Identifying Comparative Sentences in Text Documents

Nitin Jindal and Bing Liu
Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street
Chicago, IL 60607-7053

{njindal, liub}@cs.uic.edu

ABSTRACT

This paper studies the problem of identifying comparative sentences in text documents. The problem is related to but quite different from sentiment/opinion sentence identification or classification. Sentiment classification studies the problem of classifying a document or a sentence based on the subjective opinion of the author. An important application area of sentiment/opinion identification is business intelligence as a product manufacturer always wants to know consumers' opinions on its products. Comparisons on the other hand can be subjective or objective. Furthermore, a comparison is not concerned with an object in isolation. Instead, it compares the object with others. An example opinion sentence is "*the sound quality of CD player X is poor*". An example comparative sentence is "*the sound quality of CD player X is not as good as that of CD player Y*". Clearly, these two sentences give different information. Their language constructs are quite different too. Identifying comparative sentences is also useful in practice because direct comparisons are perhaps one of the most convincing ways of evaluation, which may even be more important than opinions on each individual object. This paper proposes to study the comparative sentence identification problem. It first categorizes comparative sentences into different types, and then presents a novel integrated pattern discovery and supervised learning approach to identifying comparative sentences from text documents. Experiment results using three types of documents, news articles, consumer reviews of products, and Internet forum postings, show a precision of 79% and recall of 81%. More detailed results are given in the paper.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering*. I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis*.

General Terms

Algorithms, Performance.

Keywords

Comparative sentences, sentiment classification, text mining.

SIGIR'06, August 6-11, 2006, Seattle, Washington, USA.

1. INTRODUCTION

Comparisons are one of the most convincing ways of evaluation. Extracting comparative sentences from text is useful for many applications. For example, in the business environment, whenever a new product comes into market, the product manufacturer wants to know consumer opinions on the product, and how the product compares with those of its competitors. Much of such information is now readily available on the Web in the form of customer reviews, forum discussions, blogs, etc. Extracting such information can significantly help businesses in their marketing and product benchmarking efforts. In this paper, we focus on comparisons. Clearly, product comparisons are not only useful for product manufacturers, but also to potential customers as they enable customers to make better purchasing decisions.

In the past few years, a significant amount of research was done on sentiment and opinion extraction and classification. In Section 2, we will discuss the existing literature and compare it with our work, where related research from linguistics is also included. Comparisons are related but also quite different from sentiments and opinions, which are subjective. Comparisons on the other hand can be subjective or objective. For example, an opinion sentence on a car may be "*Car X is very ugly*". A subjective comparative sentence may be

> "*Car X is much better than Car Y*"

An objective comparative sentence may be

> "*Car X is 2 feet longer than Car Y*"

We can see that in general comparative sentences use quite different language constructs from typical opinion sentences (although the first sentence above is also an opinion). In this paper, we aim to study the problem of identifying comparative sentences in text documents, e.g., news articles, consumer reviews of products, forum discussions. This problem is challenging because although we can see that the above example sentences all contain some indicators (comparative adverbs and comparative adjectives), i.e., "better", "long*er*", many sentences that contain such words are not comparatives, e.g., "*I cannot agree with you more*". Similarly, many sentences that do not contain such indicators are comparative sentences, e.g., "*Cellphone X has Bluetooth, but cellphone Y does not have*."

In this paper, we first classify comparative sentences into different categories based on existing linguistic research. We also expand them with additional categories that are important in practice. We then propose a novel approach based on pattern discovery and supervised learning to identify comparative sentences. The basic idea of our technique is to first use a keyword strategy to achieve a high recall, and then build a machine learning model to automatically classify each sentence into one of the two classes, "*comparative*" and "*non-comparative*", based on the filtered data to improve the precision. In building the learning model, *class sequential rules* automatically generated from the data are used as features.

Class sequential rules are different from traditional sequential patterns [1, 2, 25] because a class label is attached, which results in a rule with a sequential pattern on the left-hand-side of the rule, and a class on the right-hand-side of the rule. In our context, the classes are "*comparative*" or "*non-comparative*". Our system is able to generate such rules. A further advancement of our work is the use of multiple minimum supports in mining. Existing sequential pattern mining techniques in data mining use only a single minimum support [1] to control the pattern generation process so that not too many patterns are produced. The minimum support is simply the probability that a pattern appears in a sentence, which is estimated as the ratio of the number of sentences containing the pattern and the total number of sentences in the data. The single minimum support model from data mining is not sufficient for our work because some patterns (although very accurate) appear much less frequently in the data than others. If the minimum support is set very low in order to catch these infrequent patterns, the frequent words will produce a huge number of spurious patterns, which harm the classification because they may overfit the data. The multiple minimum supports model deals with this problem effectively.

Our experiment results confirmed that class sequential rules are highly useful for the final classification. Since each class sequential rule has a class confidence, which is the conditional probability that a sentence is a comparative sentence given that it contains the sequential pattern on the left-hand-side, the rules can naturally be used for classification because confidence is basically a measure of predictability. However, our results show that they do not perform well. The classifier built using a learning algorithm based on the class sequential rules performs much better. The key reason for the weaker performance of the rules is that a given sentence may satisfy several rules and the rules may conflict with each other, e.g., one rule says that a sentence is a comparative sentence, but another says it is a non-comparative sentence. To deal with such conflicts, a principled method is needed to combine all the rules to arrive at a single classification. The naïve Bayesian model provides a natural solution.

