# Extraction of Unexpected Sentences: A Sentiment Classification Assessed Approach

Dong (Haoyuan) Li
LGI2P, École des Mines d'Alès
Parc scientifique Georges Besse, 30035 Nîmes, France
Haoyuan.Li@ema.fr

Anne Laurent, Pascal Poncelet, and Mathieu Roche
LIRMM, Université Montpellier 2
161 rue Ada, 34392 Montpellier, France
{laurent, poncelet, mroche}@lirmm.fr

**Abstract**

Sentiment classification in text documents is an active data mining research topic in opinion retrieval and analysis. Different from previous studies concentrating on the development of effective classifiers, in this paper, we focus on the extraction and validation of unexpected sentences issued in sentiment classification. In this paper, we propose a general framework for determining unexpected sentences in the context of text classification. In the experiments, we present the extraction of unexpected sentences for sentiment classification within the proposed framework, and then evaluate the influence of unexpected sentences with cross-validation methods. The experimental results show the effectiveness and usefulness of our proposed approach.

**Key words:** Sentiment classification, text classification, unexpected sentences, extraction, validation.

# 1 Introduction

Sentiment classification received much attention in analyzing personal opinion orientations contained in user generated contents, such as customer reviews, online forums, discussion groups, blogs, etc., where the orientations are often classified into *positive* or *negative* polarities. Although the sentiment classification of personal opinions is determinative, the sentences expressing the sentiment opposite to the overall orientation expressed by the document can be interesting for many purposes.

For instance, a customer review that has been classified into positive opinions about a product may contain some sentences pointing out the weakness or faults of the product, or a review classified as negative may nevertheless recognizes the good points of the product. Therefore, in our previous work [18], we proposed a belief driven approach to extract *opposite sentiments* in classified free format text documents.

Indeed, sentiment classification can be regarded as a sub-category of *text classification* tasks. The task of text classification is generally performed by the *classifier* that describes how a document is classified (a systematic survey can be found in [31]). The great practical importance of text classification techniques has been addressed since the last 10 years, which covers the massive volume of user generated content available in the Web, electronic mail, customer reviews, medical records, digital publications, and so on.

On the other hand, many examples can be addressed for illustrating the sentences unexpected to document category as well as the opposite sentiments in the context of sentiment classification. For instance, in an online news group about politics events, discussions on politics are expected to be posted, however the contents on football can be considered as unexpected. One reason to study the unexpected sentences contained in text documents is that according to the principle of classifiers, unexpected sentences may decrease the accuracy of classification results. Further, another reason is that unexpected contents can be interesting because they are unexpected: as an interestingness measure for data mining, *unexpectedness* [32] is concerned by many literatures in the past years and has shown a special performance in a broad of real-world applications [8, 24, 34, 14, 25, 17, 18].

In this paper, we study the sentiment classification assessed extraction and validation of unexpected sentences. We propose a general framework for determining unexpected sentences in the context of text classification. In this framework, we use sequential pattern based *class descriptors* for generalizing the characteristics of a document with respect to its class, and *unexpected class patterns* are therefore generated from the *semantic oppositions* of the elements contained in class descriptors. An *unexpected sentence* can be stated in a text document by examining whether it contains any unexpected class patterns. The semantic oppositions of a class descriptor can be determined in various manners. For sentiment classification tasks, the semantic oppositions of sentiment can be directly determined by finding antonyms of adjectives and adverbs. Therefore, in the experiments, we present the extraction of unexpected sentences for sentiment classification within the proposed framework.

Moreover, the effectiveness of subjective approaches to discover unexpected patterns or rules are often judged with respect to domain expertise [24, 34, 25, 17, 18]. In [18], the discovered sentences containing opposite sentiment are

examined by human experts. In this paper, we also propose a cross-validation process for measuring the overall influence of unexpected sentences by using text classification methods. The experimental evaluation shows that the accuracy of classification are increased without unexpected sentences. Our experiments also show that in the results obtained from the same document sets with randomly-removed sentences, the accuracy are decreased. The comparison between the classification accuracy of the documents containing only randomly-selected sentences and containing only unexpected sentences shows that the latter is significantly lower.

The rest of this paper is organized as follows. The related work is introduced in Section 2. In Section 3, we present the extraction of unexpected sentences in text documents. Section 4 shows our experimental results on the extraction and validation of unexpected sentences. Finally, we conclude in Section 5 with future research directions.

## 2 Related Work

We study the unexpected sentences in the context of sentiment classification that classifies documents with respect to the overall sentiment expressed.

Sentiment classification is often used to determine sentiment orientation in user reviews [27, 37, 7, 11, 26, 18]. The extraction of sentiment orientations is closely connected with Natural Language Processing (NLP) problems, where the positive or negative connotation are annotated by the subjective terms at the document level [37, 7, 26]. In order to obtain precise results, many approaches also consider sentence level sentiment orientation, such as [7, 43, 11, 38, 39].

In recent literatures, many various methods have been proposed to improve the accuracy and efficiency of sentiment classification, where machine learning based text classification methods are often applied. For instance, Pang et al. [27] studied the sentiment classification problems with Naive Bayes, maximum entropy, and support vector machines; Turney [37] proposed an unsupervised learning algorithm for classifying reviews with sentiment orientations. The effectiveness of text classification techniques has been addressed in a large range of application domains including categorizing Web pages [42, 21, 33, 35], learning customer reviews [37, 7], and detecting sentiment polarities [27, 4].

Actually, sentiment classification are performed by considering the adjectives contained in sentences[9, 36]. In [18], we present the problem of finding opposite sentiments in customer reviews, where we construct a set of sentiment models from adjective based frequent structures of sentences. We use WordNet [6] for determining the antonyms of adjectives required for constructing the belief base, which has been used in many NLP and opinion mining approaches. For instance, in the proposal of [15], WordNet is also applied for detecting the semantic orientation of adjectives. In this paper, we extendedly propose a general model of document class descriptors, which considers the adjectives, adverbs, nouns, verbs and negation identifiers.

We focus on discovering unexpected sentences with respect to document class, where unexpectedness is a subjective measure of interestingness. In [23], McGarry systematically investigated interestingness measures for data mining, which are classified into two categories: the objective measures based on the statistical frequency or properties of discovered patterns, and the subjective

measures based on the domain knowledge or the class of users. Subjective measures are studied in [32], in particular the actionability and unexpectedness. The term actionability stands for reacting to the discovered patterns or sequences to users advantage. The term unexpectedness stands for the newly discovered patterns or sequences that are surprising to users.

