Recognition of Sarcasm in Microblogging Based on Sentiment Analysis and Coherence Identification

Piyoros Tunghamthiti†, Kiyoaki Shirai† and Masnizah Mohd†

Recognition of sarcasm in microblogging is important in a range of NLP applications, such as opinion mining. However, this is a challenging task, as the real meaning of a sarcastic sentence is the opposite of the literal meaning. Furthermore, microblogging messages are short and usually written in a free style that may include misspellings, grammatical errors, and complex sentence structures. This paper proposes a novel method for identifying sarcasm in tweets. It combines two supervised classifiers, a Support Vector Machine (SVM) using N-gram features and an SVM using our proposed features. Our features represent the intensity and contradictions of sentiment in a tweet, derived by sentiment analysis. The sentiment contradiction feature also considers coherence among multiple sentences in the tweet, and this is automatically identified by our proposed method using unsupervised clustering and an adaptive genetic algorithm. Furthermore, a method for identifying the concepts of unknown sentiment words is used to compensate for gaps in the sentiment lexicon. Our method also considers punctuation and the special symbols that are frequently used in Twitter messaging. Experiments using two datasets demonstrated that our proposed system outperformed baseline systems on one dataset, while producing comparable results on the other. Accuracy of 82% and 76% was achieved in sarcasm identification on the two datasets.

Key Words: Sarcasm, Microblogging, Sentiment Analysis, Coherence, Concept Knowledge

1 Introduction

Social networks are web-based applications that allow messages, photos, videos, music and other personal information to be sent to or shared with specific groups of people (e.g., family, friends, or community). Recently, social network providers such as Twitter, Facebook, and Google have reported a rapid increase in the number of users. This paper focuses on the analysis of texts on Twitter. Twitter has over 200 million active users each month (O’Carroll 2012), generating messages at a peak rate of over 230,000 tweets per minute (Ashtari 2013). Tweets are short messages, of which opinions, evaluations, and judgments often constitute an important part (Pak and Paroubek 2010). Users are limited to 140 characters per tweet. Tweets use a free writing style, and may include complex sentence structures. This may create difficulties in understanding

† School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology
Sarcasm is an ironic or satirical style of discourse in which the intended meaning is reversed. It is often used to insult or criticize a person or a thing. Recognition of sarcasm is one of the most challenging tasks in natural language processing (NLP). The task mainly focuses on determining whether the actual meaning of a given tweet is coincident with its literal meaning. It has potential applications in many other areas of NLP, especially opinion mining and sentiment analysis in social networks. Identification of sarcasm would enable consumers to obtain more accurate information about people’s opinions on topics in different domains (including politics, potential purchases and sports). It could supply companies or service providers with more precise information on the opinions of users about their products or services. This would help them to improve their planning, decision making, or business strategy.

This paper introduces a novel method for deciding whether a given tweet is sarcastic. Our sarcasm recognition system is based on supervised machine learning. In addition to the conventional N-gram of words, several extra features are derived from sentiment analysis of the tweet. These take account of the intensity and contradiction of the sentiment in the given tweet. Punctuation and special symbols that appear frequently in Twitter are also used as features. One of the important characteristics of our method is that the system considers the coherence among multiple sentences in the tweet to derive the sentiment contradiction feature. Although the contradiction of sentiment is one useful clue by which sarcasm is identified, the contradictions in incoherent sentences might not support the conclusion that they are sarcastic. We propose a sophisticated method for identifying coherence in a tweet based on unsupervised clustering and an adaptive genetic algorithm. Furthermore, a concept expansion mechanism is introduced to improve sentiment analysis. Since sentiment analysis often suffers from the use of unknown opinion words that do not appear in the sentiment lexicon, concepts related to unknown words provide help in identifying sentiment polarity.

The rest of the paper is organized as follows. Section 2 discusses previous work on the analysis of sarcasm. Section 3 describes our proposed methods. Section 4 reports the results of experiments conducted to evaluate the proposed methods. Finally, section 5 concludes the paper.

