Natural Language Engineering **20** (4): 557–581. © Cambridge University Press 2013
doi:10.1017/S1351324913000211

Classifying Korean comparative sentences for comparison analysis

SEON YANG and YOUNGJOONG KO*

Department of Computer Engineering, Dong-A University, 840 Hadan 2-dong, Saha-gu, Busan, 604-714, South Korea
e-mail: {seony.yang, youngjoong.ko}@gmail.com

(*Received 11 November 2011; revised 12 July 2013; accepted 26 July 2013; first published online 9 September 2013*)

Abstract

Comparisons sort objects based on their superiority or inferiority and they may have major effects on a variety of evaluation processes. The Web facilitates qualitative and quantitative comparisons via online debates, discussion forums, product comparison sites, etc., and *comparison analysis* is becoming increasingly useful in many application areas. This study develops a method for classifying sentences in Korean text documents into several different comparative types to facilitate their analysis. We divide our study into two tasks: (1) extracting comparative sentences from text documents and (2) classifying comparative sentences into seven types. In the first task, we investigate many actual comparative sentences by referring to previous studies and construct a lexicon of comparisons. Sentences that contain elements from the lexicon are regarded as comparative sentence candidates. Next, we use machine learning techniques to eliminate non-comparative sentences from the candidates. In the second task, we roughly classify the comparative sentences using keywords and use a transformation-based learning method to correct initial classification errors. Experimental results show that our method could be suitable for practical use. We obtained an F1-score of 90.23% in the first task, an accuracy of 81.67% in the second task, and an overall accuracy of 88.59% for the integrated system with both tasks.

1 Introduction

In many areas, comparisons are very important during decision making. For example, politicians may change their political strategies after monitoring how their policies compare with those of their competitors. Manufacturers can also change their marketing strategies after comparing their products with those of their competitors. A similar situation also applies to customers. If a customer is deciding whether to buy *Car-A* or *Car-B*, he or she will probably access the Web and type these two items into the search box. A search engine such as Google will then find relevant documents. Next, the customer will open and read each retrieved document until he or she obtains enough information. The customer's decision may be dominated by sentences that compare these two items. It is clear that obtaining information from

*Corresponding author.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 1. *Problems with a simple keyword search method*

Tasks		Problems
Task 1. Extracting comparative sentences from text documents	Problem 1	The keywords defined using linguistic criteria alone are insufficient to encompass all of the actual comparative expressions.
	Problem 2	Many non-comparative sentences also contain some of the keywords.
Task 2. Classifying comparative sentences into seven types	Problem 3	The keyword types and the sentence types may not be related.

the Web is a good and easy solution. However, it is also clear that reading many documents until sufficient information has been acquired is still a time-consuming task. If the customer only has access to a small amount of data, he or she may form biased views. By contrast, reading large amounts of data demands an enormous amount of time and effort. Therefore, it would be very useful in many areas if a comparative text mining system could access the Web to extract a large volume of comparative data and provide a summary.

The aim of our study is to extract and classify comparative sentences from text documents. We perform various experiments to find relevant features and appropriate learning techniques. We divide our work into two tasks. The first task (Task 1) is to extract comparative sentences from text documents, and the second task (Task 2) is to classify these comparative sentences into seven types: *(1) equality, (2) similarity, (3) difference, (4) greater or lesser, (5) superlative, (6) pseudo,* and *(7) implicit* comparisons. The proposed method starts with keyword searching. Ha (1999a) categorized dozens of comparative Korean keywords, which simplified the construction of the initial keyword set, K_{ling}, as follows:

$$K_{ling} = \{\text{`gat (same),' `bo-da (than),' `ga-jang (most),' } \ldots\}.$$

In addition, we could easily match each of these keywords to a particular type based on Ha's research, e.g., 'gat (same)' to '*(1) equality*,' 'bo-da (than)' to '*(4) greater or lesser*.' However, any method that depends on these keywords alone would have obvious limitations as shown in Table 1.

In Table 1, Problem 1 relates to sentences that do not contain linguistic comparisons although they contain actual comparisons. For example, the sentence: '*In the market share competition, God raised the hand of Samsung*' is not a comparison from a linguistic view, so it cannot be captured by K_{ling}. However, it actually contains a superlative comparison, which can be interpreted as '*Samsung accounted for the largest share of the market*.' Therefore, K_{ling} should be expanded to capture actual comparisons. Problem 2 is related to non-comparative sentences that contain keywords. For example, the sentence: '*They should earn money like ants*' contains a keyword 'cheo-reom (like/as),' but it is not classified as a comparison. This sentence is excluded because it does not compare people ('*they*') with ants on an equal

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

footing, i.e., it does not discuss the similarities or differences between people and ants. The sentence simply likens people to ants for effect. In summary, the keyword 'cheo-reom (like/as)' can be used in comparative and non-comparative sentences (Ha 1999a). Therefore, we need to eliminate non-comparative sentences from the keyword-containing sentences. Problem 3 shows that we cannot identify a sentence type based on keywords alone. For example, we could find the keyword 'bo-da (than)' in '*(2) similarity*' or '*(5) superlative*' sentences, as well as in '*(4) greater or lesser*' sentences (see Section 5.1).

We focus on solving these problems in the present study.

1.1 Task 1: Extracting comparative sentences from text documents

To solve Problem 1, we construct a *comparison lexicon* as follows:

Comparison lexicon = K_{ling} U {Additional keywords that are frequently used in actual comparative expressions (see Section 4.1)}

One or more entries in this lexicon can be detected in most comparative sentences. Thus, the use of this comparison lexicon can solve Problem 1. However, we found that many non-comparative sentences were also captured by the lexicon (Problem 2). The recall was 95.96% whereas the precision was 68.39% when we only used the comparison lexicon to perform Task 1. Thus, the recall was fairly high but the precision was not adequate. To improve the precision, we performed experiments using various features with three learning models: the maximum entropy method (MEM), which is a representative probabilistic model, a support vector machine (SVM) as a kernel model, and transformation-based learning (TBL), which is a rule-based model.

1.2 Task 2: Classifying comparative sentences into seven types

The problem with this task is that we can match each keyword to a particular type manually, but the keyword type is not guaranteed to be the correct sentence type (Problem 3). Investigations of numerous real comparative sentences showed that one keyword could occur in two or more types of sentences (see Section 5.1). Classifying comparative sentences using the keyword information alone delivered relatively poor performance with an accuracy of 64.25%. Thus, we conducted experiments to improve the performance and found that TBL was an appropriate learning method for Task 2. We performed an initial classification using the keyword information alone and improved the accuracy by using TBL to correct the initial classification errors in an iterative manner.

We performed a fivefold cross-validation to ensure a fair evaluation of our method. Given the specificity of our two tasks, we used the F1-score as a measure of Task 1 and the accuracy of Task 2. Our experiments yielded an F1-score of 90.23% during comparative sentence extraction (Task 1) and an accuracy of 81.67% during comparative sentence classification (Task 2). The integrated results of the two tasks had an accuracy of 88.59%. Non-comparative sentences were regarded as an eighth

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

comparative type in this integrated system. Thus, we classified all of the sentences in our corpus into eight types: seven comparative types and one non-comparative type.

The remainder of this paper is organized as follows. Section 2 briefly introduces related work. Section 3 defines the seven comparative types. Sections 4 and 5 describe our approaches to comparative sentence extraction and classification, respectively. Section 6 reports the experimental results in detail. Section 7 presents a discussion of this research and Section 8 provides our conclusions.