Unexpectedness is determined in terms of beliefs, which can be defined with respect to the notion of semantics. In [24, 25], Padmanabhan propose a belief-driven approach to find unexpected association rules, where a belief is given from association rule, and the unexpectedness is stated by the semantic opposition between patterns. In [17], we proposed the discovery of unexpected sequences and rules with respect to the completeness occurrence, and semantics of sequences, where the belief system is constructed from sequence rules and semantic contradiction between sequences. In [18], we proposed the extraction of opposite sentiments, where beliefs are defined from the contextual models of sentiment with respect to antonyms of adjectives.

# 3 Discovering Unexpected Sentences

In this section, we formalize the free-format text documents with PoS tags within the framework of sequence data mining, then we propose sequential pattern based class descriptors, from which unexpected class patterns can be generated and applied for discovering unexpected sentences.

## 3.1 Part-of-Speech Tagged Data Model

We are considering free-format text documents, where each document consists of an ordered list of sentences, and each sentence consists of an ordered list of words.

In this paper, we treat each word contained in the text as a *lemma* associated with its PoS tag, including *noun* (*n.*), *verb* (*v.*), *adjective* (*adj.*), *adverb* (*adv.*), etc., denoted as $(lemma|pos)$. For example, the word "are" contained in the text is depicted by $(be|v.)$, where *be* is the lemma of "are" and *verb* is the PoS tag of "be". Without loss of generality, we use the wild-card $*$ and simplified PoS tag for denoting a generalized word. For instance, $(*|adj.)$ denotes an adjective; $(*|adv.)$ denotes an adverb, $(*|n.)$ denotes a noun, $(*|v.)$ denotes a verb, and so on. Further, the *negation identifiers* are denoted as $(*|neg.)$, including *not*, *'nt*, *no* and *never*. We use a generalization relation between two words having the same PoS tag, which is a partial relation $\preceq$ such that: let $w_1 = (lemma_1|pos)$ and $w_2 = (lemma_2|pos)$, we have that $w_1 \preceq w_2$ implies $lemma_1 = lemma_2$ or $lemma_2 = *$. For example, we have that $(be|v.) \preceq (*|v.)$ but $(be|verb) \npreceq (film|n.)$.

A *vocabulary*, denoted as $V = \{w_1, w_2, \ldots, w_n\}$, is a collection of a limited number of distinct words. A *phrase* is an ordered list of words, denoted as $s = w_1 w_2 \ldots w_k$. A phrase can also contain generalized words. For example, $(film|n.)(be|v.)(good|adj.)$ is a phrase; $(film|n.)(*|v.)(good|adj.)$ and $(*|n.)(be|v.)(*|adj.)$ are two phrases with generalized words. The *length* of a phrase $s$ is the number of words (including generalized words) contained in this phrase, denoted as $|s|$. One single word can be viewed as a phrase with length

1. An *empty phrase* is denoted as $\emptyset$, we have that $s = \emptyset \iff |s| = 0$. A phrase with the length $k$ is called a *k-phrase*.

Within the context of mining sequence patterns [2], a word is an *item* and a phrase is a *sequence*. Given two phrases $s = w_1 w_2 \dots w_m$ and $s' = w'_1 w'_2 \dots w'_n$, if there exist integers $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that $w_i \preceq w'_{j_i}$ for all $w_i$, then $s$ is a *sub-phrase* of $s'$, denoted as $s \sqsubseteq s'$. If we have that $s \sqsubseteq s'$, we say that $s$ *is contained in* $s'$, or $s'$ *supports* $s$. If a phrase $s$ is not contained in any other phrases, then we say that the phrase $s$ is *maximal*. For example, $(film|n.)(good|adj.)$ is contained in $(film|n.)(be|v.)(good|adj.)$ but not in $(be|v.)(good|adj.)(film|n.)$; $(film|n.)(good|adj.)$ is contained in $(*|n.)(*|adj.)$ but not in $(*|v.)(*|adj.)$. The *concatenation* of phrases is denoted as $s_1 s_2 s_3 \dots$; the *subtraction* of two phrases $s_1$ and $s_2$ is denoted $s_1 \setminus s_2$ if and only if $s_2 \sqsubseteq s_1$. For instance, let $s_1 = w_a w_b w_c w_b w_d$ and $s_2 = w_b w_d$, we have that $s_2 \sqsubseteq s_1$ and $s_1 \setminus s_2 = w_a w_c w_b$: the first occurrence of $s_2$ (first $w_b$ and first $w_d$) in $s_1$ is removed.

A *sentence* is a *grammatical complete* phrase, denoted as $s^{\#}$. A *document* is a set of sentences, denoted as $D$. We do not concentrate on the order in the context of sequence data mining though a document is logically an ordered list of sentences. Moreover, in the same context, a document can be generalized to be a set of phrases. In this paper, the determination of sentence is addressed by one of the following symbols "; . ? !" in the text. Given a document $D$, the *support* or *frequency* of a phrase $s$, denoted as $supp(s, D)$, is the total number of sentences $s^{\#} \in D$ that support $s$. Given a user specified threshold of support called *minimum support*, denoted as $supp_{min}$, a phrase is *frequent* if $supp(s, D) \geq supp_{min}$.

**Text 1** *The actors in this film are all also very good. This is a good film without big budget sets. Very good sound, picture, and seats.*

**Example 1** *Text 1 contains 3 sentences. If we consider only the nouns, verbs, and adjectives contained in the text, Text 1 corresponds to a document D with 3 phrases:*

$$s_1 = (actor|n.)(film|n.)(be|v.)(good|adj.),$$
$$s_2 = (be|v.)(good|adj.)(film|n.)(big|adj.)(budget|n.)(set|n.),$$
$$s_2 = (good|adj.)(sound|n.)(picture|n.)(seat|n.).$$

*Given minimum support threshold* $min\_supp = 0.5$*, we have maximal frequent phrases* $p_1 = (be|v.)(good|adj.)$ *and* $p_2 = (film|n.)$ *where* $\sigma(p_1, D) = 0.667$ *and* $\sigma(p_2, D) = 1$.

The PoS tagged data model is purposed for the ease of data mining tasks. It is not difficult to see that the computational process cannot handle the support of the word "actor" in the sentence "the actors in this film are all also very good" without proper preprocess of the model of text. On the other hand, importing PoS tags into the data model makes it possible to focus only on specified parts of text, such as for building text class descriptors by adjectives and nouns.