We then used the naïve Bayesian model [20, 21] to learn a classifier using the class sequential rules as features. For comparison purposes, we also experimented with support vector machines (SVM) [31, 13], which is considered to be one of the strongest classifier building methods. We conducted empirical evaluation using three types of documents, news articles, consumer reviews of products, and Internet forum discussions. Our task is to identify comparative sentences in such input texts. Our experimental results based on 5-fold cross validation show an overall precision of 79% and an overall recall of 81%. The naïve Bayesian classifier outperforms SVM. Detailed results and comparisons are presented and discussed in Section 5.

We note that after comparative sentences are identified, extracting the comparative relation in each sentence is also of great importance, i.e., what are compared and which is better. This extraction problem is studied in [12].

In summary, this paper makes three contributions:

1. It proposes to study the problem of identifying comparative sentences in text. To the best of our knowledge, there is no reported study on this problem so far. Although linguists have studied the semantics of some comparative constructs, their work is more for human understanding and thus not directly applicable to our task. There is no reported computational method to identify comparative sentences.
2. A categorization of comparative sentences into different types is proposed based on the linguistic research. However, we extended the existing types by including some more which are also important in practice.
3. We propose an effective approach to solve the problem based on class sequential rules and the machine learning technology. Experiment results demonstrated the effectiveness of the proposed approach.

This paper is organized as follows: The next section discusses the related work. Section 3 gives the problem statement and categorizes different types of comparative sentences, which expands what are available from linguistics. Section 4 presents the proposed technique. Section 5 evaluates it and Section 6 concludes the paper and discusses future directions.

2. RELATED WORK

Related work to ours comes from both computer science and linguistics. Researchers in linguistics focus primarily on defining the syntax and semantics of comparative constructs. They do not deal with the identification of comparative sentences from a text document computationally. [22] studies the semantics and syntax of comparative sentences, but uses only limited vocabulary. It is not able to do our task of identifying comparative sentences. [14] discusses gradability of comparatives and measure of gradability. The semantic analysis is based on logic, which is not directly applicable to identifying comparative sentences. The types of comparatives (such as adjectival, adverbial, nominal, superlatives, etc) are described in [6]. The focus of these researches is on a limited set of comparative constructs which have gradable keywords like more, less, etc. In summary, although linguists have studied comparatives, their semantic analysis of comparatives based on logic and grammars is more for human consumption than for automatic identification of comparative sentences by computers.

In text and data mining, we have not found any direct work on comparative sentences. The most closely related work is sentiment classification and opinion extraction, which as we pointed out in the introduction section are related but quite different from our work.

Sentiment classification classifies opinion texts or sentences as positive or negative. Work of Hearst [10] on classification of entire documents uses models inspired by cognitive linguistics. Das and Chen [4] use a manually crafted lexicon in conjunction with several scoring methods to classify stock postings. Tong [29] generates sentiment (positive and negative) timelines by tracking online discussions about movies over time.

[30] applies a unsupervised learning technique based on mutual information between document phrases and the words "excellent" and "poor" to find indicative words of opinions for classification.

[24] examines several supervised machine learning methods for sentiment classification of movie reviews. [5] also experiments a number of learning methods for review classification. They show that the classifiers perform well on whole reviews, but poorly on sentences because a sentence contains much less information.

[9] investigates sentence subjectivity classification. A method is proposed to find adjectives that are indicative of positive or negative opinions. [32] proposes a similar method for nouns. Other related works on sentiment classification and opinions discovery include [9, 15, 16, 23, 27, 33, 34, 35].

In [11, 19], several unsupervised and supervised techniques are proposed to analyze opinions in customer reviews. Specifically, they identify product features that have been commented on by customers and determining whether the opinions are positive or negative. [26, 8] improve the work in [11]. However, none of these studies is on comparison, which is the focus of this work.

3. PROBLEM DEFINITION

In this section, we state the problem that we aim to solve. We first give a linguistic view of *comparatives* (also called *comparative constructions*) and identify some limitations. We then enhance them by including implicit comparatives, and state the problem that we deal with in this paper.

Since we need Part-Of-Speech (POS) tags throughout this section and the paper, let us first acquaint ourselves with some tags and their POS categories. We used Brill's Tagger [3] to tag sentences. It follows the Penn Tree Bank [28] POS Tagging Scheme. The POS tags and their categories that are important to this work are: *NN*: Noun, *NNP*: Proper Noun, *VBZ*: Verb, present tense, 3rd person singular, *JJ*: Adjective, *RB*: Adverb, *JJR*: adjective, comparative, *JJS*: adjective, superlative, *RBR*: Adverb, comparative, *RBS*: Adverb, superlative.