2 Related work

Linguistics and computer science are both related to our research. Researchers in linguistics are not interested in the computational identification of comparative sentences in a text document. They focused on defining the syntax and semantics of comparative constructs. Ha (1999a, 1999b) described Korean comparative constructs and summarized modern comparative studies, so his research helped us greatly from a linguistic viewpoint. We also referred to Jeong (2000) and Oh (2004). Jeong classified Korean adjective superlative constructions using four linguistic measures (open-scale, bounded-scale, asymmetric-scale, and unary-scale), and Oh discussed the gradability of Korean comparatives.

To the best of our knowledge, there have been no previous studies of Korean comparison mining systems in computer engineering. The most relevant previous research was a study of English comparison mining by Jindal and Liu (2006a). Their goal was to identify comparative sentences in English text documents. First, they collected 83 keywords, including comparative/superlative part-of-speech (POS) tags, as well as additional words, e.g., 'prefer,' and some key phrases, e.g., 'up against.' Their keyword set captured 94% of the comparative sentences in their annotated corpus. However, only 32% of sentences were genuine comparative sentences. Next, they classified the sentences extracted by the keyword set using class sequential rules (CSRs) and a naïve Bayes classifier. A CSR is a classification rule with a sequence X on the left and a class label y on the right of the rule. Given a CSR '$X \rightarrow y$' and a pair (S, c), which consists of a sequence S and S's class c, the pair is said to *cover* the CSR if X is a subsequence of S, while it is said to *satisfy* the CSR if it covers the CSR and at the same time c is equal to y. Given an annotated sequence set, two parameters, i.e., *support* and *confidence*, are associated with each rule and used to evaluate the reliability of the rule in a sequence set. The two parameters are as follows:

$$\text{support} = \frac{\text{number of sequences that satisfy the rule}}{\text{number of total sequences}},$$

$$\text{confidence} = \frac{\text{number of sequences that satisfy the rule}}{\text{number of the sequences that cover the rule}}.$$

Jindal and Liu finally achieved a precision of 79% and a recall of 81%.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

They also studied comparative relation extraction (Jindal and Liu 2006b). Given a set of comparative sentences, e.g., 'The optics of Canon are better than those of Sony,' they extracted two comparators (Canon and Sony), a feature (optics), and a relation (better) using label sequential rules (LSRs). An LSR is a labeling rule with the form '$X \rightarrow Y$,' where Y is a labeled sequence and X is a sequence produced from Y by replacing some of Y's items with wildcards. The definitions of support and confidence are similar to those given above. LSRs can also be created from an annotated sequence set. Please see Liu (2006) for more detailed information regarding CSR and LSR.

The same techniques can be applied to our research. However, Korean is characterized as an agglutinative language whereas English is an inflecting language, so they differ greatly in many respects. One of the main differences related to our study is that there are POS tags for comparatives and superlatives in English, whereas the POS tagger for Korean does not provide any comparative and superlative tags. Thus, one of the main contributions of our study is to perform Korean comparison mining without using comparative and superlative POS tags. We also added another comparison type, which we refer to as an 'implicit comparison.' We built a set of comparative keywords to ensure high performance of our method, and conducted experiments using various features and learning techniques. We describe this in more detail in the following sections.

In many cases, a comparative sentence contains the speaker's feelings or opinions. Thus, comparison mining is closely related to sentiment analysis or opinion mining. Riloff and Wiebe (2003) presented a bootstrapping process to learn linguistically rich extraction patterns for subjective expressions. Riloff, Wiebe and Wilson (2003) extracted patterns and learned subjective nouns. Kaji and Kitsuregawa (2007) proposed a method for building sentiment lexicons in Japanese. Wiebe, Wilson and Cardie (2005) presented a method for annotating corpus using an annotation scheme. One of the most frequently used subjectivity annotation tools designed for English is OpinionFinder (Wiebe and Riloff 2005), which is based on a large subjectivity lexicon (Wilson, Wiebe and Hoffmann 2005). Wan (2008) proposed the annotation of Chinese reviews using a polarity lexicon as well as a set of negation words and intensifiers. Wan (2009) also proposed the use of machine translation to translate Chinese reviews into English. Esuli and Sebastiani (2006a, 2006b) addressed the task of deciding whether a given term has a positive or negative connotation, or no subjective connotation. The method of Kim and Hovy (2006) involved the annotation of German documents using a lexicon translated from English. Several methods rely on online review repositories, including movie or product reviews in English (Turney 2002; Hu and Liu 2004; Pang and Lee 2004; Ding, Liu and Yu 2008), product reviews in Japanese (Kanayama and Nasukawa 2006), or hotel reviews in Chinese (Li and Sun 2007).

The present study used three learning techniques: MEM, SVM, and TBL. Berger, Della Pietra, A. and Della Pietra, J. (1996) presented a maximum entropy approach to natural language processing (NLP). Joachims (1998) introduced a transductive SVM for text classification. Various TBL studies have been conducted.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 2. *Seven comparative types and example sentences*

Type	Sentence example
Explicit comparison	
(1) Equality	'X-wa Y-neun di-ja-in-i **gat**-da.' (*X and Y have the* **same** *design.*)
(2) Similarity	'X-ui di-ja-in-eun Y-ha-go **bi-seut-ha**-da.' (*The design of X is* **similar** *to that of Y.*)
(3) Difference	'X-neun geu jeom-e-seo Y-wa **da-reu**-da.' (X **differs** from Y in that respect.)
(4) Greater or lesser	'X-cha-eui seung-cha-gam-i Y-cha-**bo-da** bu-deu-reop-da.' (*Car X gives a smoother ride* **than** *Car Y.*)
(5) Superlative	'hu-bo-deul jung-e-seo X-ga **ga-jang** sin-rwe-ga gan-da.' (*X is the* **most** *reliable among the candidates.*)
(6) Pseudo	'pon-X-neun sil-yong-jeok-i-**ra-gi-bo-da** jang-sik-yong-i-da.' (*Phone X is ornamental* **rather than** *practical.*)
Hidden comparison *(7) Implicit*	'syo-ping-mol-X-**neun** su-su-ryo eops-i jeon-aek hwan-bul-eul bo-jang-ha-**ji-man**, syo-ping-mol-Y-**neun** hwan-bul su-su-ryo-reul yo-gu-han-**da**.' (*Shopping Mall X guarantees no fee full refund, but Shopping Mall Y requires refund-fee.*)

Brill (1992, 1995) first introduced TBL and presented a case study of POS tagging. Black and Vasilakopoulos (2002) used a modified TBL technique for named-entity recognition.

3 Comparative type definition

This section defines seven comparative types as shown in Table 2.

Many linguists have studied comparatives. We define six broad types of comparatives, i.e., '*(1) equality*' to '*(6) pseudo*,' based on Ha's research (1999a). We consider that the first five types can be understood intuitively, so we omit any further any explanation of them. However, the sixth type needs more explanation. A pseudo-comparison is sometimes referred to as a *metalinguistic comparison*, which includes comparative sentences that compare two (or more) properties of one target entity such as '*Phone X is ornamental rather than practical.*' The seventh type was added in order to include hidden comparisons. For example, the seventh example sentence given in Table 2 does not compare two shopping malls directly. Instead, it gives an implicit suggestion that the use of *X* is more beneficial than that of *Y*. This sentence can be considered as a non-comparative sentence from a linguistic point of view. However, this type of sentence contains important comparison information in the same way as other explicit comparisons, which is why we included these sentences in the scope of the comparatives.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

4 Task 1: Extracting comparative sentences from text documents

This section describes how our first task is performed by solving problems 1 and 2. First, we construct a *comparison lexicon* to detect comparative sentence candidates (CS candidates), and we use the learning techniques to filter the candidates.