## 3.2 Class Descriptors

In [31], Sebastiani generalized the text classification problem as the task of assigning a Boolean value to each pair $\langle D_j, C_i\rangle \in \mathcal{D} \times \mathcal{C}$ where $\mathcal{D}$ is a domain of documents and $\mathcal{C} = \{C_1, C_2, \ldots, C_{|\mathcal{C}|}\}$ is a set of predefined classes. A value $True$ assigned to $\langle D_j, C_i\rangle$ indicates a decision to classify $D_j$ under $C_i$, while a value of $False$ indicates a decision not to classify $D_j$ under $C_i$. A *target function* $\Phi : \mathcal{D} \times \mathcal{C} \to \{True, False\}$ is called the *classifier*. In practical, a *classification status value* (or *categorization status value*) function $\Omega_i : \mathcal{D} \to [0, 1]$ is considered in the classifier for class $C_i \in \mathcal{C}$. A *threshold* $\tau_i$ is therefore defined such that for a document $D_j$, $\Omega_i(D_j) \geq \tau_i$ is interpreted as $True$ while $\Omega_i(D_j) < \tau_i$ is interpreted as $False$. Most of existing text classifiers can be generalized to this model.

Given a document $D$ and a sentence $s^{\#} \notin D$ such that for a class $C_i$ we have $\Omega_i(D \cup s^{\#}) > \Omega_i(D)$, then there exists a set $S$ of phrases such that for each phrase $s \in S$ we have $s \sqsubseteq s^{\#}$ and $\Omega_i(D \cup s) > \Omega_i(D)$. We say that such a phrase $s$ *supports* the class $C_i$, denoted as $s \models C_i$, and this phrase $s$ is called a *key phrase* of $C_i$. Further, given a key phrase $s$ of a class $C_i$, there exists a set $W$ of words such that for each word $w \in W$ we have $w \subseteq s$ and $\Omega_i(D \cup w) > \Omega_i(D)$. We say that such a word $w$ *supports* the class $C_i$, denoted as $w \models C_i$, and this word $w$ is called a *key word* of $C_i$. In additional, we denote $s \not\models C_i$ (respectively for $w \not\models C_i$) that the phrase $s$ *is not* a key phrase of the class $C_i$, in this meaning, $s \not\models C_i$ does not imply but include the case $\Omega(D \cup s) < \Omega(D)$.

With a threshold $\tau_i$ for a class $C_i$ and a document $D$, let $D \models C_i$ denote that $\Omega_i(D) \geq \tau_i$ is interpreted as $True$ for the classification task, then the following theorem is immediate.

**Theorem 1** *Given a class $C_i$ and a document $D$, if $D \models C_i$, then there exists a subset $D' \subseteq D$ such that for each sentence $s^{\#} \in D'$ we have $s^{\#} \models C_i$, and for each sentence $s^{\#} \in (D \setminus D')$ we have $s^{\#} \not\models C_i$.*

Notice that for Theorem 1, the set $(D \setminus D')$ can be empty. In this case, each sentence $s^{\#} \in D$ supports the class $C_i$. According to the definitions of sentence and phrase in Section 2.1, we have the following lemma.

**Lemma 1** *Given a class $C_i$ and a document $D \models C_i$, the document $D$ contains a set $S$ of maximal phrases such that if $s \in S$ then $s \models C_i$.*

Considering a document domain $\mathcal{D}$ and a set $\Pi = \{D_1, D_2, \ldots, D_{|\Pi|}\} \in \mathcal{D}$ of documents pre-classified under a class $C_i$, that is, for each $D_j \in \Pi$ we have $D_j \models C_i$, let $\Gamma = \{s^{\#} \in D \mid D \in \Pi\}$ be the sentences contained in all documents and $S_i^+$ be the set of all maximal key phrases contained in $\Gamma$. For any two phrases $s_m, s_n \in S_i^+$ we have $s_m \not\sqsubseteq s_n$, $s_m, s_n \subseteq \Gamma$ and $s_m, s_n \models C_i$. The set $S_i^+$ is called the *predictive phrase set* of the class $C_i$.

**Definition 1** *Let $S_i^+$ be the predictive phrase set of a given document class $C_i$, the class descriptor of the class $C_i$ is a set $P_i^+$ of phrases such that: (1) each phrase $s \in P_i^+$ consists only of words with PoS tag in $\{adj., adv., n., v., neg.\}$; (2) for each phrase $s \in P_i^+$, there exists a phrase $s' \in S_i^+$ such that $s \sqsubseteq s'$; (3) for any two phrases $s_m, s_n \in P_i^+$, we have $s_m \not\sqsubseteq s_n$. Each phrase $s \in P_i^+$ is a class pattern.*

However, given a large set $\Pi$ of pre-classified documents under the class $C_i$, it is practically difficult to construct the predictive phrase set $S_i^+$ containing all predictive phrases in each document. On the other hand, association rules [1] and sequential patterns [2] have been used for building text classifiers [20, 19, 3, 13], where word frequency is a key factor for computing classification status value. In this paper, we consider the frequent phrases contained in the pre-classified document set as an approximation of the predictive phrase set, so that the class descriptor can further be approximately built from the discovered frequent phrases by filtering the adjectives, adverbs, nouns, verbs, and negation identifiers.

**Definition 2** *Let $\Pi$ be a set of text document under the class $C_i$, an approximate class descriptor of the document set $\Pi$ for the class $C_i$, denoted as $\Delta_i(\Pi)$, is the set of maximal frequent phrases consisting of adjectives, adverbs, nouns, verbs, and negation identifiers in the total text $\Gamma$ of the document set $\Pi$, with respect to a user defined minimum support threshold.*

In the rest of the paper, unless explicitly noticed, we consider the *approximate class descriptor* as the *class descriptor*.

A class descriptor consists of a set of maximal frequent phrases where each phrase is a class pattern, which can be modeled by its structure. A class pattern $p = w_1 w_2 \dots w_n$ is an ordered list of words, which can also be denoted as $p = (lemma_1|pos_1)(lemma_2|pos_2)\dots(lemma_n|pos_n)$. The structure $pos_1$-$pos_2$-$\dots$-$pos_n$ is called a *class pattern model*. If a class pattern consists of $k$ words, then we say that it is a $k$-phrase class pattern, corresponding to a $k$-phrase class pattern model. For instance, the 2-phrase class pattern $(famous|adj.)(actor|n.)$ corresponds to the class pattern model "ADJ.-N." (we present the PoS tags as upper case in a class pattern model).