3.1 Linguistic Perspective

Linguists have studied comparatives in the English language for a long time. [17] defines comparatives as universal quantifiers over degrees. For example, "*John is taller than he was*", the degree *d* is John's height and John is tall to degree *d*. In other words, comparatives are used to express explicit orderings between objects with respect to the degree or amount to which they possess some gradable property [14]. The two broad types of comparatives as given in [6] are:

1) *Metalinguistic Comparatives*: Those which compare the extent to which an entity has one property to a greater or lesser extent than another property. For example, "*Ronaldo is angrier than upset*."
2) *Propositional Comparatives*: Those that make a comparison between two propositions. This category has subcategories:
   a. *Nominal Comparatives:* They compare the cardinality of two sets of entities denoted by nominal phrases. Ex: "*Paul ate more grapes than bananas*"
   b. *Adjectival Comparatives:* They usually have words that end with *–er*, *more*, *less*, etc. (occurring with the conjugate *than*) and equative *as* (ex: *as good as*). Ex: "*Ford is cheaper than Volvo*."
   c. *Adverbial Comparatives:* They are similar to nominal and adjectival ones except that they generally occur after a verb phrase. Ex: "*Tom ate more quickly than Jane*."

Then there are superlatives which are a form of adjectives or adverbs that express the highest or a very high degree of quality of what is being described. They have two categories:

1) *Adjectival Superlatives:* Such a superlative is used to say what thing or person has the most of a particular quality within a group or of its kind. Ex: "*John is the tallest person.*"
2) *Adverbial Superlatives*: The superlative is used to say what thing or person does something to the greater degree within a group or of its kind. Ex: "*Jill did her homework most frequently*"

English grammar also has *coordinations* like "*John and Sue, both like sushi*" which sometimes express the relation of type *equality*. These sentences use coordinating conjunctions like *and, or,* etc.

The above linguistic classification of comparative sentences has two limitations.

1) **Non-comparatives with comparative words**: In linguistics, sentences containing comparative adjectives (*JJR*) and adverbs (*RBR*) (e.g., *more*, *less*, *better*, *longer* and words ending with *–er*), words like *same*, *similar*, *differ* and those used with equative *as* (e.g., *same as*, *as well as*) or superlative adjectives (*JJS*) and adverbs (*RBS*) (e.g., *most*, *least*, *best*, *tallest* and words ending with *–est*) are considered comparisons. However, in practice these *comparative indicators* may not be used for comparisons, e.g.,

   "*In the context of speed, faster means better*"

   "*John has to try his best to win this game*"

   Although these two sentences contain comparative words, they are not comparisons for practical purposes.

   There is also an issue of meaningless comparison, e.g., "*More men than James like scotch on the rocks*" It compares non-compatible entities.

2) **Limited coverage**: In practice, there are many comparative sentences which do not contain any of the above comparative words. Consider a few examples:

   "*In market capital, Intel is way ahead of Amd*."

   "*Nokia, Samsung, both cell phones perform badly on heat dissipation index*."

   "*The M7500 earned a World bench score of 85, whereas Asus A3V posted a mark of 89*"

   None of them contain any comparative words above.

3.2 Our Enhancements

To address the first limitation, we will use computational methods (e.g., machine learning methods) to distinguish comparatives and non-comparatives.

To address the second limitation, we added *user preferences*, and *implicit comparatives*.

1) *User preference*: Linguistic classification primarily deals with sentences which have POS tags *JJR*, *RBR*, *JJS*, and *RBS*, which usually express direct comparisons of two objects. However, the user may also express a comparison indirectly through preferences, e.g., "*I prefer Intel to Amd.*"

which is similar to "*Intel is better than Amd*".

2) *Implicit comparatives*: As we discussed earlier, in linguistics comparatives express ordering of objects. However, in many comparative cases, no explicit ordering is expressed (although it may be implied). For example, "*camera X has 2 MP, whereas camera Y has 5 MP*."

We therefore propose an enhanced categorization of comparatives, which is discussed below in Section 3.3.

3.3 Problem Statement

In this work, we study comparatives at the sentence level. Thus, we state the problem based on sentences.

Definition (comparative sentence): A *comparative sentence* is a sentence that expresses a relation based on similarities or differences of more than one object.

Definition (objects and their features): An *object* is an entity that can be a person, a product, an action, etc, under comparison in a comparative sentence. Each object has a set of features, which are used to compare objects.

A comparison can be between two or more objects, groups of objects, one object and the rest of the objects. It can also be between an object and its previous or future versions.

Types of comparatives: We group comparatives into four types. The first three of which are *gradable* comparatives and the fourth one is *non-gradable* comparative. The *gradable* types are defined based on the relationships of *greater or less than*, *equal to*, and *greater or less than all others*.

1) *Non-Equal Gradable*: Relations of the type *greater* or *less than* that express an ordering of some objects with regard to certain features. This type includes user preferences, and also those comparatives that do not use *JJR* and *RBR* words

2) *Equative*: Relations of the type *equal to* that state two objects as equal with respect to some features.

3) *Superlative*: Relations of the type *greater* or *less than all others* that rank one object over *all* others.

4) *Non-Gradable*: Sentences which compare features of two or more objects, but do not grade them. Sentences which imply:

    1. *Object A* is similar to or different from *Object B* with regard to some features.
    2. *Object A* has feature F1, *Object B* has feature F2 (F1 and F2 are usually substitutable).
    3. *Object A* has feature F, but *object B* does not have.

Incidentally, these definitions are also used as guidelines to annotate (or label) sentences for the evaluation of our technique.