2 Related Work

Sarcasm is defined as “saying the opposite of what you mean” (Quintilien and Butler 1953). It arises from contradictory intentions, which violate the expectations and felicity conditions of the listener (Stringfellow Jr. 1994; Grice 1975; Gibbs Jr. and Colston 2007). Sarcasm is also
understood as the use of ironic statements to express disdain in the guise of approval (Doerfler 2012). In sarcasm, ridicule or mockery is used harshly, often crudely and contemptuously, for destructive purposes (Singh 2012).

A wide variety of studies have analyzed sarcasm or irony in microblogging. A method of sarcasm detection based on semi-supervised pattern acquisition and classification with the acquired pattern-based and punctuation-based features has been proposed (Tsur, Davidov, and Rappoport 2010; Davidov, Tsur, and Rappoport 2010). Reyes and Rosso (2012) characterized irony using six kinds of features: n-grams, POS-grams, funny profiling, positive/negative profiling, affective profiling, and pleasantness profiling. Naive Bayesian classifier, Support Vector Machine (SVM), and decision tree were used to train classifiers for irony detection, achieving an acceptable level of accuracy. This was extended into a more complex model (Reyes, Rosso, and Veale 2013). This method introduced a new set of features at four levels: signatures, degree of unexpectedness, style, and emotional scenarios. Barbieri and Saggion (2014) proposed a method based on seven sets of lexical and semantic features, including the frequency of the words in reference corpora, their intensity, their written or spoken nature, their length, and the number of related synsets in WordNet (Miller 1995).

Following the definition of sarcasm, contradiction of sentiment in the tweet is an important clue for the identification of sarcasm. Riloff, Qadir, Surve, De Silva, Gilbert, and Huang (2013) introduced a novel bootstrapping algorithm that automatically learned lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. The learning process relied on the assumption that a positive sentiment verb phrase usually appears to the left of a negative situation phrase in a sarcastic tweet. The bootstrapping algorithm was comprised of the following two steps. The first step involved the learning of negative situation phrases that followed positive sentiment phrases, with “love” used as the initial seed of the positive sentiment word. In the second step, positive sentiment phrases that occurred near the negative situation phrases were learned. After iterating these two steps, the list of negative situation and positive sentiment phrases obtained was used to identify sarcasm by checking whether the tweet contained the positive sentiment word in close proximity to the negative situation phrase. Their method yielded some improvement in recall for sarcasm identification. However, the method was limited to a restricted number of specific syntactic structures. Also, sarcasm could not be identified accurately when it was constructed across separate clauses or multiple sentences. Sentiment contradiction was also considered in the sentiment analysis of figurative language (including sarcasm) in tweets (Xu, Santus, Laszlo, and Huang 2015).

Carvalho, Sarmento, Silva, and de Oliveira (2009) investigated the use of a set of pre-defined
surface patterns (i.e., emoticons, onomatopoeic expressions for laughter, heavy punctuation marks, quotation marks, and positive interjections) in newspaper articles. They showed that surface patterns were a much more accurate guide (from 45% to 85%) than deeper linguistic analysis. Thelwall, Buckley, and Paltoglou (2012) assessed the SentiStrength sentiment lexicon in a variety of online contexts. Their results showed that the use of punctuation, including single punctuation marks, repetitive punctuation marks, question marks, and exclamation marks, played a key role in improving the sentiment score. Since punctuation and special symbols such as emoticons are often used in tweets to emphasize the user’s emotional attitude, they should be taken into account in sarcasm identification.

This paper presents a supervised method for identifying sarcasm in texts posted on Twitter. Results from sentiment analysis are used, including the contradiction of sentiments used in Riloff’s method (Riloff et al. 2013). However, we introduce a unique sentiment contradiction feature, which also considers coherence within the tweet. Punctuation and symbols play a central role in our method, because their use is a distinctive feature of tweets.