**Text 2** *The other actors deliver good performances as well.*

**Example 2** *Assume that the sentence listed in Text 2 is contained in one of a large set $\Pi$ of text documents, which can be represented as*

$$s = (other|adj.)(actor|n.)(deliver|v.)(good|adj.)(performance|n.)(well|adv.),$$

*where $p_1 = (actor|n.)(good|adj.)$ and $p_2 = (good|adj.)(performance|n.)$ are two 2-phrases, and $p_3 = (actor|n.)(deliver|v.)(good|adj.)$ is a 3-phrase contained in $s$. Let $\Gamma$ be the total text of all documents in $\Pi$. Given a user specified minimum support threshold $min\_supp$, if we have $\sigma(p_1, \Gamma) \geq min\_supp$, $\sigma(p_2, \Gamma) \geq min\_supp$, and $\sigma(p_3, \Gamma) \geq min\_supp$, then $p_1$, $p_2$, and $p_3$ are 3 class patterns of the class $C_i$, respectively corresponding to class pattern models "N.-ADJ.", "ADJ.-N.", and "N.-V.-ADJ.".*

## 3.3 Unexpected Sentences

Given a class pattern $p$ of a text document set $\Pi$ under a class $C_i$, we consider the pattern $p$ as a *belief* on the class $C_i$. Hence, an *unexpected class pattern* is a phrase that semantically contradicts the class pattern $p$.

We first propose the notion of *$\phi$-opposition pattern* of class patterns. For facilitating the following descriptions, let us consider the *semantic opposition*

*relation* $w_1 = \neg w_2$ between two words, which denotes that the word $w_1$ semantically contradicts the word $w_2$. We have $w_1 = \neg w_2 \iff w_2 = \neg w_1$. The semantic opposition between words can be determined by finding the antonyms or computing the semantic relatedness of concepts. Currently, the computation of semantic relatedness between concepts have been addressed by various methods [5, 28, 10, 44].

**Definition 3** *Let* $p = w_1 w_2 \dots w_k$ *and* $p' = w'_1 w'_2 \dots w'_k$ *be two* $k$*-phrase class pattern. If* $p'$ *has a sub-phrase* $\eta = w_1^\eta w_2^\eta \dots w_\phi^\eta$ *and* $p$ *has a sub-phrase* $\varphi = w_1^\varphi w_2^\varphi \dots w_\phi^\varphi$*, where* $\phi \leq k$*, such that* $p' \setminus \eta = p \setminus \varphi$ *and for any* $1 \leq i \leq \phi$ *we have* $w_i^\eta = \neg w_i^\varphi$*, then the phrase* $p'$ *is a* $\phi$*-opposition pattern of* $p$*.*

Given a class pattern $p$, there exist various $\phi$-opposition patterns of $p$. For example, by detecting the antonyms of words, for a 2-phrase class pattern $(be|v.)(good|adj.)$, $(be|v.)(bad|adj.)$ is one of its 1-opposition pattern since $(good|adj.) = \neg(bad|adj.)$; for a 3-phrase class pattern $(be|v.)(good|adj.)(man|n.)$, according to $(good|adj.) = \neg(bad|adj.)$ and $(man|n.) = \neg(woman|n.)$, two 1-opposition patterns and one 2-opposition pattern can be generated.

Notice that the negation is not token into account with the notion of $\phi$-opposition pattern, however it is considered as a general word. For example, $(*|neg.)(bad|adj.)$ is generated as a 1-opposition pattern of the class pattern $(*|neg.)(good|adj.)$.

To take into consideration the negation of sentences, the notion of *$\phi$-negation pattern* is proposed as follows.

**Definition 4** *Let* $p = w_1 w_2 \dots w_k$ *be a* $k$*-phrase class pattern and* $p' = w'_1 w'_2 \dots w'_{k'}$ *be a* $k'$*-phrase class pattern where* $p \sqsubseteq p'$ *and* $k' = k + \phi$ *(*$\phi > 0$*). If* $w \in (p' \setminus p)$ *implies* $w = (*|neg.)$*, then the phrase* $p'$ *is a* $\phi$*-negation pattern of* $p$*.*

Not difficult to see, the generation of $\phi$-negation patterns depends on the value of $\phi$. For example, from the class pattern $(be|v.)(good|adj.)$, a 2-negation pattern $(*|neg.)(be|v.)(*|neg.)(good|adj.)$ can be generated.

Unexpected class patterns can be therefore generated from $\phi$-opposition and $\phi$-negation patterns of a class pattern. In this paper, we focus on 1-opposition and 1-negation patterns for generating unexpected class patterns.

Given a class descriptor $P_i^+$ of a text document set $\Pi$ under a class $C_i$, let $S_i^-$ be the ensemble of all $\phi$-opposition and $\phi$-negation patterns of each class pattern $p \in P_i^+$. The set $P_i^- = S_i^- \setminus P_i^+$ is called an *unexpected class descriptor* of the class $C_i$. Each phrase contained in $P_i^-$ is an *unexpected class pattern*. If a sentence contains an unexpected class pattern, then this sentence is an *unexpected sentence*.

The extraction of unexpected sentences can be performed with respect to the framework of (1) extracting class descriptors from pre-classified documents; (2) building unexpected class descriptors from $\phi$-opposition patterns and $\phi$-negation patterns of each class descriptor; (3) extracting unexpected sentences that contain unexpected class descriptors.

Not difficult to see, this framework can be performed to extract unexpected sentences with respect to general text classification problems if the unexpected class descriptors can be built.

To evaluate the unexpected sentences extracted from predefined classes of documents, we propose a four-step validation process:

1. The test on the classification of original documents, which shows the accuracy of each class of documents, denoted as $\alpha(D)$;

2. The test on the classification of the documents with randomly-removed $n$ sentences ($n$ is the average number of unexpected sentences per document) in each document, which shows the accuracy of disturbed documents, denoted as $\alpha(D \setminus R)$;

3. The test on the classification of the documents without unexpected sentences, which shows the accuracy of cleaned documents, denoted as $\alpha(D \setminus U)$;

4. The test on the classification of the documents only consists in unexpected sentences, which shows the accuracy of unexpectedness, denoted as $\alpha(U)$.

With comparing to the accuracy of original documents $\alpha(D)$, let the change of accuracy of the documents with randomly-removed sentences be $\delta_R = \alpha(D \setminus R) - \alpha(D)$ and let the change of accuracy of the documents without unexpected sentences be $\delta_U = \alpha(D \setminus U) - \alpha(D)$. According to the principe of text classifiers, we have the following property if the removed unexpected sentences are really unexpected to the document class.

**Property 1** *(1) $\delta_U > 0$; (2) $\delta_U \geq \delta_R$; (3) $\delta_R \leq 0$ is expected.*

Therefore, if the results of the cross-validation of document classification shows that the changes of accuracies correspond to the hypothesis on discovered unexpected sentences as proposed in 1, the we can say that the exception phrases contained in discovered unexpected sentences are valid, because the elimination of such sentences increases the accuracy of the classification task.