Tasks: We identify two main tasks in dealing with comparisons:

- Identifying comparative sentences from a given text data set.
- Extracting comparative relations from sentences.

In this work, we focus on the first task, i.e. identifying comparative sentences from text documents. The second task is studied in [12].

Challenges: Two main challenges of this work are as follows:

1. Not all sentences with POS tags JJR, RBR, JJS and RBS are comparisons, e.g., "*In the context of speed, faster means better*."
2. Some sentences are comparisons but do not use any indicator word. For example, "*Coffee is expensive, but Tea is cheap*."

The data sets that we used consisted of disparate types, reviews, forum postings, news articles, which posed their own challenges. The main problem is badly formed sentences, e.g., short and incomplete sentences, violation of grammar rules, lack of punctuations, no proper casing of words, etc.

Subjective and objective comparisons: As mentioned earlier, a comparative sentence can be subjective or objective. A *subjective comparison* expresses an opinion. An *objective comparison* expresses a comparison that is objectively measurable. For example, "*I like car X more than car Y*" expressed a subjective comparison. "*John is taller than Tom*" and "*John is taller than Tom by 2 inches*" are both objective comparisons (assume they are true). In this work, we do not classify subjective and objective comparisons. We leave that to our future work.

4. PROPOSED TECHNIQUE

We now present the proposed technique to identify comparative sentences. The approach is a combination of *class sequential rule* (CSR) mining and machine learning. Sequential patterns in the rules are used as features. CSRs are found automatically using a class sequential rule mining system. A keyword strategy that takes advantage of the nature of our problem is also designed to filter out sentences that are unlikely to be comparative sentences. For classification, we experimented with two approaches:

1. Directly applying class sequential rules.
2. Using a machine learning algorithm to build a classifier based on the rules.

We will discuss both approaches shortly. Below, we first define class sequential rules, and then generate the data to be used for discovering such rules.

4.1 Class Sequential Rules with Multiple Minimum Supports

Sequential pattern mining (SPM) is an important data mining task [1, 2, 25]. Given a set of input sequences, the SPM task is to find all sequential patterns that satisfy a user-specified minimum support (or frequency) constraint. A sequential pattern is simply a sub-sequence that appears more frequently in the input sequences than the minimum support threshold. Many algorithms exist for mining such patterns in data mining [e.g., 1, 2, 25].

A class sequential rule (CSR) is a rule with a sequential pattern on the left and a class label on the right of the rule. Unlike classic sequential pattern mining, which is unsupervised, we mine sequential rules with fixed classes. The new method is thus supervised. We now define class sequential rules formally.

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items. A *sequence* is an ordered list of itemsets. An *itemset X* is a non-empty set of items. We denote a sequence s by $\langle a_1a_2\ldots a_r\rangle$, where a_i is an itemset, also called an *element* of s, and a_i is denoted by $\{x_1, x_2, \ldots, x_k\}$, where x_j is an item. An item can occur only once in an element of a sequence, but can occur multiple times in different elements. A sequence $s_1 = \langle a_1 a_2 ... a_r \rangle$ is a *subsequence* of another sequence $s_2 = \langle b_1 b_2 ... b_m \rangle$, if there exist integers $1 \leq j_1 < j_2 < ... < j_{r-1} \leq j_r$ such that $a_1 \subseteq b_{j_1}$, $a_2 \subseteq b_{j_2}$, ..., $a_r \subseteq b_{j_r}$. We also say that s_2 *contains* s_1.

Let us see an example. We have $I = \{1, 2, 3, 4, 5, 6, 7\}$. The sequence $\langle \{3\}\{4, 5\} \rangle$ is contained in $\langle \{6\}\{3, 7\}\{4, 5, 6\} \rangle$ because $\{3\} \subseteq \{3, 7\}$ and $\{4, 5\} \subseteq \{4, 5, 8\}$. However, $\langle \{3, 8\} \rangle$ is not contained in $\langle \{3\}\{8\} \rangle$ and vice versa.

The input sequence data D for mining is a set of pairs, i.e., $D = \{(s_1, y_1), (s_2, y_2), …, (s_n, y_n)\}$, where s_i is a sequence and $y_i \in Y$ is its class label. Y is the set of all classes. In our context, $Y = \{$*comparative*, *non-comparative*$\}$. A *class sequential rule* (CSR) is an implication of the form

$$X \rightarrow y, \text{ where } X \text{ is a sequence, and } y \in Y.$$

A data instance (s_i, y_i) in D is said to *cover* the CSR if X is a subsequence of s_i. A data instance (s_i, y_i) is said to *satisfy* a CSR if X is a subsequence of s_i and $y_i = y$. The *support* (sup) of the rule is the fraction of total instances in D that satisfies the rule. The *confidence* (conf) of the rule is the proportion of instances in D that covers the rule also satisfies the rule.

Table 1 gives an example sequence database with five sequences and two classes, c_1 and c_2. Using the minimum support of 20% and minimum confidence of 40%, one of the discovered CSRs is:

$$\langle \{1\}\{3\}\{7, 8\} \rangle \rightarrow c_1 \qquad [\text{support} = 2/5 \text{ and confidence} = 2/3]$$

Data sequences 1 and 2 satisfy the rule, and data sequences 1, 2 and 5 cover the rule.