4.1 Constructing the comparison lexicon: Solving Problem 1

As described in Section 1, it was simple to construct K_{ling}, the comparative keyword set, based on Ha's research (1999a). Next, we investigated many actual comparative sentences and expanded K_{ling}. Finally, the comparison lexicon, which comprised three subsets, was constructed as follows:

- The elements of K_{ling} and their synonyms:

  e.g., 'gat (same),' 'ttok-gat (same),' 'dong-il-ha (same),' 'bi-seut-ha (similar),' 'eot-bi-seut-ha (similar),' 'yu-sa-ha (similar),' …

- Idioms:

  e.g., 'X-eui son-eul deul-eo-ju ': This idiom literally means '*raise the hand of X*.' However, it commonly means '*the winner is X*.' For example, the sentence, 'si-jang jeom-yu-yul kyeong-jaeng-e-seo sin-eun sam-sung-eui son-eul deul-eo-ju-eoss-da' (*In the market share competition, God raised the hand of Samsung*) means that Samsung accounted for the largest share of the market.

- Long-distance-words sequences:

  A long-distance-words sequence comprises words that are not consecutive in a sentence.

  e.g., <X-neun, ji-man, Y-neun, 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). The seventh example sentence given in Table 2, 'syo-ping-mol-X-neun su-su-ryo eops-i jeon-aek hwan-bul-eul bo-jang-ha-ji-man, syo-ping-mol-Y-neun hwan-bul su-su-ryo-reul yo-gu-han-da' (*Shopping Mall X guarantees a no-fee full refund, but Shopping Mall Y requires a refund-fee*), contains a long-distance-words sequence. If a single word 'ji-man (but)' is used as a keyword instead of this long-distance-words sequence, it would capture too many non-comparative sentences so the precision value would be too low. The use of long-distance-words sequences prevents the precision value from dropping to an excessively low level.

We collected 70 comparison words, 46 idioms, and 61 long-distance-words sequences. Thus, the final comparison lexicon contained a total of 177 elements. Our lexicon does not include comparative/superlative POS tags. Unlike English, there are no Korean comparative/superlative POS tags. We refer to each element as 'CK' in the following sections. The comparison lexicon had an F1-score of 79.86% (precision:

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 3. *Distribution of the CKs*

Type	Number of CKs	Example CKs (contained in the sentences given in Table 2)
(1) Equality	13 (7.3%)	'gat (same)'
(2) Similarity	16 (9.0%)	'bi-seut-ha (similar)'
(3) Difference	16 (9.0%)	'da-reu (different)'
(4) Greater or lesser	57 (32.2%)	'bo-da (than)'
(5) Superlative	18 (10.2%)	'ga-jang (most)'
(6) Pseudo	1 (0.6%)	'ra-gi-bo-da (rather than)'
(7) Implicit	56 (31.6%)	'<X-neun, ji-man, Y-neun, da> (S(X) V but S(Y) V)'
Total	177 CKs	

68.39%, recall: 95.96%) in Task 1. It should be noted that we obtained an F1-score of 65.95% (precision: 49.98%, recall: 96.90%) when we simply used words such as 'ji-man (but),' 'in-de (but),' 'ban-myeon (on the other hand),' as CKs, instead of struggling to find appropriate long-distance-words sequences. However, using long-distance-words sequences improved the F1-score by nearly 14%.

It should also be noted that the lexicon was constructed before collecting the corpus for evaluation. Thus, a few CKs were not present in the collected corpus, so some sentences annotated as comparatives in the corpus were not captured by the lexicon. These remaining errors in the sentences accounted for 4.04% of the total. '*A squid has ten legs and an octopus has eight legs*' was one of the remaining errors. This sentence did not contain a CK, but it was classified as a comparative sentence by the annotators who concluded that this sentence compared the number of squid legs and that of octopus legs. We will continue to develop methods to capture the remaining errors.

Table 3 shows the distribution of the CKs. As it shows, '*(4) greater or lesser*' accounted for the largest proportion. Indeed, '*greater or lesser*' is the most representative comparative type. We could find many different expressions to compare two or more entities directly based on their superiority or inferiority. The second largest proportion was categorized as '*(7) implicit*.' In addition to the example sentence given in Table 2, we found various sentence structures that contain meaningful implicit comparisons. In order to include these hidden comparisons with different structures, we would need to collect a relatively large number of CKs.

4.2 Filtering CS candidates: Solving Problem 2

The recall of 95.96% with the comparison lexicon was sufficiently high. However, the precision of 68.39% was relatively low and it needed to be improved. During the detection of CS candidates, the lexicon also captured many non-comparative sentences, e.g., 'bi-ga ol geot gat-da (*I think it will rain*).' This sentence is a non-comparative sentence although it contains a CK, 'gat.' This CK generally means

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

'same,' but it often expresses the meaning of 'conjecture.' Since it is an adjective in both cases, it is very difficult to distinguish the difference.

To filter non-comparative sentences from the CS candidates in an effective manner, we tested three learning methods: MEM, SVM, and TBL. These learning methods were used to classify the CS candidates into two classes (*comparative* and *non-comparative*) to eliminate the non-comparative sentences from the candidates. SVM delivered the best performance of these three methods. Thus, we selected SVM as the proposed learning method for Task 1. The features comprised all of the sequences of continuous words within a radius of three words from each CK, i.e., each sub-sequence containing CK within three words on the left and on the right of the CK. Each word in the sequence is replaced with its POS tag to reflect various expressions. However, as CKs play the most important role when identifying comparative sentences, they were represented as a combination of their lexicalization and POS tag, e.g., 'gat/PA.' (The symbol 'PA' is a Korean POS tag, which means 'the stem of an adjective.') Finally, the feature we used for learning is represented as '$X \rightarrow y$,' where X is the sequence described above and y is a class. In the experimental sections, we describe the overall process we used for feature definition and the effectiveness of the proposed features in detail.

5 Task 2: Classifying comparative sentences into seven types

This section describes our second task: classifying comparative sentences into seven comparative types. We address the problem of comparative type ambiguities and use TBL to resolve the ambiguities.

5.1 Resolving the comparative type ambiguities: Solving Problem 3

As mentioned in Section 1, it was easy to match each CK to a particular type, e.g., 'gat (same)' to '*(1) equality*.' However, although the CK type itself can play an important role in sentence classification, a method that uses only the CK information has serious limitations. For example, although the CK 'bo-da (than)' was defined as a suffix that expresses '*(4) greater or lesser*' comparisons in the study of Ha (1999a), it is found often in different types of sentences. In other words, each type of CK is not predetermined and it depends on the specific senses of the CK that need to be assigned. Table 4 shows unmatched cases for the CK type and the sentence type.