# 4 Experimental Evaluation

In this section, we present our experimental evaluation on the unexpected sentences in free format text documents within the context of sentiment classification, where the unexpected class descriptors are built from antonyms of word (determined by WordNet, including adjectives and adverbs) contained in class descriptors.

The data set concerned in our experiments is the movie review data from [26], which consists of pre-classified 1,000 positive-sentiment and 1,000 negative-sentiment text reviews. Thus, we consider "positive" and "negative" as two document classes in our experiments, and the goal is to discover unexpected sentences against the two classes and to validate discovered unexpected sentences.

## 4.1 Discovery of Unexpected Sentences

All documents are initially tagged by the TreeTagger [12] toolkit introduced in [30] to identify the PoS tag of each word. In order to reduce the redundancy in sequence-represented documents, we only consider the words that constitute the class descriptors including the adjectives, adverbs, verbs, nouns, and the negation identifiers. All words associated with concerned tags are converted to

PoS tagged sentences with respect to the order appeared in the documents, and all other words are ignored.

| Class | Documents | Sentences | Distinct Words | Average Length |
|---|---|---|---|---|
| Positive | 1,000 | 37,833 | 28,777 | 23.8956 |
| Negative | 1,000 | 36,186 | 27,224 | 22.2015 |

Table 1: Total number of sentences and distinct words, with average sentence length.

The total corpus contained in the data set consists of 1,492,681 words corresponding to 7.6 Megabytes. Table 1 lists each class of 1,000 documents of the movie review data set in sequence format. A dictionary totally containing 39,655 entries of item:word mapping is built for converting the sequences back into text for next steps.

The discovery of class descriptors is addressed as a training process with the same corpus. For each class, positive or negative in our experiments, all 1,000 sequence-represented documents are combined into one large sequence database, and then we perform *closed sequential pattern* mining algorithm CloSpan [40] to find class patterns describing the document class. Figure 1 shows the number of the discovered sequential patterns with different sequence length. According to the figure, the numbers of 4-length and 5-length sequential patterns strongly deceases when the minimum support value increases, for instance, with $min\_supp = 0.05\%$, the numbers of 2-, 3-, 4-, and 5-length sequential patterns of the class "positive" are respectively 7013, 3677, 705, and 46. Therefore, in order to obtain signifiant results, we find the class patterns limited to 2- and 3-length sequential patterns for next steps of our experiments.

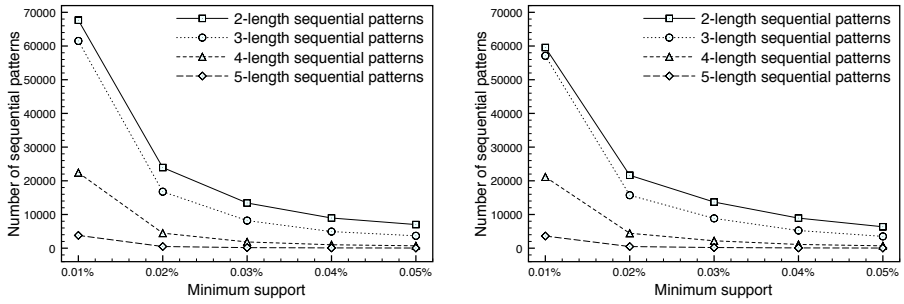
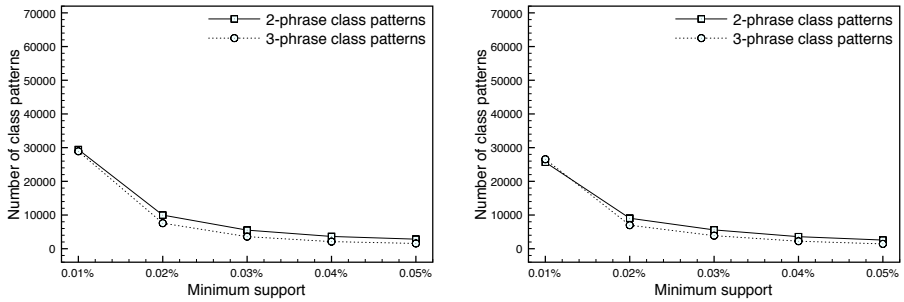



Figure 1: Number of discovered sequential patterns with different sequence length of: (a) the class "positive"; (b) the class "negative".

As described in Section 3.2, we extract the sequential patterns consisting of the adjectives, adverbs, nouns, verbs, and negation identifiers as the class descriptor. Figure 2 shows the total numbers of 2-phrase and 3-phrase class patterns that contain at least and at most one adjective or/and adverb, since the adjectives and adverbs are essential in sentiment classification.

The appearance of discovered 2-phrase class pattern models are listed in Table 2, ordered by the alphabet of models and $(*|neg.)$ with respect to different




(a) (b)

Figure 2: Number of 2-phrase and 3-phrase class patterns of: (a) the class "positive"; (b) the class "negative".

minimum support values. In order to save paper size, we only list the models corresponding to the $min\_conf$ values 0.01%, 0.03%, and 0.05%. For discovered 3-phrase class pattern models, the top-10 most frequent ones corresponding to $min\_conf = 0.01\%$ are listed in Table 3.

| Class Pattern Models | positive 0.01% | negative 0.01% | positive 0.03% | negative 0.03% | positive 0.05% | negative 0.05% |
|---|---|---|---|---|---|---|
| ADJ.-ADV. | 1089 | 892 | 134 | 134 | 34 | 32 |
| ADJ.-N. | 4049 | 3109 | 566 | 517 | 257 | 206 |
| ADJ.-V. | 2813 | 2474 | 581 | 558 | 321 | 276 |
| ADV.-ADJ. | 1654 | 1314 | 219 | 221 | 83 | 76 |
| ADV.-N. | 3348 | 3014 | 452 | 469 | 209 | 169 |
| ADV.-V. | 3084 | 2954 | 728 | 781 | 394 | 390 |
| N.-ADJ. | 2571 | 2045 | 292 | 286 | 127 | 100 |
| N.-ADV. | 2929 | 2729 | 438 | 478 | 194 | 189 |
| V.-ADJ. | 3841 | 3367 | 940 | 901 | 507 | 448 |
| V.-ADV. | 3157 | 2940 | 846 | 931 | 498 | 492 |
| NEG-ADJ. | 329 | 314 | 103 | 90 | 60 | 49 |
| ADJ.-NEG | 254 | 232 | 70 | 64 | 38 | 34 |
| NEG-ADV. | 166 | 147 | 79 | 83 | 66 | 62 |
| ADV.-NEG | 147 | 138 | 71 | 71 | 51 | 52 |

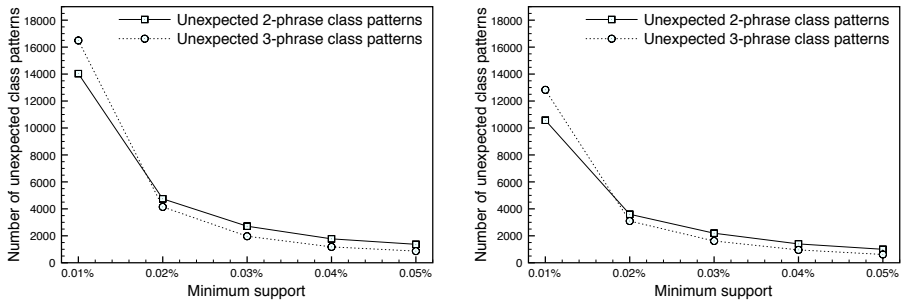
Table 2: 2-phrase class pattern models.