Coherence identification is used in discourse analysis to analyze relationships within written texts. In previous research, linguistic coherence has been defined in a variety of ways. Bartlett (1932) defined coherence as a “continuity of senses.” Similarly, coherence has been understood as “the mutual access and relevance within a configuration of concepts and relations” (Beaugrande and Dressler 1981). Soon, Ng, and Lim (2001) introduced a complex method that linked coreferencing noun phrases both within and across sentences. Twelve features were used to create a set of feature vectors from the documents; then, a classifier was learned from training documents. C5 (Pandya and Pandya 2015) was used as the learning algorithm in this study. The results showed performance comparable to that of state-of-the-art non-learning-based systems on the MUC-6 and MUC-7 standard datasets. However, their method mainly focused on coreference resolution, rather than on the identification of coherent relationships. We have previously presented a naive method for coherence identification in tweets using a set of heuristic rules based on grammatical relations (Tunghanthitthi, Shirai, and Mohd 2014). In this paper, we introduce a more sophisticated method, based on unsupervised clustering and an adaptive genetic algorithm.

3 Proposed Method

Our sarcasm identification method uses a supervised learning framework. Figure 1 shows the overall system. First, an input tweet is pre-processed by removing stop words, lemmatization, and so on. Next, four kinds of features are extracted: a word N-gram (N = 1, 2, 3), a sentiment
score, sentiment contradiction, and punctuation and special symbols. Two classifiers are applied to judge whether the given tweet is sarcastic. One is an SVM classifier that uses the N-gram features; the other is an SVM that uses the remaining features. These classifiers are trained from labeled data, i.e., a collection of tweets with sarcasm tags. A simple voting method is applied to determine the final judgment. If two classifiers disagree, the result that has a larger margin, defined as the distance between a vector of the given tweet and a separate hyperplane, is chosen. In the rest of this section, we will explain in more detail how the classifiers are trained.

3.1 Data Preprocessing

Before the SVMs are trained, the tweets to be used as training data are preprocessed. First, the Stanford Lemmatizer\(^2\) is applied to identify the part-of-speech (POS) and to transform the words into lemma form. Usernames, URLs, and hashtags are removed, since they are weakly informative in sarcasm classification. This preprocessing is applied each time a new tweet is classified.

3.2 Proposed Features

In addition to the standard N-gram features, we use three further features to characterize the properties of sarcasm. Although the basic ideas behind our proposed features are shared with previous work on the analysis of sarcasm, the ways in which the features are extracted are different. The uniqueness of our sentiment contradiction feature will be explained in section 3.2.2. Note that, in this study, the feature vector is binary; a weight of 1 is given to the feature if it exists in the tweet, and 0 otherwise.

---

\(^1\) We used LIBLINEAR (https://www.csie.ntu.edu.tw/~cjlin/liblinear/) with default parameters.

\(^2\) http://nlp.stanford.edu/software/corenlp.shtml
3.2.1 Sentiment score feature

Sarcasm is characterized by violations of expectations and of felicity conditions (Stringfellow Jr. 1994). We therefore use sentiment analysis to recognize the level of violation and of aggressiveness in the words of the tweet. The sentiment score feature represents the intensity of positive or negative sentiment in the tweet. This is measured from the sentiment scores of the words, derived from a public sentiment lexicon. Our method also considers the concepts of the words when conducting sentiment analysis. Since many potential sentiment words are not listed in the sentiment lexicon, our system is designed to extract the concept of an unknown sentiment word to improve the accuracy of the sentiment analysis. This procedure is called “concept expansion” in this paper. Let us consider an example of a sarcastic tweet. T1.

T1: I love going to work on holidays.

Suppose that the system can identify only the positive word “love” from the sentiment lexicon, while the polarity of the other words is unknown. By concept expansion, we can recognize that the word “work” refers to “tiring” or “stressful situation” and that “holiday” refers to “day where person stay home and relax.” The system is now able to identify the two additional sentiment words “holiday” and “work,” which represent positive and negative sentiments, respectively. In this way, concept expansion can compensate for gaps in the sentiment lexicon.

