*** is tagged ***z***.
3. the second preceding word of ***k*** is tagged ***z***.
4. the second following word of ***k*** is tagged ***z***.
5. the preceding word of ***k*** is tagged ***z*,** and the following word of ***k*** is tagged ***w***.
6. the preceding word of ***k*** is tagged ***z*,** and the second preceding word of ***k*** is tagged ***w***.
7. the following word of ***k*** is tagged ***z*,** and the second following word of ***k*** is tagged ***w***.

Table 4: Transformation templates

For evaluation of threshold values, we performed experiments with three options as given in Table 5.

Threshold	0	1	2
Accuracy	79.99	***81.67***	80.04

Table 5: Evaluation of threshold option (%); Threshold *n* means that the learning iterations continues while C_i-$E_i \geq n+1$

We achieved the best performance with the threshold option 1. Finally, we classified comparative sentences into seven types using TBL with an accuracy of 81.67%.

5.2.3 Integrated results of Task 1

We sum up our proposed method for Task 1 as two steps as follows;

1) The comparison lexicon detects CS-candidates in text documents, and then SVM eliminates the non-comparative sentences from the candidates. Thus, all of the sentences are divided into two classes: a comparative class and a non-comparative class.
2) TBL then classifies the sentences placed in the comparative class in the previous step into seven comparative types.

The integrated results showed an overall accuracy of 88.59% for the eight-type classification. To evaluate the effectiveness of our two-step processing, we performed one-step processing experiments using SVM and TBL. Table 6 shows a comparison of the results.

Processing		**Accuracy**
One-step processing (classifying eight types at a time)	comparison lexicon & SVM	75.64
	comparison lexicon & TBL	72.49
Two-step processing (proposed)		***88.59***

Table 6: Integrated results for Task 1 (%)

As shown above, Task 1 was successfully divided into two steps.

5.3 Mining comparative entities and predicates

For the mining task of comparative entities and predicates, we used 460 comparative sentences (*Greater or lesser*: 300, *Superlative*: 160). As previously mentioned, we allowed multiple-word comparative elements. Table 7 lists the portion of multiple-word comparative elements.

Multi-word rate	**SE**	**OE**	**PR**
Greater or lesser	30.0	31.3	8.3
Superlative	24.4	9.4 (32.6)	8.1

Table 7: Portion (%) of multiple-word comparative elements

As given above, each multiple-word portion, especially in SEs and OEs, is quite high. This fact proves that it is absolutely necessary to allow multiple-word comparative elements. Relatively lower rate of 9.4% in *Superlative*-OEs is caused by a number of omitted OEs. If sentences that do not have any OEs are excluded, the portion of multiple-words becomes 32.6% as written in parentheses.

Table 8 shows the effectiveness of simplification processes. We calculated the error rates of CE-candidate detection before and after simplification processes.

Simplification processes		SE	OE	PR
Greater or lesser	Before	34.7	39.3	10.0
	After	***4.7***	***8.0***	***1.7***
Superlative	Before	26.3	85.0 (38.9)	9.4
	After	***1.9***	***75.6*** ***(6.3)***	***1.3***

Table 8: Error rate (%) in CE-candidate detection

Here, the first value of 34.7% means that the real SEs of 104 sentences (among total 300 *Greater or lesser* sentences) were not detected by CE-candidate detection before simplification processes. After the processes, the error rate decreased to 4.7%. The significant differences between before and after indicate that we successfully detect CE-candidates through the simplification processes. Although the *Superlative*-OEs still show the seriously high rate of 75.6%, it is also caused by a number of omitted OEs. If sentences that do not have any OEs are excluded, the error rate is only 6.3% as written in parentheses.

The final results for Task 2 are reported in Table 9. We calculated each probability of CE-candidates using MEM and SVM. Both MEM and SVM showed outstanding performance; there was no significant difference between the two machine learning methods (SVM and MEM). Hence, we only report the results of SVM. Note that many sentences do not contain any OE. To identify such sentences, if SVM tagged every “N” in a sentence as “not OE”, we tagged the sentence as “no OE”.