If we rely only on the CK type, the above three example sentences should be labeled as the '*(4) greater or lesser*' type. As shown in Table 4, however, the first sentence belongs to the '*(2) similarity*' type and the third sentence belongs to the '*(5) superlative*' type. This addresses the ambiguity of the CK type itself. During the investigation of a large number of actual comparative sentences, we found numerous ambiguous cases. Among 177 CKs, we found 82 CKs in two or more types of sentences and 33 CKs in three or more types of sentences. The CK type with the greatest ambiguity was the '*(3) difference*' type. Among the sentences containing the CKs of the '*(3) difference*' type, almost half of the sentences were tagged as other sentence types. For example, the sentence: '*As I expected, Google is different*

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 4. *The CK, 'bo-da (than),' in three types of sentences*

Major type of the CK	Example sentences containing the CK	Possible sentence types
(4) Greater or lesser	'ka-me-ra-X-eui pum-jil-eun ka-me-ra-Y-bo-da joh-chi-do na-ppeu-ji-do ant-da.' (*The quality of Camera-X is neither better nor worse than Camera-Y.*) ➔ This can be interpreted as '*The quality of Camera X is* **similar** *to that of Camera-Y.*'	*(2) Similarity*
(4) Greater or lesser	'ka-me-ra-X-ga ka-me-ra-Y-bo-da pum-jil-i joh-ta.' (*The quality of Camera-X is better than that of Camera-Y.*)	*(4) Greater or lesser*
(4) Greater or lesser	'ka-me-ra-X-neun da-reun eo-tteon ka-me-ra-bo-da pum-jil-i joh-ta.' (*Camera-X is better than any other cameras in terms of quality.*) ➔ This can be interpreted as '*Camera-X is the* **best** *camera in terms of quality.*'	*(5) Superlative*

from other search sites' that contains the CK 'da-reu (different)' was tagged as a '*(5) superlative*' sentence. Annotators have concluded that the sentence actually means '*I think Google is the best*' after examining the context. These ambiguities made the method using CKs alone deliver poor performance with an accuracy of 64.25%. In the remaining 35.75% of the sentences, the sentence type was different from the CK type. Therefore, an additional process was required to solve this problem.

5.2 Applying the TBL method

We obtained the best performance when we used '*continuous POS tag sequences within a radius of three words from each CK*' as features during comparative sentence extraction (Task 1), so we also performed experiments using the same learning methods and the same features for comparative sentence classification (Task 2). Since TBL produced the best performance, it was selected as the proposed learning method for Task 2.

TBL uses a set of transformation rules to correct the output of an initial classifier. It was first introduced in 1992 (Brill 1992) and it has since been used for various NLP applications. After various experiments, we increased the accuracy value to 81.67%. First, we roughly classified the comparative sentences using the CK type itself. After this initial classification, TBL generated a set of error-driven transformation rules and a scoring function ranked these rules. We defined our scoring function as $Score(r_i) = C_i - E_i$, where r_i is the ith rule, C_i is the number of corrected

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

sentences after r_i is applied, and E_i is the number in the opposite case. The ranking process was executed iteratively. The highest scoring rule was selected repeatedly. The iterations stop when the scoring function reaches a certain threshold. We set the threshold value as 1 after tuning. This means that we only use the rules with a score of 2 or more. We explain the TBL process in detail in the experimental section.

6 Experimental evaluation

6.1 Experimental settings

The corpus used in this paper was constructed from April to August 2008 and it is the first corpus among our datasets used for comparison analysis.[1] It comprised of 277 Web documents. To represent different types of text, we collected data from disparate resources using famous rivals as queries as follows:

- Sites
  - 130 documents from news sites
    - e.g., news.naver.com, media.daum.net, and news.google.co.kr
  - 147 documents from other sites
    - E-market reviews, e.g., www.interpark.com, www.gmarket.co.kr, and www.auction.co.kr
    - Discussion forums, e.g., sports.media.daum.net/ncenter/debate
    - Community-based QA systems, e.g., kin.naver.com and ask.nate.com
    - Personal blogs, e.g., blog.daum.net, section.blog.naver.com, and blog.cyworld.com
- Queries
  - People: '오바마, 클린턴 (Obama, Clinton),' '노무현, 이명박 (Roh Moo-hyun, Lee Myung-bak),' '박근혜, 이명박 (Park Geun Hye, Lee Myung-bak),' '김연아, 아사다마오 (Kim Yu-na, Asada Mao),' '파바로티, 도밍고 (Pavarotti, Domingo),' '소녀시대, 원더걸스 (Girls' Generation, Wonder Girls),' etc.
  - Products or others: '갤럭시, 아이폰 (Galaxy, iPhone),' '구몬, 눈높이 (Kumon, Noonnopi),[2]' '현대차, 기아차 (Hyundai motor, Kia motor),' '삼양라면, 농심라면 (Samyang ramyun, Nongshim ramyun),[3]' '레알마드리드, 바르셀로나 (Real Madrid, Barcelona),' '맨유, 맨시티 (Man Utd, Man City),' etc.

After collecting the documents (a total of 7,384 sentences), two trained annotators tagged the sentences with a Cohen's kappa value of 0.85. The third annotator annotated any conflicting parts of the corpus as an adjudicator. All three annotators discussed any conflict and reached agreements. Figure 1 shows the distribution of the corpus.

[1] http://islab.donga.ac.kr/KorComMinTestSets.htm
[2] Home-study materials.
[3] Instant noodles.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 5. *Example sentence used for Task 1 experiments: this sentence contains a CK, so it was detected by the comparison lexicon as a CS candidate. However, it is not a comparative sentence and it should be eliminated. The feature examples given in Tables 6–9 were extracted from this sentence*

Sentence	'bi-ga ol geot gat-da' (*I think it will rain*)
Class	Non-comparative
CK	'gat (same)'
Lexicalization/POS tag	bi/NCN + ga/JCS + o/PV + l/ETM + geot/NBN + **gat/PA**+ da/EF

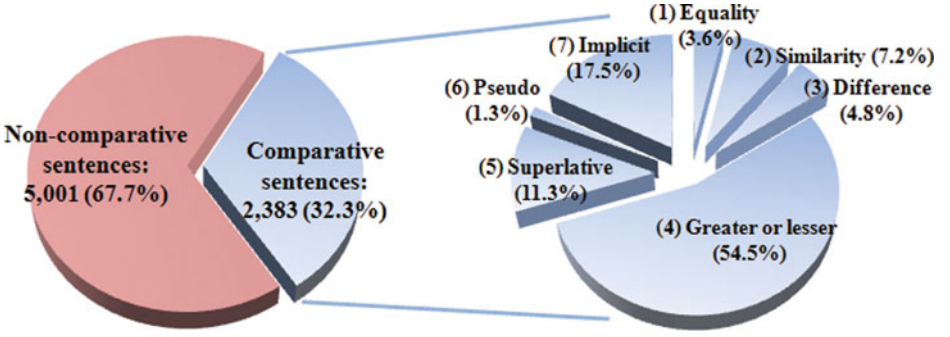


Fig. 1. (Colour online) Distribution of the corpus. The left chart shows the number of comparative sentences and non-comparative sentences in our corpus. The right chart shows the distribution of the seven comparative types.

We performed a fivefold cross-validation in our evaluation. First, our corpus was partitioned into five random subsets. A single subset was retained as the test data to evaluate our methods and the remaining four subsets were used as the training data. The cross-validation process was then repeated five times, where each of the five subsets was used as the test data in rotation. The final five results were averaged. We used a paired *t*-test to test the significance of differences (Refaeilzadeh, Tang and Liu 2009).