The unexpected class patterns are generated from the semantic oppositions of class patterns. In our experiments, the lexical database WordNet [6] is used for determining the antonyms of adjectives and adverbs for constructing semantic oppositions. For a class pattern, if there exist an adjective and an adverb together, then only the antonyms of the adjective will be considered; if the adjective and adverb have no antonym, then this class pattern will be ignored; if there exist more than one antonym, than more than one unexpected class pattern will be generated from all antonyms. The total numbers of unexpected 2-phrase and 3-phrase class patterns are shown in Figure 3.

The total numbers of unexpected sentences determined from unexpected 2-phrase and 3-phrase class patterns are shown in Figure 4, and the total numbers

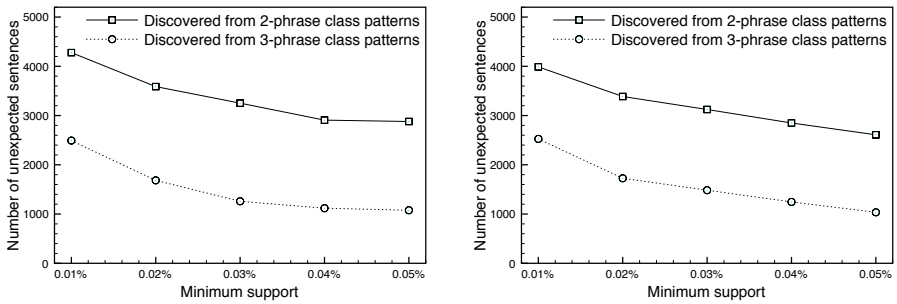
| Number | Models for class "positive" | Number | Models for class "negative" |
|---|---|---|---|
| 2289 | V.-V.-ADV. | 2343 | V.-V.-ADV. |
| 2121 | V.-ADV.-V. | 2106 | V.-ADV.-V. |
| 1801 | V.-V.-ADJ. | 1689 | V.-V.-ADJ. |
| 1691 | V.-ADJ.-N. | 1616 | ADV.-V.-V. |
| 1607 | ADV.-V.-V. | 1433 | V.-ADJ.-N. |
| 1546 | V.-ADJ.-V. | 1362 | V.-ADJ.-V. |
| 1340 | V.-ADV.-N. | 1212 | N.-V.-ADV. |
| 1276 | N.-V.-ADV. | 1159 | V.-ADV.-N. |
| 1045 | ADJ.-V.-V. | 969 | ADJ.-V.-V. |
| 946 | N.-V.-ADJ. | 861 | V.-N.-ADV. |

Table 3: 10 most frequent 3-phrase class pattern models.



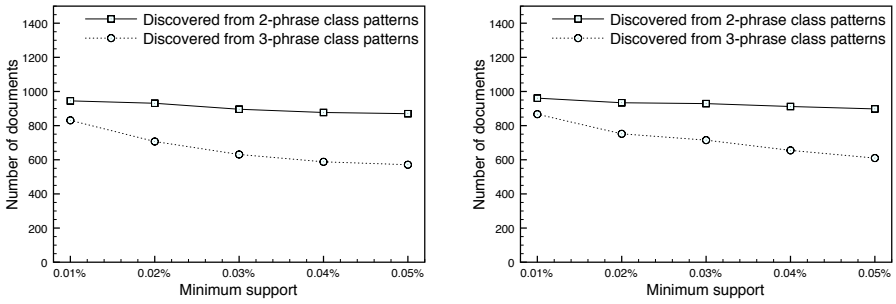

(a) (b)

Figure 3: Number of 2-phrase and 3-phrase unexpected class patterns of: (a) the class "positive"; (b) the class "negative".




(a) (b)

Figure 4: Number of unexpected sentences discovered from 2-phrase and 3-phrase unexpected class patterns of: (a) the class "positive"; (b) the class "negative".




(a) (b)

Figure 5: Number of documents that contain unexpected sentences discovered from 2-phrase and 3-phrase unexpected class patterns of: (a) the class "positive"; (b) the class "negative".

of documents that contain unexpected sentences are shown in Figure 5.

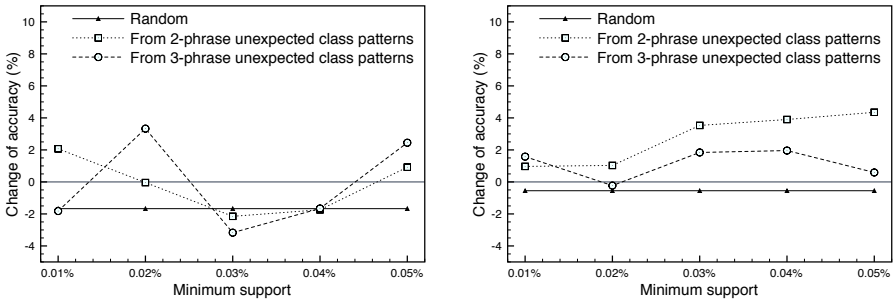
## 4.2 Validation of Unexpected Sentences

The goal of the evaluation is to use the text classification method to validate the unexpectedness stated in the discovered unexpected sentences with respect to the document class. The unexpectedness is examined by the Bow toolkit [22] with comparing the average accuracy of text classification tasks with and without unexpected sentences.

Three methods, *k-Nearest Neighbor* ($k$-NN), *Naive Bayes*, and *TFIDF* are selected for testing our approach by using classification tasks. The $k$-NN method [41] based classifiers are example-based that for deciding whether a document $D \models C_i$ for a class $C_i$, it examines whether the $k$ training documents most similar to $D$ also are in $C_i$. The Naive Bayes based classifiers (see [16]) compute the probability that a document $D$ belongs to a class $C_i$ by an application of Bayes' theorem, which accounts for most of the probabilistic approaches in the text classification. Nevertheless, the TFIDF (term frequency-inverse document frequency) [29] based classifiers compute the term frequency for deciding whether a document $D$ belongs a class $C_i$, however an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the collection and increases the weight of terms that occur rarely. Briefly, in order to learn a model, a prototype vector based on the TFIDF weight of terms is computed for each class, and then the cosine value of a new document between each prototype vector is calculated to assign the relevant class.