In this research, two lexicons are used to obtain the sentiment scores of the words: SentiStrength (Thelwall et al. 2012) and SenticNet (Cambria and Hussain 2012). SentiStrength is a sentiment lexicon that uses linguistic information and rules to detect sentiment in English text. The lexicon provides positive and negative sentiment scores for various types of polarity words such as booster words, question words, emotion words, negation words, slang, idioms, and emoticons. The score is represented as an integer from −5 to +5 where a larger absolute value represents a stronger sentiment. SenticNet is a sentiment lexicon giving sentiment scores for common-sense concepts. The score is scaled from −1 to +1 to signify the polarity and intensity of the sentiment. The score from SenticNet is multiplied by 5 and rounded so that the sentiment scores from both lexicons are represented by an integer from −5 to +5. Finally, the polarity score of the word \( w \), \( po\text{-}score(w) \), is defined by Equation (1), where \( score_{SS} \) and \( score_{SN} \) are the scores given by SentiStrength(SS) and SenticNet(SN), respectively.

\[
po\text{-}score(w) = \begin{cases} 
\frac{1}{2} \left( score_{SS}(w) + score_{SN}(w) \right) & \text{if } w \in SS \text{ and } w \in SN \\
\text{score}_{SS}(w) \text{ or } score_{SN}(w) & \text{if } w \in SS \text{ or } w \in SN \\
0 & \text{otherwise}
\end{cases}
\] (1)
The extended polarity score of $w$, $ex_{po\_score}(w)$, is given by Equation (2)

$$ex_{po\_score}(w) = \begin{cases} \text{po\_score}(w) & \text{if } w \in SS \text{ or } w \in SN \\ \frac{1}{|C|} \sum_{c \in C(w)} \text{po\_score}\_c(c) & \text{if } C(w) \text{ is derived by concept expansion} \\ 0 & \text{otherwise} \end{cases}$$ (2)

where $C(w)$ is a set of concepts expanded from the word $w$. The details of concept expansion will be explained in subsection 3.3.

Finally, the sum of the sentiment scores of all positive or negative words in the tweet, $sum_{pos\_score}$ or $sum_{neg\_score}$, is calculated using Equation (3) or (4)

$$sum_{pos\_score} = \sum_{pos\_w} ex_{po\_score}(pos\_w)$$ (3)

$$sum_{neg\_score} = \sum_{neg\_w} ex_{po\_score}(neg\_w)$$ (4)

where $pos\_w$ or $neg\_w$ is a word whose extended sentiment score is positive or negative. We define six sentiment score features “po-degree”, where po is either “positive” or “negative” and degree is a rating of “low,” “medium,” or “high.” The features “positive-low,” “positive-medium,” and “positive-high” are activated if $sum_{pos\_score} = 0$, $0 < sum_{pos\_score} \leq 2$, and $sum_{pos\_score} > 2$. The range of sentiment scores for each class is determined based on our intuition. The sentiment score features for negative polarity are defined similarly.

### 3.2.2 Sentiment contradiction feature

As noted above, sarcasm is normally present in a sentence that expresses a meaning that is the opposite of the intended meaning. Therefore, we also use sentiment analysis to find contradictions in sentiment polarity among words in the tweet. Although conflicts in polarity have been used in previous analyses of sarcasm, a novel characteristic of our proposed feature is the detection of coherence in the tweet. We assume that coherence plays a significant role, because sentiment contradiction in incoherent sentences may not indicate sarcasm. Consider the following two examples.

**T2:** I am coughing and choking. I am feeling great.

**T3:** I am coughing and choking. Mary is still fine.

Since both negative words (“coughing” and “choking”) and positive words (“great” or “fine”) are present, sentiment contradiction is detected in both T2 and T3. However, T2 is sarcastic, while T3 is not. In T2, the two sentences are coherent, or related each other. This coherence is
captured by the fact that the pronoun “I” is the subject of both sentences. T2 clearly contains a contradiction in logical meaning, through the use of opposing words (coughing and choking $\neq$ great). Thus T2 can be classified as sarcastic. In contrast, in T3 there is no signifier of logical connection between the words in the two sentences, as the negative and positive words clearly refer to the different subjects “I” and “Mary.” In this case, the tweet can be regarded as non-sarcastic even though a sentiment contradiction was found.

The sentiment contradiction feature is represented as either $contra$ or $contra+coher$. The feature $contra$ is activated if two conditions are satisfied: 1) the tweet consists of only one sentence and 2) a contradiction in sentiment score is found as both $sum_{pos}\_score$ (defined by Equation (3)) and $sum_{neg}\_score$ (Equation (4)) are greater than 0. The feature $contra+coher$ is activated if three conditions are fulfilled: 1) the tweet consists of two or more sentences, 2) contradiction in the sentiment score is found, and 3) the tweet is classified as coherent or incoherent. The procedures for identifying coherence in the tweet will be discussed in subsection 3.4.