6.2 Experiments used to access Task 1

In this section, we present our experimental processes used to extract comparative sentences from the text documents. As described in Section 4, we constructed a comparison lexicon initially. The lexicon included 95.96% of the comparative sentences, which shows that it was suitable for detecting CS candidates. However, the lexicon had a relatively low precision of 68.39%. There were many non-comparative sentences among the candidates, so we had to filter them. Next, we describe our experiments step by step using the example sentence given in Table 5.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 6. *Results obtained with lexical unigrams and bigrams (%)*

Feature	Feature examples	Method	Precision	Recall	F1-score
Lexical unigram	비[bi], 가[ga], 을[ol], 것[geot], 같[gat], 다[da]	MEM	78.17	63.34	69.94
		SVM	87.86	72.57	79.49
Lexical bigram	비가[bi-ga], 가을[ga-ol], 을것[ol-geot], 것같[geot-gat], 같다[gat-da]	MEM	69.59	55.41	61.66
		SVM	80.15	68.26	73.73

6.2.1 Baseline systems used for Task 1

As there were no previous studies related directly to our Korean comparison analysis system, we performed experiments using all of the lexical unigrams and bigrams with MEM and SVM.[4] Table 6 presents the results.

As shown in Table 6, SVM with lexical unigrams delivered the highest performance in the four experiments described above. However, the F1-scores were not higher than those measured based on the comparison lexicon. Thus, the F1-score of 79.86% (precision: 68.39%, recall: 95.96%) achieved using the lexicon was set as our baseline performance level.

6.2.2 Experiments with lexical sequence features

Next, we performed experiments using all of the continuous lexical sequences within a radius of one word from each CK as features. A 'lexical sequence' is a sequence consisting of real words and a 'lexical sequence feature' means that we used the lexical sequence as our learning feature. Table 7 shows the experimental results. As mentioned in Section 4.2, the CK itself was always represented in the form of a 'lexicalization/POS tag.' In the TBL experiment, a feature was an error-driven transformation rule. The non-comparative sentences detected as CS candidates were incorrectly detected sentences. The error-driven transformation rules were created from these incorrect sentences. In other words, the feature <geot gat/PA da> given in Table 7 corresponds to the rule '*Change the class of this sentence from "Comparative" to "Non-comparative" if this sentence contains a continuous lexical sequence <geot gat/PA da>*.'

As shown in Table 7, the lexical sequence features delivered a better performance than the baseline features. A *t*-test of the F1-scores showed that SVM was statistically significantly better than the other two methods ($p < 0.05$). Thus, SVM was selected as the proposed learning method for Task 1 and it was used in all subsequent experiments.

[4] MEM: http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html; SVM: http://www.support-vector-machines.org/S VM_soft.html. A linear kernel with the TFIDF scheme was used for the SVM. We did not optimize any learning parameters. All of the parameters were set to their default values, e.g., the Gaussian prior of MaxEnt = 2, c-value of SVM = 1.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 7. *Results with the lexical sequence features (%)*

Feature	Feature examples	Methods	Precision	Recall	F1-score
Lexical sequence	<것[geot] 같[gat]/PA 다[da]>, <것[geot] 같[gat]/PA>, <같[gat]/PA 다[da]>	MEM	87.14	86.24	86.69
		SVM	87.06	87.65	**87.35**
		TBL	78.19	90.48	83.39

6.2.3 Experiments with POS tag sequence features

The lexical sequence (L), POS tag sequence (P), and a combination of the lexical and POS tag sequences (L&P) were compared to determine the most appropriate choice. With L&P, POS tags were used for noun words. In all other cases, we used lexicalization. The following two sentences explain why we used L&P.

1. 'A-cha ga-gyeok-eun B-cha-man-keum bi-ssa-da' (*The price of Car-A is expensive like that of Car-B*).
2. 'C-hwe-sa jang-nan-gam-eun D-hwe-sa-man-keum bi-ssa-da' (*The toys produced by Company-C are expensive like those produced by Company-D*).

These two sentences are exactly the same if each noun word (underlined part) is replaced by the same symbol. We found a number of such cases. Thus, we replaced every noun word with its POS tag. However, as listed in Table 8, P gave a better performance than L or L&P.

To obtain the POS features, we used the *Dong-A* POS tagger implemented in our laboratory, which had an accuracy of about 94% with the newspaper data and about 90% with other type of multi-domain data. The majority of errors that occurred in this POS tagger were related to the names of persons, organizations, products, etc., i.e., named entities (NEs). We did not evaluate the POS tagging performance using our corpus. However, we tagged the NEs in our corpus separately, so the POS tagging results with the corpus were expected to be relatively high. As shown in Table 8, we obtained the best performance when using the POS tag sequence. The F1-score of the POS tag sequence was statistically significant at $p < 0.01$.

6.2.4 Selecting the best radius length

To determine the best radius length, we conducted the additional experiments as shown in Table 9. Overall, a higher radius length increased the precision value but reduced the recall value. Based on the F1-score, a radius of three words (r3) produced a better performance than r1 and r2, and there were significant differences ($p < 0.01$). By contrast, although the F1-score with r3 was slightly higher than that with r4 and r5, there were no significant differences. Thus, we selected r3, the shortest radius, as our final radius length.

6.2.5 Final results for Task 1

Figure 2 summarizes our experimental results for Task 1.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 8. *Results obtained by comparing the three types of sequence features (%)*

Feature	Feature examples	Precision	Recall	F1-score
Lexical sequence (L)	<것[geot] 같[gat]/PA 다[da]>, <것[geot] 같[gat]/PA>, <같[gat]/PA 다[da]>	87.06	87.65	87.35
POS tag sequence (P)	<NBN 같[gat]/PA EF>, < NBN 같[gat]/PA>, <같[gat]/PA EF>	87.65	88.74	**88.19**
Combination (L&P)	<NBN 같[gat]/PA 다[da]>, <NBN 같[gat]/PA>, <같[gat]/PA 다[da]>	86.85	88.02	87.43

Table 9. *Comparison using five different radius lengths (%)*

Radius lengths	Feature examples	Precision	Recall	F1-score
r1	<NBN 같[gat]/PA EF>, <NBN 같[gat]/PA>, <같[gat]/PA EF>	87.65	88.74	88.19
r2	<ETM NBN 같[gat]/PA>, <NBN 같[gat]/PA EF>, ..., <같[gat]/PA EF>	90.47	88.56	89.50
r3	<PV ETM NBN 같[gat]/PA EF>, <ETM NBN 같[gat]/PA EF>, ..., <같[gat]/PA EF>	92.24	88.31	**90.23**
r4	<JCS PV ETM NBN 같[gat]/PA EF>, <PV ETM NBN 같[gat]/PA EF>, …, <같[gat]/PA EF>	93.48	87.09	90.17
r5	<NCN JCS PV ETM NBN 같[gat]/PA EF>, <JCS PV ETM NBN 같[gat]/PA EF>, …, <같[gat]/PA EF>	94.24	86.35	90.10

We successfully detected the CS candidates using the comparison lexicon and filtered them based on their features in the form of '*a continuous POS tag sequence within a radius of three words from each CK*.' This delivered outstanding results with an F1-score of 90.23%. However, the performance should be improved further. The errors in Task 1 remained the same as errors in Task 2. We will continue to study to improve the performance.