Final Results	SE	OE	PR
Greater or lesser	86.00	89.67	92.67
Superlative	84.38	71.25	90.00
Total	**85.43**	**83.26**	**91.74**

Table 9: Final results of Task 2 (Accuracy, %)

As shown above, we successfully extracted the comparative entities and predicates with outstanding performance, an overall accuracy of 86.81%.

6 Conclusions and Future Work

This paper has studied a Korean comparison mining system. Our proposed system achieved an accuracy of 88.59% for classifying comparative sentences into eight types (one non-comparative type and seven comparative types), and an accuracy of 86.81% for mining comparative entities and predicates. These results demonstrated that our proposed method could be used effectively in practical applications. Since the comparison mining is an area of increasing interest around the world, our study can contribute greatly to text mining research.

In our future work, we have the following plans. Our first plan is to complete the mining process on all the types of sentences. The second one is to conduct more experiments for obtaining better performance. The final one is about an integrated system. Since we perform Task 1 and Task 2 separately, we need to build an end-to-end system.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0015613)

References

Adam L. Berger, Stephen A. Della Pietra and Vicent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39-71.

William J. Black and Argyrios Vasilakopoulos. 2002. Language-Independent named Entity Classification by modified Transformation-based Learning and by Decision Tree Induction. In *Proceedings of CoNLL’02*, 24:1-4.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of ANLP’92*, 152-155.

Eric Brill. 1995. Transformation-based Error-Driven Learning and Natural language Processing: A Case Study in Part-of-Speech tagging. *Computational Linguistics*, 543-565.

Gil-jong Ha. 1999a. *Korean Modern Comparative Syntax*, Pijbook Press, Seoul, Korea.

Gil-jong Ha. 1999b. Research on Korean Equality Comparative Syntax, *Association for Korean Linguistics*, 5:229-265.

In-su Jeong. 2000. Research on Korean Adjective Superlative Comparative Syntax. *Korean Han-min-jok Eo-mun-hak*, 36:61-86.

Nitin Jindal and Bing Liu. 2006. Identifying Comparative Sentences in Text Documents, In *Proceedings of SIGIR'06*, 244-251.

Nitin Jindal and Bing Liu. 2006. Mining Comparative Sentences and Relations, In *Proceedings of AAAI'06*, 1331-1336.

Thorsten Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many relevant Features. In *Proceedings of ECML'98*, 137-142

Soomin Kim and Eduard Hovy. 2006. Automatic Detection of Opinion Bearing Words and Sentences. In *Proceedings of ACL'06*.

Dong-joo Lee, OK-Ran Jeong and Sang-goo Lee. 2008. Opinion Mining of Customer Feedback Data on the Web. In *Proceedings of ICUIMC'08*, 247-252.

Shasha Li, Chin-Yew Lin, Young-In Song and Zhoujun Li. 2010. Comparable Entity Mining from Comparative Questions. In *Proceedings of ACL'10*, 650-658.

Kyeong-sook Oh. 2004. The Difference between 'Man-kum' Comparative and 'Cheo-rum' Comparative. *Society of Korean Semantics*, 14:197-221.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Proceedings of NLP/VLC'95*, 82-94.

Ellen Riloff and Janyce Wiebe. 2003. Learning Extraction Patterns for Subjective Expressions. In *Proceedings of EMNLP'03*.

Seon Yang and Youngjoong Ko. 2009. Extracting Comparative Sentences from Korean Text Documents Using Comparative Lexical Patterns and Machine Learning Techniques. In *Proceedings of ACL-IJNLP:Short Papers*, 153-156

Seon Yang and Youngjoong Ko. 2011. Finding relevant features for Korean comparative sentence extraction. *Pattern Recognition Letters*, 32(2):293-296