### 3.2.3 Punctuation and special symbols feature

Many studies have shown that punctuation plays an important role in text communications – signaling pauses, a change in the tone of voice, or indicating strong feelings or exaggeration. Punctuation is influential in text classification, especially in the area of sentiment analysis. Thus, punctuation and special symbols are used as features in our research. Seven symbols or words are treated as punctuation or special symbol features: 1) emoticons, 2) repetitive sequences of punctuation, 3) repetitive sequences of characters, 4) capitalized words, 5) slang or booster words, 6) exclamation marks, and 7) idioms. The frequency of these elements in the tweet, denoted as $fre$, is classified into three classes: “low” ($fre = 0$), “medium” ($1 \leq fre \leq 3$), and “high” ($fre > 3$). The range of frequency for each class was determined through preliminary experiments. Our punctuation and special symbol features are represented as pairings of one of seven elements and three frequency classes, introducing 21 features in total.

### 3.3 Concept Expansion and Pruning

#### 3.3.1 Concept expansion

Next we explain the procedure for deriving the set of concepts, i.e., $C(w)$ in Equation (2). A concept lexicon called ConceptNet 5.0$^3$ is used to expand the concepts of a word whose sentiment score is unknown. ConceptNet 5.0 is a semantic network comprising common-sense knowledge

---

$^3$ http://conceptnet5.media.mit.edu/
and concepts, represented in the form of nodes (words or short phrases) and labeled edges (relationships between them). For example, the sentence “A dog is an animal” is parsed into an assertion as “dog/IsA/animal.” The assertion consists of two nodes (“dog” and “animal”) and one edge (“IsA”). There are 31 types of relationship, such as “PartOf,” “UsedFor,” “MadeOf,” etc. ConceptNet contains over 800,000 assertions. These assertions are ranked based on voting by users, to ensure the quality and significance of each assertion. In our method, for each word $w$, the top five ranked concepts are set as $C(w)$.

Since the concepts in ConceptNet 5.0 can be represented as phrases (such as “day where person stay home and relax”), the polarity score of the concept $c$ ($po\_score\_c(c)$ in Equation (2)) is defined as the average of the polarity score of the words in the concept:

$$\text{po\_score}\_c(c) = \frac{1}{|C|} \sum_{w \in C} \text{po\_score}(w)$$

(5)

where $C$ stands for the set of sentiment words in $c$.

### 3.3.2 Concept pruning

Although concept expansion is an effective way of recognizing the polarity of an unknown sentiment word, some irrelevant concepts are generated. They may cause errors when applied to sarcasm identification. For example, five concepts can be expanded from “holiday” in the tweet T4.

T4: The typhoon is still blowing hard. What is a nice holiday!

holiday => [“special day,” “day where person stay home and relax,” “special event celebrate by person,” “special day that celebrate event,” “day where person do not have to work”]

In this case, the concept “special event celebrate by person” should not be expanded, since “holiday” in T4 refers not to an event but to a day. We introduce a procedure called “concept pruning” to prevent such irrelevant concepts from being expanded.

Concept pruning is done in three steps: 1) word sense disambiguation (WSD) 2) keyword extraction, and 3) similarity measurement. In the first step, WSD is performed to find the actual meaning of the unknown sentiment word within the tweet. SenseLearner 2.0\(^4\) is used to determine the WordNet sense of the word. In the second step, the disambiguated sense and one of the five most highly ranked concepts of the word are represented as a set of keywords $K_s$ and $K_c$, respectively. $K_s$ is a set of words in the gloss of the WordNet sense $s$, while $K_c$ is a set of words in the concept $c$. Only nouns, verbs, adjectives and adverbs are extracted as keywords. In

\(^4\) http://lit.csci.unt.edu/~senselearner/
the final step, the similarity between $K_s$ and $K_c$ is measured by Equation (6).

$$\text{sim}(K_s, K_c) = \max_{w_s \in K_s, w_c \in K_c} \text{sim}_\text{word}(w_s, w_c)$$

(6)

In this study, Word2Vec (Mikolov, Chen, Corrado, and Dean 2013) and Resnik’s algorithm (Resnik 1995) are used to compute the word similarity $\text{sim}_\text{word}(w_s, w_c)$. If $\text{sim}(K_s, K_c)$ is greater than a threshold value $T_c$, the concept $c$ is kept, otherwise it is pruned.