6.3 Experiments used to access Task 2

This section describes the experiments we conducted to access Task 2: classifying the comparative sentences into seven types. We performed various experiments to determine the method that delivered the best performance.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

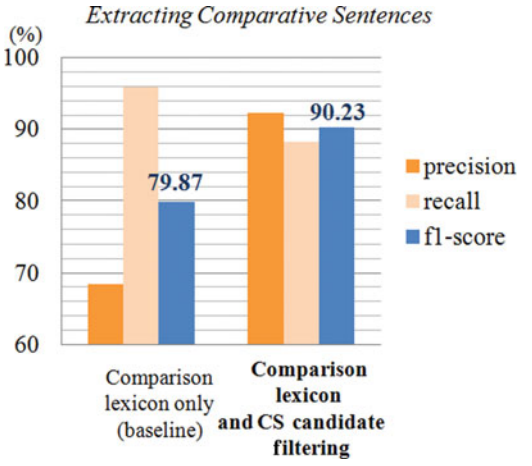


Fig. 2. (Colour online) Experimental results for Task 1.

6.3.1 *Baseline systems for Task 2*

First, we performed experiments using all of the lexical unigrams and bigrams with MEM and SVM in the same way as we did for the previous baseline systems in Task 1. Table 10 presents an example sentence and Table 11 shows the experimental results.

We performed additional experiments. We found that the best performance was obtained when we used '*continuous POS tag sequences within a radius of three words from each CK*' as the features for Task 1, so we also used these same features for Task 2. The results are given in Table 12.

In the TBL experiment, we initially annotated the type of a sentence using the type of the CK itself. Next, we generated error-driven transformation rules from the incorrectly labeled sentences such as the sentence given in Table 10. For example, the feature <ETM MM NCN bo-da/JCA NCN JCS> corresponds to the rule '*Change the type of this sentence from "(4) greater or lesser" to "(6) superlative" if this sentence contains a continuous POS tag sequence <ETM MM NCN bo-da/JCA NCN JCS>*.' Note that we performed fivefold cross-validations for each learning and evaluation step. The TBL rules were also extracted from the four subsets and tested using the remaining one subset. The final five results were finally averaged.

By contrast, MEM/SVM used no initial annotation step because they were originally one-step processes. Thus, we added seven features, from <TYPE1> to <TYPE7>, to indicate the major type of each CK. The type of 'bo-da (than),' i.e., the CK in Table 12, is '*(4) greater or lesser*,' so we added a feature <TYPE4> in the existing features. The performance was improved slightly compared with the method that did not use the CK type feature. However, it was still lower than that with TBL. This showed that TBL was a more suitable learning method than the others. As shown in Tables 11 and 12, TBL delivered the highest performance with an accuracy of 76.85% and there were statistical significances at $p < 0.01$. So the accuracy of 76.85% was set as our baseline performance level.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 10. *Example sentence for Task 2 experiments. The CK contained in this sentence belongs to '(4) greater or lesser' type, so this sentence was classified as the '(4) greater or lesser' type during the initial classification step. However, this sentence actually belongs to the '(5) superlative' type. Thus, it should be classified correctly into the different type. The feature examples given in Tables 11 and 12 were extracted from this sentence*

Sentence	'X-neun da-reun eo-tteon ka-me-ra-bo-da pum-jil-i joh-da' (*X is better than any other cameras in terms of quality*)
The type of sentence	*'(5) Superlative'*
CK	'bo-da (than)'
The type of CK	*'(4) Greater or lesser'*
Lexicalization/POS tag	X/NQ + neun/JXT + da-reu/PA + eun/ETM + eo-tteon/MM + ka-me-ra/NCN + **bo-da/JCA** + pum-jil/NCN + i/JCS + joh/PA + da/EF

Table 11. *Baseline systems used for Task 2 with keywords only or lexical unigrams/bigrams (%)*

Feature	Feature examples	Method	Accuracy
CK only	The default (or major) type of CK	.	64.25
Lexical unigram	X, 는[neum], 다[da], 른[reun], 어[eo], 떤[tteon], 카[ka], 메[me], 라[ra], 보[bo], 다[da], 품[pum], 질[jil], 이[i], 좋[joh], 다[da]	MEM	69.44
		SVM	70.58
Lexical bigram	X 는[X-neun], 는다[neun-da], 다른[da-reun], 른어[reun-eo], 어떤[eo-tteon], 떤카[tteon-ka], 카메[ka-me], 메라[me-ra], 라보[ra-bo], 보다[bo-da], 다품[da-pum], 품질[pum-jil], 질이[jil-i], 이좋[i-joh], 좋다[joh-da]	MEM	69.99
		SVM	70.75

6.3.2 Experiments with TBL

We conducted more TBL experiments to improve the accuracy. We defined 14 transformation templates as listed in Table 13. The first seven templates refer to the CK and its surrounding POS tags and the last seven templates refer to the CK and its surrounding lexical information.

Numerous transformation rules were generated based on the templates given in Table 13. For example, '*Change the type of this sentence from "(4) greater or lesser" to "(6) superlative" if this sentence contains the CK, "bo-da (than)," and the second preceding word of the CK is tagged MM'* is a rule generated by the third template. To verify the effectiveness of the templates, we performed the experiments described in Table 14. We obtained the best performance when we only used POS tags in the template definitions. We performed a *t*-test, which showed that the difference was statistically significant at $p < 0.05$.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 12. *Additional baseline systems for Task 2 using POS tag sequences (%)*

Feature	Feature examples	Method	Accuracy
POS tag sequence	<ETM MM NCN 보다[bo-da]/JCA NCN JCS PA>,	MEM	71.25
	<ETM MM NCN 보다[bo-da]/JCA NCN JCS>, …,	SVM	73.64
	<보다[bo-da]/JCA NCN>	**TBL**	**76.85**
CK type, POS tag sequence	<TYPE4>, <ETM MM NCN 보다[bo-da]/JCA NCN JCS PA>, <ETM MM NCN 보다[bo-da]/JCA NCN JCS>, …, <보다[bo-da]/JCA NCN>	MEM	72.00
		SVM	73.98

Table 13. *Definitions of TBL templates*

Change the type of the sentence from x to y if this sentence contains the CK, $\boldsymbol{k}$, and …	
1. the preceding word of $\boldsymbol{k}$ is tagged $\boldsymbol{z}$.	*POS tags*
2. the following word of $\boldsymbol{k}$ is tagged $\boldsymbol{z}$.	
3. the second preceding word of $\boldsymbol{k}$ is tagged $\boldsymbol{z}$.	
4. the second following word of $\boldsymbol{k}$ is tagged $\boldsymbol{z}$.	
5. the preceding word of $\boldsymbol{k}$ is tagged $\boldsymbol{z}$, and the following word of $\boldsymbol{k}$ is tagged $\boldsymbol{w}$.	
6. the preceding word of $\boldsymbol{k}$ is tagged $\boldsymbol{z}$, and the second preceding word of $\boldsymbol{k}$ is tagged $\boldsymbol{w}$.	
7. the following word of $\boldsymbol{k}$ is tagged $\boldsymbol{z}$, and the second following word of $\boldsymbol{k}$ is tagged $\boldsymbol{w}$.	
8. the preceding word of $\boldsymbol{k}$ is $\boldsymbol{z}$.	*Lexical information*
9. the following word of $\boldsymbol{k}$ is $\boldsymbol{z}$.	
10. the second preceding word of $\boldsymbol{k}$ is $\boldsymbol{z}$.	
11. the second following word of $\boldsymbol{k}$ is $\boldsymbol{z}$.	
12. the preceding word of $\boldsymbol{k}$ is $\boldsymbol{z}$, and the following word of $\boldsymbol{k}$ is $\boldsymbol{w}$.	
13. the preceding word of $\boldsymbol{k}$ is $\boldsymbol{z}$, and the second preceding word of $\boldsymbol{k}$ is $\boldsymbol{w}$.	
14. the following word of $\boldsymbol{k}$ is $\boldsymbol{z}$, and the second following word of $\boldsymbol{k}$ is $\boldsymbol{w}$.	