Table 1. An example sequence database for mining CSRs

	Data Sequence	Class
1	$\langle \{1\}\{3\}\{5\}\{7, 8, 9\} \rangle$	c_1
2	$\langle \{1\}\{3\}\{6\}\{7, 8\} \rangle$	c_1
3	$\langle \{1, 6\}\{9\} \rangle$	c_2
4	$\langle \{3\}\{5, 6\} \rangle$	c_2
5	$\langle \{1\}\{3\}\{4\}\{7, 8\} \rangle$	c_2

Given a labeled sequence data set D, a minimum support (*minsup*) and a minimum confidence (*minconf*) threshold, CSR mining finds all class sequential rules in D. The mining algorithm is involved and beyond the scope of this paper (see [20] for details).

Multiple minimum supports: The above model uses a single minimum support to control the rules to be generated. This is, however, not sufficient in our case because some words that indicate comparison appear very frequently, while some others appear rarely. Existing sequential pattern discovery algorithms in data mining uses only a single minimum support to control the pattern generation process. This is inadequate because to identify patterns that involve infrequent words, we need to set the minimum support very low, which causes those frequent words to generate a huge number of spurious CSR rules that are harmful to classification (due to overfitting). We thus propose the multiple minimum supports model to deal with this problem. In this model each word is set a minimum support based on the frequency that it appears in the training data.

This model enables us to find those rare patterns without generating too many overfitting rules that harm classification. Note that multiple minimum support association rule mining was studied in [18]. However, its algorithm cannot be applied here since the algorithm needs to sort the words in each sentence in a particular order, which is not permitted in our case since that destroys the word sequence. To achieve the multiple minimum support effect, we use the following method, where τ is a parameter selected through experiments (we used 0.10). From lines 1 and 3, we can see that *minsup* changes according to the actual frequency of the items in the data. Thus, for frequent items the *minsup* will be high and for rare item the *minsup* will be low. The function CSR generates all the rules related to the items in W.

```
1. Compute the frequencies of all the items in the training data
2. for each group of items W with the same frequency do
3.     minsup = frequency(W) * τ;
4.     CSR(trainingData, W, minsup, minconf);
5. end_for
```

4.2 Constructing the Data Set for Mining

We now discuss how to construct a data set from text documents. Since in this project we work at the sentence level, intuitively each sentence should be treated as a sequence (almost). However, we cannot use the raw words in each sentence because the contents of some sentences may be very different, but their underlying language patterns can be the same. Using the raw words, such patterns may not be found. For example, the following two sentences compare completely different objects:

"*Intel is better than Amd*", and

"*Laptops are smaller than desktop PCs*"

Simply comparing the words, the system will not see any pattern apart from the fact that there is one common word "than". However, a human person can clearly see a pattern. If we replace each word with its POS tag, the pattern becomes apparent. Thus, POS tags capture content independent language patterns, which is useful to us.

4.2.1 The Keyword Strategy

Our study shows an interesting phenomenon about comparative sentences. That is, it is easy to find a small set of keywords that covers almost all comparative sentences, i.e., with a very high recall. However, the precision is not high. This allows us to design the following strategy for learning.

Keyword strategy: Since the recall is very high and the precision is low, we simply try to improve the precision. More precisely, we consider only those sentences that contain at least one keyword, and then generate class sequential rules to filter out those non-comparative sentences. This has an implication on our data generation. That is, sentences that do not contain any keywords are discarded.

Let us see what the keywords are. Apart from *-er* words, there are many other indicative words for comparisons, e.g., *beat*, *exceed*, *outperform,* etc. We have compiled a list of keywords. We first manually found a list of 30 words by going through a subset of comparative sentences. We then used WordNet [7] to find their synonyms. After manual pruning, a final list of 69 words is produced. Note that the keyword set also contains some phrases such as *number one*, and *up against*. Non-gradable comparative sentences do not necessarily use these keywords. So, we include 9 more words and phrases such as *but*, *whereas*, *on the other hand*, etc, which are sometimes used in non-gradable comparisons. The phrases such as *as far as*, where a word is sandwiched between two equatives *as* are also used but all such phrases are considered as the same keyword. The words with POS tags of *JJR*, *RBR*, *JJS* and *RBS* are also good indicators. However, we do not use each individual raw word as a keyword. Instead, we use their POS tags, i.e., *JJR*, *RBR*, *JJS* and *RBS*, as four keywords only. Thus, the set of keywords, *K*, is defined as:

$$\begin{aligned} K = \; & \{JJR, RBR, JJS, RBS\} \cup \\ & \{\text{words such as } \textit{favor}, \textit{prefer}, \textit{win}, \textit{beat}, \textit{but}, \text{etc}\} \cup \\ & \{\text{phrases such as } \textit{number one, up against}, \text{etc}\} \end{aligned}$$

All together, we have 83 keywords and key phrases. Although identifying these keywords was time consuming, it is only a one-time effort. We will make this list available as a community resource. It is possible that a more automatic method can be used to find these words through machine learning. However, that will need a huge number of manually labeled sentences. Manual labeling is time-consuming too. Thus, it may be more cost effective to manually identify such keywords instead, which is what we did. We do not claim that our set is complete. As our project progresses and other researchers work on the problem, we foresee that the list will be expanded and become more and more complete. We do believe that we have a good number of indicative words and phrases.