The parameter $T_c$ was optimized on development data. Figure 2 shows the accuracy of sarcasm identification at different threshold values of $T_c$. This suggested that Word2Vec provides better accuracy than Resnik’s algorithm, and that accuracy is greatest when $T_c = 0.2$. From these results, we chose Word2Vec as the word similarity measure, and set $T_c$ as 0.2.

### 3.4 Coherence identification

This subsection presents our method for identifying coherence among the sentences in a tweet. The proposed method is based on a clustering-based adaptive genetic algorithm (CAGA). The coherence identification used in this study creates clusters of coherent and incoherent tweets for a given set of tweets. The overall procedure of coherence identification is shown in Figure 3. In the training phase, the input set of tweets was annotated with coherence tags to indicate whether each tweet was coherent or not. We then constructed a manually annotated data set consisting of 800 coherent tweets and 800 incoherent tweets.

To cluster the coherent and incoherent tweets, each tweet was represented as a feature vector.

---

5 The details of the development data will be introduced in Subsection 4.1.
6 The ranges of $\text{sim}(K_s, K_c)$ of Word2Vec and Resnik’s algorithm are $[-1,1]$ and $[0,10]$, respectively.
7 http://nlp.stanford.edu/ner/
Let us suppose that sentence $s_1$ precedes $s_2$ in the tweet; then, the word $w_1$ or $w_2$ is the subject, noun, or pronoun in $s_1$ or $s_2$. The features used for clustering are summarized in Table 1. The detailed procedure by which the semantic class agreement feature (9th feature) is derived as follows.

1. For each $i = 1$ and $2$, the sense of $w_i$, called $\text{sense}_i$, is disambiguated by SenseLearner 2.0.
2. A set of hypernyms and hyponyms of $\text{sense}_i$ and $\text{sense}_i$ itself, called $\text{SHH}_i$, is created.
3. The similarity of all possible pairs of synsets from $\text{SHH}_1$ and $\text{SHH}_2$ is measured using a method proposed by Resnik (Resnik 1995).

**Table 1** Features for clustering coherent/incoherent tweets

<table>
<thead>
<tr>
<th>Feature feature 1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pronoun feature 1</td>
<td>$w_1$ is reflexive, personal or possessive pronoun.</td>
</tr>
<tr>
<td>Pronoun feature 2</td>
<td>$w_2$ is reflexive, personal or possessive pronoun.</td>
</tr>
<tr>
<td>String match feature</td>
<td>$w_1$ and $w_2$ are identical.</td>
</tr>
<tr>
<td>Definite noun phrase feature</td>
<td>$w_2$ starts with the determiner “the.”</td>
</tr>
<tr>
<td>Demonstrative noun phrase feature</td>
<td>$w_2$ starts with “this,” “that,” “these,” or “those.”</td>
</tr>
<tr>
<td>Both proper names feature</td>
<td>Both $s_1$ and $s_2$ contain named entities. Proper names are recognized by the Stanford Named Entity Recognizer (NER).7</td>
</tr>
<tr>
<td>Coreference resolution feature</td>
<td>$s_1$ and $s_2$ contain coreference resolution property recognized by Stanford Deterministic Coreference Resolution System.</td>
</tr>
<tr>
<td>Number agreement feature</td>
<td>$w_{c1}$ and $w_{c2}$ agree in number (i.e., they are both singular or plural), where $w_{c1}$ and $w_{c2}$ are judged as coreferred by the Stanford Deterministic Coreference Resolution System.</td>
</tr>
<tr>
<td>Semantic class agreement feature</td>
<td>$w_1$ and $w_2$ are semantically similar.</td>
</tr>
<tr>
<td>Acronym or abbreviation feature</td>
<td>A tweet contains an acronym or abbreviation (i.e., “lol,” “ynwa”).</td>
</tr>
<tr>
<td>Emoticon feature</td>
<td>A tweet contains an emoticon (i.e., “😊,” “:-)”).</td>
</tr>
</tbody>
</table>
(4) The feature is activated when the similarity of one of the synset pairs is greater than a threshold value. This was set to 1.37, based on intuition.