In our experiments, two groups of tests are performed, without and with pruning top-N words selected by highest average mutual information with the class variable. The purpose of this pruning is to reduce the size of feature set in order to emphasize the effects of removing unexpected sentences or randomly seleted sentences. Each test is performed with 20 trials of a randomized test-train split 40%-60%, and we take into account the final average values of accuracy. All tests are based on the unexpected sentences extracted with 2-phrase and 3-phrase unexpected class patterns obtained by different *min_supp* values from 0.01% to 0.05%.

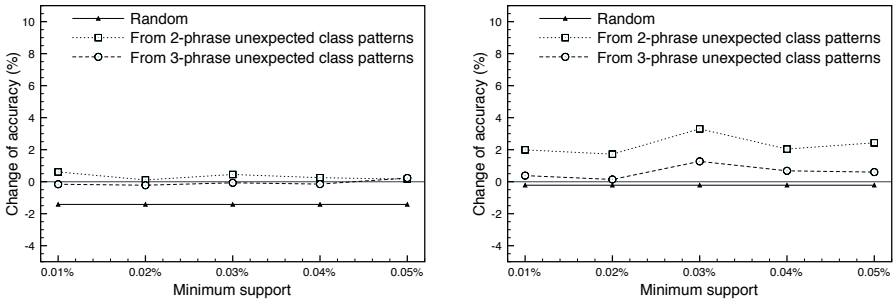
The evaluation results on the change of accuracy are shown in Figure 6, 7, and 8. The results are compared with removing the same number of randomly selected sentences from the documents. In each figure, the average accuracy of the original documents $\alpha(D)$ is considered as the base line "0", and the change of accuracy $\delta_R$ of the documents with randomly-removed sentences is considered as a reference line.




(a) (b)

Figure 6: Change of average accuracy before and after eliminating unexpected sentences by using $k$-NN method: (a) without pruning the most frequent common words; (b) with top-10 the most frequent common words pruned.

In the test results on the $k$-NN classifier shown in Figure 6(a), the change of accuracy is variant with respect to the *min_supp* value for extracting class patterns, however the results shown in Figure 6(b) well confirms Property 1. The behavior shown in Figure 6(a) also shows that although selecting frequent terms improves the accuracy of classification tasks, the frequent words common to all classes decrease the confidence of the accuracy of classification.

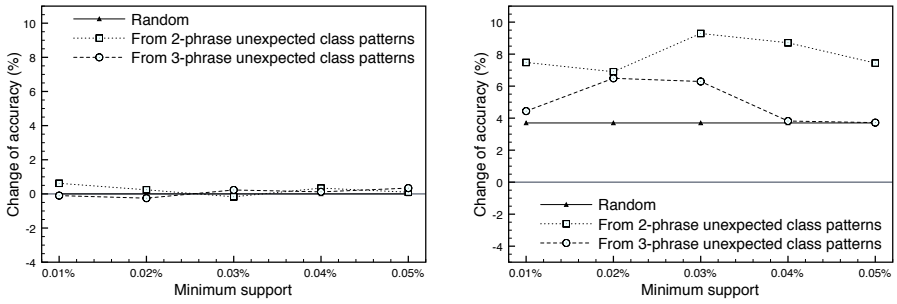



(c) (d)

Figure 7: Change of average accuracy before and after eliminating unexpected sentences by using Naive Bayes method: (a) without pruning the most frequent common words; (b) with top-10 the most frequent common words pruned.

Because Naive Bayes classifiers are probability based, Figure 7(a) is reasonable: the unexpected class patterns contained in all eliminated unexpected sentences weakly affect the probability whether a document belongs to a class since the eliminated terms are not frequent, but randomly selected sentences contains terms important to classify the documents. The prune of the most

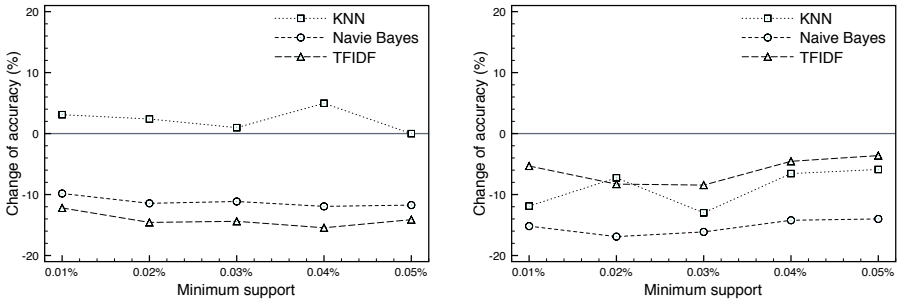
frequent common words enlarges the effects of unexpected sentences, thus the results shown in Figure 7(b) perfectly confirms Property 1.




(a) (b)

Figure 8: Change of average accuracy before and after eliminating unexpected sentences by using TFIDF method: (a) without pruning the most frequent common words; (b) with top-10 the most frequent common words pruned.

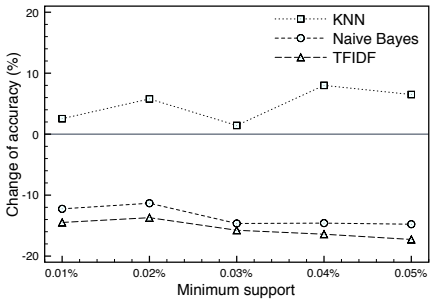
According to the principle of TFIDF weight, Figure 8(a) shows that the effect of comment frequent words in classification tasks is important, so that the elimination of limited number of sentences does not change the overall accuracy. Different from Naive Bayes classifiers, Figure 8(b) well confirms Property 1.(1) and Property 1.(2), however Property 1.(3) is not satisfied because the elimination of random selected sentences increases the overall accuracy of the classification.



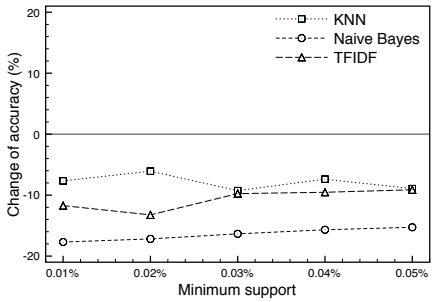

(a) (b)

Figure 9: Changes of average accuracy between original documents and the documents consisting of the unexpected sentences discovered from 2-phrase unexpected class patterns: (a) without pruning the most frequent common words; (b) with top-10 the most frequent common words pruned.