It is important to note again that not all sentences that contain these keywords are comparative sentences. In fact, a large number of them are not. As we will see in the experiment section, only 32% sentences that contain one or more of these keywords are genuine comparative sentences. However, these keywords are able to capture 94% comparative sentences. That is, 94% is the recall and 32% is the precision if we use only these keywords to identify comparative sentences.

4.2.2 Building the Sequence Database

We are ready to generate the data set as follows:

1. For each sentence that contains at least one keyword or key phrase, we use the words that are within the radius of 3 of each keyword in the sentence as a sequence in our data. Our experiments show that the radius of 3 was optimum. Radius of 4 or more gave many spurious patterns that overfit the data. Using too few words does not give sufficient information.
2. Each word is then replaced with its POS tag. We do not use the actual words. For each keyword, we combine the actual keyword and the POS tag to form a single item. The reason for this is that some keywords have multiple POS tags depending upon their uses. Their specific usages can be important in determining whether a sentence is a comparative sentence or not. For example, the keyword "*more*" can be a comparative adjective (*more/JJR*) or a comparative adverb (*more/RBR*) in a sentence.
3. A class is attached to each sequence according to whether the sentence is a comparative or non-comparative sentence.

For example, consider the comparative sentence "*this/DT camera/NN has/VBZ significantly/RB more/JJR noise/NN at/IN iso/NN 100/CD than/IN the/DT nikon/NN 4500/CD.*" It has the keyword *more*. The final sequence put in the database is:

$$\langle\{\text{NN}\}\{\text{VBZ}\}\{\text{RB}\}\{\text{moreJJR}\}\{\text{NN}\}\{\text{IN}\}\{\text{NN}\}\rangle \quad \textit{comparative}$$

Note that if a sentence contains multiple keywords, each keyword generates a sequence in the sequence database.

CSR rule generation: After the database is constructed we can generate class sequential rules, which meet the minimum confidence threshold (we use 60% in our experiments, which work very well). The minimum support for each item is controlled by τ, which we set to 0.10.

Manual rules: We also added some rules compiled manually. Such rules are more complex and hard to be generated by current pattern mining techniques. For instance, we found that conjugates such as *whereas/IN, but/CC, however/RB, while/IN, though/IN, although/IN, etc*., occur with a comparative keyword in a sentence is a good indicator for a comparison. We have 13 such rules.

4.3 Classification Learning

Recall a CSR basically expresses the probability that a sentence is a comparison if it contains the pattern *X*. Clearly, we can use these rules for classification directly. We tried the following:

For each sentence, we find all the rules that are satisfied by the sentence, and choose the rule with the highest confidence to classify the sentence. If this rule's class is "*comparative*" then the sentence is classified as a comparative sentence, and otherwise a non-comparative sentence. This is a reasonable strategy because the confidence is a predictive measure.

However, this method does not work well as we will see in the experiment results. We believe that the key reason is that a given sentence often satisfies several rules. These rules may have conflicting classes. Choosing only one may be quite dangerous. To deal with the conflicts, a principled method is needed to combine all the rules to arrive at a single classification. The naïve Bayesian classification model (NB) [20, 21] provides a natural solution as it is able to combine multiple probabilities to arrive at a single probability decision. Our experiment results show that the classifier built using this learning approach based on the class sequential rules performs much better. We will not describe the NB model here as it is quite standard in machine learning.

Prepare the data set for learning: The NB model cannot learn directly using the sequence database because it cannot consider word sequence. We create a new database using CSRs for NB learning. The feature set is:

$$\begin{aligned} \textit{Feature Set} \; = \; & \{X \mid X \text{ is the sequential pattern in CSR } X \rightarrow y\} \cup \\ & \{Z \mid Z \text{ is the pattern in a manual rule } Z \rightarrow y\} \end{aligned}$$

The classes are still "*comparative*" and "*non-comparative*". Each sentence forms a tuple in the data. If the sentence has a particular pattern in the feature set, the corresponding feature value is 1, and is 0 otherwise. Using the resulting data, it is straightforward to perform NB learning. We also tried a SVM learner (*LIBSVM*, http://www.csie.ntu.edu.tw/~cjlin/libsvm/). It did not perform as well in our application.

5. EXPERIMENTAL RESULTS

This section evaluates our approach and discusses the results. We first describe the data sets used in our experiments and then present the experimental results.

5.1 Data Sets and Labeling

We collected data from disparate resources to represent different types of text. Our data consist of

- Consumer reviews on such products as *digital cameras*, *DVD players*, *MP3 players* and *cellular phones*. This data set is first used in [11], which studies opinions in reviews. The reviews were downloaded from amazon.com.
- Forum discussions from different websites on such topics as *Intel vs AMD*, *Coke vs Pepsi*, and *Microsoft vs Google*.
- News articles on random topics such as *automobiles*, *ipods*, and *soccer vs football*.

The sentence distribution of the data sets is given in Table 2.