The last two features in Table 1 were introduced because acronyms, abbreviations, and emoticons are often treated as separate sentences. In other words, the isolated acronym/abbreviation/emoticon is always considered to be coherent with the other sentences. If three or more sentences are present, the feature vector is constructed as follows. The feature vectors of all pairs of sentences, denoted by $\vec{c}_{ij}$, are created. The value of each dimension is set at 0 if the values at the same dimension of all $\vec{c}_{ij}$ are 0, and otherwise at 1.

After the extraction of feature vectors, unsupervised clustering is performed. We trialed three representative clustering algorithms: K-means, the EM (expectation maximization) algorithm (McLachlan and Krishnan 2008), and hierarchical clustering. Note that the number of clusters ($N_c$) should be predefined in these algorithms. $N_c$ is optimized using training data.

The remaining problem is to set the weights of the feature vectors. The initial weights are binary: if the feature is present, the weighting is 1, whereas if the feature is absent, the weighting is 0. The weights of the present and absent features are then optimized using an adaptive genetic algorithm (GA). The weight of each feature is changed from $-1$ to $+1$ in steps of 0.2. The GA searches for an optimal set of feature weights at which the clustering performance is highest.

In the test phase, the tweets in the test data are converted to feature vectors with the optimized weights; then, unsupervised clustering is performed. The cluster labels are used as the coherence feature for sarcasm identification. That is, $contra+coher$ described in 3.2.2 is a set of $N_c$ features represented as $contra+cl_i$, where $cl_i$ stands for the i-th cluster to which the tweet belongs.

Table 2 shows the performance of the coherence identification on training data for three clustering algorithms and different number of clusters ($N_c$). Accuracy is defined as the ratio of agreement between the gold and predicted coherence tags. The coherence tag of each tweet is predicted as follows – each cluster is judged to be a cluster of coherent or incoherent tweets by voting the coherence tags of the tweets in the cluster. Then all the tweets within a coherent or incoherent cluster are regarded as coherent or incoherent. As shown in Table 2, the EM algorithm outperformed the others. The greatest accuracy was achieved when $N_c$ was set to 8.

<table>
<thead>
<tr>
<th>$N_c$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>.7711</td>
<td>.7748</td>
<td>.7770</td>
<td>.7795</td>
<td>.7792</td>
<td>.7790</td>
<td><strong>.7816</strong></td>
<td>.7807</td>
<td>.7801</td>
<td>.7769</td>
</tr>
<tr>
<td>EM</td>
<td>.8130</td>
<td>.8151</td>
<td>.8172</td>
<td><strong>.8202</strong></td>
<td>.8173</td>
<td>.8135</td>
<td>.8144</td>
<td>.8159</td>
<td>.8133</td>
<td>.8102</td>
</tr>
<tr>
<td>Hierarchical</td>
<td>.7322</td>
<td>.7326</td>
<td>.7349</td>
<td>.7363</td>
<td><strong>.7387</strong></td>
<td>.7372</td>
<td>.7343</td>
<td>.7344</td>
<td>.7329</td>
<td>.7300</td>
</tr>
</tbody>
</table>

Table 2 Accuracy of coherence identification
The advantage of the CAGA is that clustering of the tweets is performed in an unsupervised manner. Note that the coherence tags of the tweets are used only for the determination of feature weights by the GA, optimization of the number of clusters $N_c$, and selection of the clustering algorithm (K-means, EM, or hierarchical). If the training data is not annotated with coherence tags, the CAGA can be run as follows. For each step of the iterative learning of feature weights using the GA, our sarcasm identification system is trained on the obtained coherent/incoherent clusters, and the accuracy of sarcasm identification on development data is measured. The GA searches for the optimal feature weights at which sarcasm recognition is maximized. This procedure is indicated by the dotted lines in Figure 3. Similarly, optimization of $N_c$ and selection of the clustering algorithm are also possible. Therefore, although a significantly greater computational cost is incurred, manual annotation of the coherence to training data can be avoided. In this sense, the CAGA can be regarded as a semi-supervised method.