Therefore, we selected seven templates as our TBL transformation templates (templates 1–7), which were related to '*POS tags within a radius of two words from each CK*.'

6.3.3 Selecting the best threshold option

We performed experiments to evaluate the threshold value used in the scoring function. Table 15 shows the changes in the accuracy values based on different threshold values.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 14. *Evaluations of the TBL templates given in Table 13 (%)*

Templates	Number of the templates	Accuracy
All fourteen templates	14	78.22
POS tags (templates 1–7)	**7**	**79.99**
Lexical information (templates 8–14)	7	79.01

Table 15. *Results of the comparison of the three threshold options (%)*

Threshold	Accuracy
Threshold = 0 ($C_i - E_i \geq 1$)	79.99
Threshold = 1 ($C_i - E_i \geq 2$)	**81.67**
Threshold = 2 ($C_i - E_i \geq 3$)	80.34

We obtained the best result when we used a threshold value of 1. The difference in the accuracy was significant at $p < 0.01$. We considered that rules with a score of 1 could cause overfitting problems.

6.3.4 Final results for Task 2

In the final results, we classified the comparative sentences into seven types with an accuracy of 81.67%. Table 16 shows the precision, recall, and F1-score for each of the seven types.

It should be noted that the sixth type, '*(6) pseudo*,' had an F1-score of 100% in our corpus. The only CK of that type, 'ra-gi-bo-da (rather than),' has extremely limited usage. As far as we know, this CK is used only for '*(6) pseudo*' comparisons. It should also be noted that the seventh type, *'(7) implicit*,' had the lowest accuracy. We consider that this type is relatively difficult to analyze automatically because sentences that contain hidden comparisons have a diverse range of language structures. Figure 3 summarizes our experimental results for Task 2.

6.4 Experiments based on system integration

This section shows how the two tasks were integrated. In this end-to-end system, the non-comparative sentences belonged to a special comparative type (the eighth type). In other words, we classified all of the sentences in the corpus into eight types: one non-comparative type and seven comparative types. Our proposed method is summarized as follows:

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

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 16. *Results with each of the seven comparative types (%)*

Type	Precision	Recall	F1-score
(1) Equality	54.55	80.00	64.86
(2) Similarity	85.71	69.23	76.60
(3) Difference	69.25	82.94	75.48
(4) Greater or lesser	86.76	90.48	88.58
(5) Superlative	78.95	83.33	81.08
(6) Pseudo	100.00	100.00	100.00
(7) Implicit	69.39	55.74	61.82
Total	81.67 (micro-averaging)		

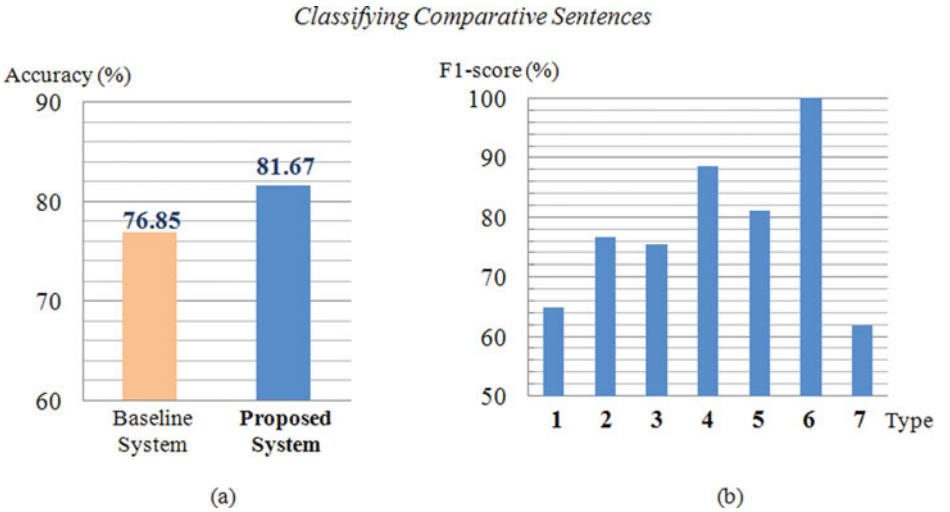


Fig. 3. (Colour online) Experimental results for Task 2: (a) the performance with our proposed system (b) the F1-score with each comparative type.

The results using the integrated system had an accuracy of 88.59% with classification based on eight types. To evaluate the effectiveness of our two-step processing method, we performed one-step processing experiments. Figure 4 shows a comparison of the results, where 'Lexicon and SVM' indicates that we detected the CS candidates initially using the comparison lexicon and then classified the candidates into eight types using SVM with the features employed in Task 1 (continuous POS tag sequences within a radius of three words from each CK). Lexicon and TBL indicates that we detected CS candidates by the comparison lexicon and then classified the candidates into eight types using TBL with the rules generated by the same templates as employed in Task 2 (the first seven templates given in Table 13).

As shown in Figure 4, our method was divided into two steps successfully.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

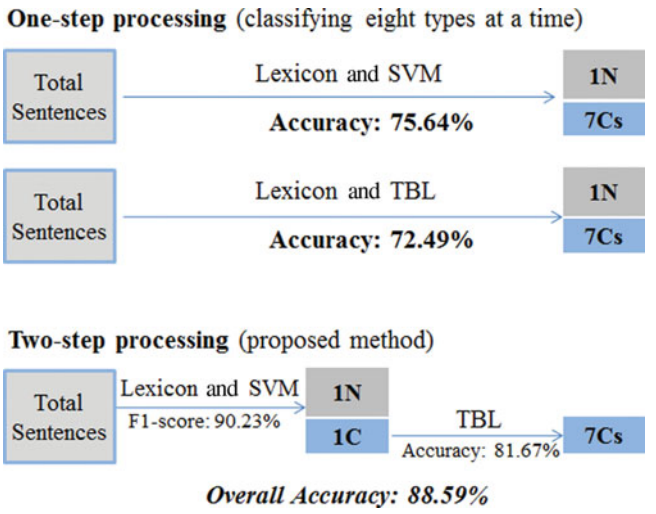


Fig. 4. (Colour online) Final results with system integration: '1N' denotes that there was a single non-comparative type, '7C' denotes the seven comparative types, and '1C' denotes all of the comparative sentences.

7 Discussion

7.1 What was the motivation for this study?

In many cases, people might choose only one of two (or three or more) objects. For example, if you were trying to decide whether to buy phone A or phone B, you would probably look for information on the Web to assist with your choice. Typing these two items into a search engine such as Google, Yahoo, or Naver would allow you to find documents related to these items. You could then open and read each document retrieved until you had enough information to make an informed decision about the best item to buy. It is obvious that obtaining information from the Web is simpler than previous methods such as surveys, asking people directly, and reading books. However, it is also clear that reading each Web document would still be a time-consuming task. Therefore, a comparison analysis system that is capable of providing a summary of the comparisons between two entities would be very useful in many areas such as marketing. An example of the summary results provided by our final comparison system is shown in Table 17.