Table 2. Number of sentences in different data sets

Data sets	Comparative Sentences	Non-Comparative Sentences
Reviews	308	2857
Forums	257	760
News articles	340	1368
Total	905	4985

Labeling: The data sets were all manually labeled. Since the labeling is subjective, it was done by four human labelers. In order to make the labeling consistent among the labelers, we first defined different categories of comparative sentences as discussed in Section 3.3. The labelers were asked to strictly follow the definitions. For the conflicting cases, a discussion was initiated to convince one another one way or the other to reach an agreement.

5.2 Experimental Results

We now give the precision, recall and F-score results at each step of our technique. Several of them can be considered as baselines. We will also show the precision, recall and F-score values on individual data sets.

The overall results are given in Figure 1, which contains the precision, recall and F-score values of all the steps (different techniques). All the results except for the first two were obtained through 5-fold cross validation. We discuss the results below:

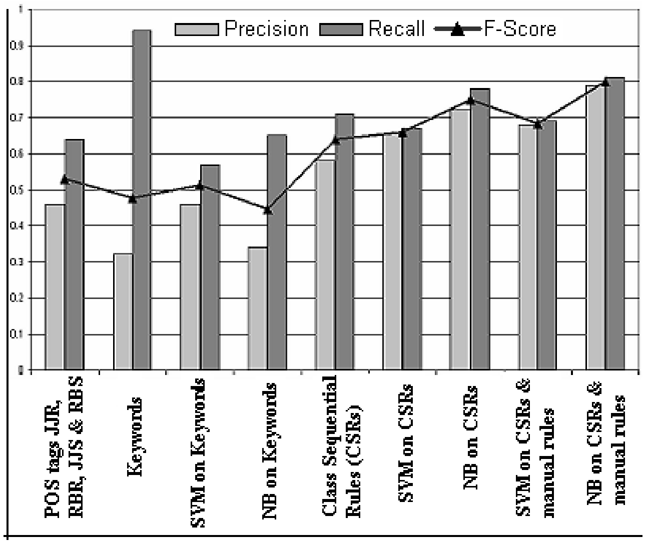

Figure 1. Precision, recall and F-score values of different approaches for the problem

1) *POS tags of JJS, RBR, JJS and RBS*: We used the Brill's Tagger [3]. If a sentence contains anyone of the above tags, it is classified as a comparative sentence. We obtained the recall of 64%. Clearly, many sentences were left out because they do not have these tags. The precision is less than 46%, which indicate that many sentences that contain above tags are not comparative sentences.

2) *Keywords*: Using only those important keywords, we obtained a recall of 94%, i.e., every sentence that contains one or more of the keywords as discussed in Section 4.2.1 is considered as a comparative sentence. This shows that these keywords are good indicators. However, the precision is very low, 32%. Thus, the F-score is poor.

3) *SVM and naïve Bayesian using keywords as features*: The F-score is notably improved after SVM learning is applied. We used the LIBSVM package, kernel = GAUSSIAN, gamma = 0.0623 and *C* = 97 gave the best F-score of 51%. However, naïve Bayesian's result is poorer.

4) *Class Sequential Rules (CSRs)*: If the multiple minimum support class sequential rules are used alone for classification, we achieve the precision of 58%, the recall of 71% and the F-score of 64%, which is a much better result than those of the above methods.

5) *SVM and NB using Class Sequential Rules (CSRs)*: The F-scores improve significantly after learning is applied, especially with naïve Bayesian. The F-score jumps to 75% with the naïve Bayesian.

6) *SVM and NB using both CSRs and manual rules*: Using all rules consisting of class sequential rules (CSRs) and manual rules, the F-scores improve further for both SVM and naïve Bayesian. However, the naïve Bayesian outperforms SVM with a precision of 79% and a recall of 81%. Thus, the manual rules helped increase the F-score by approximately 5%. These show that the manual complex rules are useful. However, they are hard to find by automatic algorithms. Our future work will study how such rules may be mined automatically. The results for manual rules alone are not included because the recall is very low (as they are compiled only to capture complex patterns).

The performance of the naïve Bayesian classifier on individual data sets is given in Table 3.

Table 3. Precision, recall and F-score values for different data sets from naïve Bayesian

Data sets	Precision	Recall	F-Score
Reviews	0.84	0.80	82%
Articles	0.75	0.80	77%
Forums	0.73	0.83	78%

The recalls are almost the same for all three data sets. The precision for *reviews* is higher than for *articles* and *forums*. The articles and forum postings had very long sentences which tend to satisfy a large number of patterns, even if they are not comparisons. This results in the lower precisions. The recalls of *reviews* and *forums* are affected in an opposite way, i.e., the classifier could not recognize comparative sentences that are very short as they satisfy very few or no patterns. The recall for articles is also affected because they contain a higher percentage of non-gradable comparisons which are harder to pick.

6. CONCLUSIONS AND FUTURE WORK

This paper proposed the study of identifying comparative sentences. Such sentences are useful in many applications, e.g., marketing intelligence, product benchmarking, and e-commerce. We first analyzed different types of comparative sentences from both the linguistic point of view and the practical usage point of view, and showed that existing linguistic studies have some limitations. We then made several enhancements. After that we proposed a novel rule mining and machine learning approach to identifying comparative sentences. Empirical evaluation using diverse text data sets showed its effectiveness. In our future work, we will prove both the precision and recall of the proposed technique, and also study how to automatically classify subjective and objective comparisons.