We also test the accuracy of the classification tasks on the documents consisting of only unexpected sentences, to study the characteristics of unexpected sentences, as shown in Figure 9 and Figure 10. Not difficult to see, the unexpected sentences are difficult to be classified with comparing to original documents. As discussed in previous analysis, the effect of the most frequent common words in $k$-NN based classifiers is strong.




(c) Without pruning words.




(d) Top-10 frequent words pruned.

Figure 10: Changes of average accuracy between original documents and the documents consisting of the unexpected sentences discovered from 3-phrase unexpected class patterns: (a) without pruning the most frequent common words; (b) with top-10 the most frequent common words pruned.

# 5 Conclusion

In this paper, we study the effects of unexpected sentences in sentiment classification. We first formalized text documents with PoS tags, and then proposed the notion of class descriptors and class patterns, from which we further proposed the notion of unexpected class patterns. A phrase containing an unexpected class pattern is therefore an unexpected sentence. In consequence, we evaluated discovered unexpected sentences by text classification, including *k-nearest neighbor*, *naive Bayes*, and *TFIDF* methods. The experimental results show that the discovery of unexpected sentences is effective and the accuracy of classification can be improved by eliminating unexpected sentences in text documents.

The approach proposed in this paper considers 1-opposition and 1-negation unexpected class patterns, which limits the performance of discovering unexpected sentences, although the effectiveness has been already shown. In our future research, we will focus on the construction of complex unexpected class patterns, such as 2-opposition and 2-negation patterns. Further, our approach is theoretically common for discovering unexpected sentences with respect to the general text classification problems, however, the generation of $\phi$-opposition unexpected patterns are currently limited in determining the antonyms of words, which is suitable for adjective and adverb based document classes, for example the positive and negative polarities in sentiment classification. In order to practically porting our approach to more general cases, for example topic-based document classes, we are interested in adopting semantic hierarchies for generating $\phi$-opposition unexpected patterns by determining the relatedness between concepts.

# References

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pages 3–14, 1995.

[3] M.-L. Antonie and O. R. Zaiane. Text document categorization by term association. In *ICDM*, pages 19–26, 2002.

[4] F. Benamara, C. Cesarano, and D. Reforgiato. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *ICWSM*, 2007.

[5] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[6] Cognitive Science Laboratory, Princeton University. WordNet: A lexical database for the english language. http://wordnet.princeton.edu/.

[7] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, 2003.

[8] G. Dong and J. Li. Interestingness of discovered association rules in terms of neighborhood-based unexpectedness. In *PAKDD*, pages 72–86, 1998.

[9] A. Esuli and F. Sebastiani. PageRanking WordNet synsets: An application to opinion mining. In *ACL*, pages 424–431, 2007.

[10] J. Gracia and E. Mena. Web-based measure of semantic relatedness. In *WISE*, pages 136–150, 2008.

[11] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, 2004.

[12] Institute for Computational Linguistics of the University of Stuttgart. TreeTagger: A language independent part-of-speech tagger. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

[13] S. Jaillet, A. Laurent, and M. Teisseire. Sequential patterns for text categorization. *Intelligent Data Analysis Journal*, 10(3):199–214, 2006.

[14] S. Jaroszewicz and T. Scheffer. Fast discovery of unexpected patterns in data, relative to a bayesian network. In *KDD*, pages 118–127, 2005.

[15] J. Kamps, R. J. Mokken, M. Marx, and M. de Rijke. Using WordNet to measure semantic orientation of adjectives. In *LREC*, pages 1115–1118, 2004.

[16] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *ECML*, pages 4–15, 1998.

[17] D. H. Li, A. Laurent, and P. Poncelet. Mining unexpected sequential patterns and rules. Technical Report RR-07027 (2007), LIRMM, 2007.

[18] D. H. Li, A. Laurent, M. Roche, and P. Poncelet. Extraction of opposite sentiments in classified free format text reviews. In *DEXA*, pages 710–717, 2008.

[19] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *ICDM*, pages 369–376, 2001.

[20] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD*, pages 121–128, 1998.

[21] A. Markov, M. Last, and A. Kandel. Fast categorization of Web documents represented by graphs. In *WEBKDD*, pages 56–71, 2006.

[22] A. K. McCallum. Bow: A toolkit for statistical language modeling text retrieval, classification and clustering. http://www.cs.cmu.edu/˜mccallum/bow, 1996.

[23] K. McGarry. A survey of interestingness measures for knowledge discovery. *The Knowledge Engineering Review*, 20(1):39–61, 2005.

[24] B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. In *KDD*, pages 94–100, 1998.

[25] B. Padmanabhan and A. Tuzhilin. On characterization and discovery of minimal unexpected patterns in rule discovery. *IEEE Transactions on Knowledge and Data Engineering*, 18(2):202–216, 2006.

[26] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, pages 271–278, 2004.

[27] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *EMNLP*, pages 79–86, 2002.

[28] S. P. Ponzetto and M. Strube. Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research*, 30:181–212, 2007.

[29] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[30] H. Schmid. Probabilistic Part-of-Speech tagging using decision trees. In *NeMLaP*, 1994.

[31] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[32] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *KDD*, pages 275–281, 1995.

[33] Y. Song, D. Zhou, J. Huang, I. G. Councill, H. Zha, and C. L. Giles. Boosting the feature space: Text classification for unstructured data on the Web. In *ICDM*, pages 1064–1069, 2006.

[34] M. Spiliopoulou. Managing interesting rules in sequence mining. In *PKDD*, pages 554–560, 1999.

[35] W. Su, J. Wang, and F. H. Lochovsky. Automatic hierarchical classification of structured deep Web databases. In *WISE*, pages 210–221, 2006.

[36] P. D. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *ECML*, pages 491–502, 2001.

[37] P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424, 2002.

[38] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*, 2005.

[39] T. Wilson, J. Wiebe, and R. Hwa. Recognizing strong and weak opinion clauses. *Computational Intelligence*, 22(2):73–99, 2006.

[40] X. Yan, J. Han, and R. Afshar. CloSpan: Mining closed sequential patterns in large databases. In *SDM*, pages 166–177, 2003.

[41] Y. Yang and C. G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 13(3):252–277, 1994.

[42] H. Yu, J. Han, and K. C.-C. Chang. PEBL: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004.

[43] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP*, pages 129–136, 2003.

[44] T. Zesch, C. Müller, and I. Gurevych. Using wiktionary for computing semantic relatedness. In *AAAI*, pages 861–866, 2008.