The generation of specific information such as 'Phone A is better' is the final goal of our comparison mining study. To this end, we should do 'Task 3' that is beyond the scope of the present study. To complete Task 3, we will need to extract comparable entities (e.g., phone A and phone B), properties (e.g., sound quality, design, and A/S), and other comparative relations. Task 3 is an information extraction task, which is ongoing at present.

7.2 Summary of the entire process employed by the proposed system

This subsection provides a summary of the overall process and its implementation as follows:

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 17. *An example of the summary results provided by our final comparison system*

Property	Comparative type (input: *phone A, phone B*)	Output, %
Sound quality	Equality	*n*%
	Phone A is better	*n*%
	Phone B is better	*n*%
Design	Similarity	*n*%
	Phone A is better	*n*%
	Phone B is better	*n*%
A/S	…	…
…	…	…
Overall	…	…

▶ Experimental environment

- Tool: POS tagger, machine learning tools
  - We used SVM and TBL in experiments to identify the best combination for usage in our system.
- Pre-task: Defining comparative types
  - We defined seven types; six types based on Korean linguistics and an additional type based on practical requirements.
- Resource: Comparison lexicon, corpus
  - The comparison lexicon originally contained linguistic-based keywords but was expanded using actual comparison-related expressions.
  - The corpus consisted of Web documents. Each sentence was annotated as a comparative or a non-comparative and each comparative sentence was tagged with a corresponding comparative type.
- Feature: We obtained a good performance using '*POS tag sequences within a certain radius from each CK*.'

▶ System procedure (see Table 18).

8 Conclusions and future work

In this study, we developed a method for automatically identifying different types of Korean comparative sentences. In our experiments, we obtained an F1-score of 90.23% when extracting comparative sentences from text documents and an accuracy of 81.67% when classifying comparative sentences into seven types. An integrated system that combined the two processes delivered a high accuracy of 88.59%. These results demonstrate that our proposed method could be used in practical applications. No previous studies have addressed the automatic identification Korean comparative sentences, so our method may contribute to Korean text analysis research. In addition, since the comparison analysis research is currently in its early phase throughout the world, our proposed techniques may also contribute greatly to the information technology (IT) industry.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Table 18. *Summary of the overall procedure*

Phase	Process
Setting	1. Defining comparative types
	·→ 2. Building a comparison lexicon
	·→ 3. Building the corpus
	·→ 4. Attaching POS tags to the corpus
Task 1	·→ 5. Detecting CS candidates using the comparison lexicon
	·→ 6. Extracting features from the CS candidates
	·→ 7. Classifying the CS candidates into two classes, i.e., *comparative* and *non-comparative*, using machine learning techniques (SVM)
Task 2	·→ 8. Extracting features (or TBL rules) from the comparative sentences
	·→ 9. Classifying comparative sentences into comparative types using machine learning techniques (TBL)

In our future research, we aim to improve the comparative sentence identification results further. In particular, we will investigate implicit comparisons in greater depth. We also plan to extract comparative relations such as holders, entities, and properties. The appropriate analysis of these comparative relations will allow us to construct a complete comparison analysis system.

Acknowledgements

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0065279).

References

Berger, A. L., Della Pietra, S. A., and Della Pietra, V. J. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics* **22**(1): 39–71.

Black, W. J., and Vasilakopoulos, A. 2002. Language-independent named entity classification by modified transformation-based learning and by decision tree induction. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, Taipei, Taiwan, vol. 20, pp. 1–4.

Brill, E. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural language Processing (ANLP 1992)*, Trento, Italy, pp. 152–5.

Brill, E. 1995. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Computational Linguistics* **21**(4): 543–65.

Ding, X., Liu, B., and Yu, P. S. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of ACM International Conference on Web Search and Data Mining (WSDM 2008)*, Stanford, USA, pp. 231–40.

Esuli, A., and Sebastiani, F. 2006a. Determining term subjectivity and term orientation for opinion mining. In *Proceedings of European Chapter of the Association for Computational Linguistics (EACL 2006)*, Trento, Italy, pp. 193–200.

Esuli, A., and Sebastiani, F. 2006b. SentiWordNet: a publicly available lexical resource for opinion mining. In *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Ha, G.-j. 1999a. *Korean Modern Comparative Syntax*. Seoul, Korea: Pijbook Press.
Ha, G.-j. 1999b. Research on Korean equality comparative syntax. *Association for Korean Linguistics* **5**: 229–65.
Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2004 (KDD 2004)*, Seattle, USA, pp. 168–77.
Jeong, I.-s. 2000. Research on Korean adjective superlative comparative syntax. *Korean Han-min-jok Eo-mun-hak* **36**: 61–86.
Jindal, N., and Liu, B. 2006a. Identifying comparative sentences in text documents. In *Proceedings of Association for Computing Machinery/Special Interest Group on Information Retrieval (SIGIR 2006)*, Seattle, USA, pp. 244–51.
Jindal, N., and Liu, B. 2006b. Mining comparative sentences and relations. In *Proceedings of Association for Advancement of Artificial Intelligence (AAAI 2006)*, Boston, USA, pp. 1331–6.
Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of European Conference on Machine Learning (ECML 1998)*, Chemnitz, Germany, pp. 137–42.
Kaji, N., and Kitsuregawa, M. 2007. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, Czech Republic, pp. 1075–83.
Kanayama, H., and Nasukawa, T. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia, pp. 355–63.
Kim, S.-M. and Hovy, E. 2006. Identifying and analyzing judgment opinions. In *Proceedings of the Human Language Technology Conference – North American Chapter of the Association for Computational Linguistics (NAACL 2006)*, New York City, USA, pp. 200–207.
Li, J., and Sun, M. 2007. Experimental study on sentiment classification of Chinese review using machine learning techniques. In *International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 2007)*, Beijing, China, pp. 393–400.
Liu, B. 2006. *Web Data Mining*. New York City, USA: Springer.
Oh, K.-s. 2004. The difference between 'Man-kum' comparative and 'Cheo-rum' comparative. *Society of Korean Semantics* **14**: 197–221.
Pang, B., and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Barcelona, Spain, pp. 271–8.
Refaeilzadeh, P., Tang, L., and Liu, H. 2009. Cross-validation. In *Encyclopedia of Database Systems*, pp. 532–8. New York City, USA: Springer.
Riloff, E., and Wiebe, J. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2003)*, Sapporo, Japan, pp. 105–12.
Riloff, E., Wiebe, J., and Wilson, T. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, New York City, USA, pp. 25–32.
Turney, P. D. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, USA, pp. 417–24.
Wan, X. 2008. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, Waikiki, Hawaii, USA, pp. 553–61.
Wan, X. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the Association of Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL–IJCNLP 2009)*, Singapore, pp. 235–43.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press

Wiebe, J., and Riloff, E. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*, Mexico City, Mexico, pp. 486–97.

Wiebe, J., Wilson, T., and Cardie, C. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* **39**: 165–210.

Wilson, T., Wiebe, J., and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, Canada, pp. 347–54.

https://doi.org/10.1017/S1351324913000211 Published online by Cambridge University Press