7. ACKNOWLEDGEMENTS

We thank Rajmonda Sulo and Makio Tamara for many useful discussions, annotation of data and helping us in literature survey.

8. REFERENCES

[1] Agrawal, R. Srikant, R. Mining sequential patterns. *ICDE'94*, 1994.

[2] Ayres, J., Flannick, J., Gehrke, J., Yiu, T. Sequential pattern mining using a bitmap representation. *KDD'02*

[3] Brill, E. A simple rule-based part of speech tagger. *ANL*, 1992.

[4] Das, S. and Chen, M., Yahoo! for Amazon: Extracting market sentiment from stock message boards. *APFA,* 2001.

[5] Dave, K., Lawrence, S., and Pennock, D. Mining the Peanut Gallery: Opinion extraction and semantic classification of product reviews. *WWW'03*, 2003.

[6] Doran, C., Egedi, D., Hockey, B. A., Srinivas, B., Zaidel, M. XTAG System-A wide coverage grammar for English**.** *COLING'94,* 1994.

[7] Fellbaum, C. WordNet: an electronic lexical database, MIT Press, 1998.

[8] Carenini, G. Ng, R., Zwart, E. Extracting knowledge from evaluative text. *ICKC'05*, 2005.

[9] Hatzivassiloglou, V., and Wiebe, J. Effects of adjective orientation and gradability on sentence subjectivity. *COLING'00*, 2000.

[10] Hearst, M., Direction-based text interpretation as an information access refinement. In P. Jacobs, editor, *Text-Based Intelligent Systems*. Lawrence Erlbaum Assoc., 1992.

[11] Hu, M., and Liu, B. Mining and summarizing customer reviews. *KDD'04*, 2004.

[12] Jindal, N., and Liu, B. Mining comparative sentences and relations. *AAAI'06*, 2006

[13] Joachims, T. Making large-scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), 1999.

[14] Kennedy, C. Comparatives, semantics of. *In Enclycopedia of Language and Linguistics*, Second Edition, Elsevier, 2005.

[15] Kobayashi, N., Iida, R., Inui, K. and Matsumoto, Y. Opinion mining on the Web by extracting subject-attribute-value relations. *AAAI-CAAW'06*, 2006.

[16] Ku, L-W., Liang, Y-T. and Chen, H-H. Opinion extraction, summarization and tracking in news and blog corpora. *AAAI-CAAW'06*, 2006.

[17] Lerner, J. and Pinkal M. Comparatives and nested quantification. *CLAUS-Report 21,* 1992.

[18] Liu, B., Hsu, W. and Ma, Y. Mining association rules with multiple minimum supports. *KDD'99*, 1999.

[19] Liu, B., Hu, M. and Cheng, J. Opinion observer: analyzing and comparing opinions on the Web. *WWW'*05, 2005.

[20] Liu, B. *Web Data Mining: Exploring hyperlinks, Contents, and Usage Data*. A forthcoming book. 2006/2007.

[21] Mitchell, T. *Machine learning*, McGraw-Hill, 1997.

[22] Moltmann, F., *Coordination and comparatives*. Ph.D. dissertation. MIT, Cambridge Ma., 1987.

[23] Nasukawa, T. and Yi, J. Sentiment analysis: Capturing favorability using natural language processing. *K-CA*, 2003.

[24] Pang, B., Lee, L., and Vaithyanathan, S., Thumbs up? Sentiment Classification Using Machine Learning Techniques. *EMNLP'02*, 2002.

[25] Pei, J. Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. Mining sequential patterns by Pattern-Growth: The PrefixSpan approach. *IEEE Trans. on Knowl. and Data Engineering*, 16(10), 2004.

[26] Popescu, A-M and Etzioni. O. Extracting product features and opinions from reviews. *EMNLP-05,* 2005.

[27] Riloff, E. and Wiebe, J. Learning extraction patterns for subjective expressions. *EMNLP'03*, 2003.

[28] Santorini, B., Part-of-Speech tagging guidelines for the Penn Treebank project. *Technical report MS-CIS-90-47*. U. Pennsylvania. Dep't. Of Computer Science, 1990.

[29] Tong, R. An operational system for detecting and tracking opinions in on-line discussions. *SIGIR 2001 Workshop on Operational Text Classification*, 2001.

[30] Turney, P. 2002. Thumbs up or Thumbs down? Semantic orientation applied to unsupervised classification of reviews. *ACL'02*, 2002.

[31] Vapnik. V. The nature of statistical learning theory, Springer, 1995.

[32] Wiebe, J., Bruce, R., and O'Hara, T. Development and use of a gold standard data set for subjectivity classifications. *ACL'99*, 1999.

[33] Wilson, T., Wiebe, J., and Hwa, R. Just how mad are you? Finding strong and weak opinion clauses. *AAAI'04*, 2004.

[34] Yu, H., and Hatzivassiloglou, V. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *EMNLP'03*, 2003.

[35] Zhai, C, Velivelli, A., and Yu, B. A cross-collection mixture model for comparative text mining, *KDD'04*, 2004.