4 Evaluation

4.1 Data

Two datasets were used in our experiments: 1) the ARTK (automatically retrieved tweets using keywords) dataset and 2) the SemEval-2015 Task 11 dataset. The ARTK dataset consists of 300,000 tweets, of which 150,000 were selected as sarcastic, while the remaining 150,000 were selected as non-sarcastic. To identify sarcastic tweets, the hashtag “#sarcasm” was used as the query keyword. The non-sarcastic tweets were retrieved by searching with keywords randomly selected from WordNet. Clearly, some sarcastic tweets are posted without the “#sarcasm” hashtag. We checked a random sample of 300 tweets lacking this hashtag, and found that only 3.7% were sarcastic. We therefore ignored this noise in the experiment. Twitter4J\(^9\) was used as the tool for preparing the collection of tweet data. The development dataset of 30,000 tweets that was used for parameter optimization of $T_c$ was constructed in the same way. We also used the SemEval-2015 Task 11 dataset for sentiment analysis of figurative language in Twitter. This is a collection of tweets annotated with a sentiment score of between −5 and +5. This dataset contained hashtags indicating figurative language, such as #sarcasm, #irony, #metaphor, and so on. Tweets with hashtags #sarcasm or #irony were regarded as being sarcastic, and the remainder as non-sarcastic. Note that #irony tweets were categorized as sarcastic, since we found the difference between sarcasm and irony to be very subtle, making them difficult for human

\(^9\) http://twitter4j.org/en/index.html
evaluators to distinguish. The training set contained 8,000 tweets, while the test set contained 4,000 tweets. In this dataset, 35% of the tweets were sarcastic.

4.2 Task

The experimental task was to identify the sarcasm class (sarcastic or not) of a given tweet. The proposed system as well as the baselines were trained and tested by 5-fold cross validation on the ARTK dataset. On the SemEval-2015 task 11 dataset, the classifiers were trained from training data and evaluated on test data.

Three baselines were established for comparison with our proposed methods. Baseline 1 was created by assuming that sarcasm usually occurs in a sentence that expresses a meaning opposite to the intended meaning. Therefore, tweets that contained both a positive and negative word ($\text{sum}_{\text{pos}} \text{score} > 0$ and $\text{sum}_{\text{neg}} \text{score} > 0$) were regarded as sarcastic. Baseline 2 was an SVM trained with only N-gram (uni-gram, bi-gram, and tri-gram) features. Since the N-gram is a common and well-known feature in sarcasm identification, this was considered a strong baseline. Baseline 3 followed the method proposed by Riloff et al. (2013).

The evaluation criteria were the accuracy of sarcasm classification as well as recall, precision, and an F-measure for retrieval of sarcastic tweets. In the following discussion, accuracy was the main measure used to compare the methods.

4.3 Results

Table 3 shows the accuracy (A), recall (R), precision (P), and F-measure (F) of the different methods on the ARTK and SemEval datasets. Baseline 1 achieved 0.58 and 0.57 accuracy on the two datasets. Interestingly, the performance of Baseline 1 was acceptable, even though the

<table>
<thead>
<tr>
<th>Method</th>
<th>ARTK</th>
<th></th>
<th></th>
<th>SemEval</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>A</td>
<td>R</td>
</tr>
<tr>
<td>Baseline 1 (sentiment contra.)</td>
<td>.5847</td>
<td>.5466</td>
<td>.6276</td>
<td>.5843</td>
<td>.5672</td>
<td>.5454</td>
</tr>
<tr>
<td>Baseline 2 (N-gram)</td>
<td>.7517</td>
<td>.7347</td>
<td>.7597</td>
<td>.7470</td>
<td>.7332</td>
<td>.7100</td>
</tr>
<tr>
<td>Baseline 3 (Riloff et al. 2013)</td>
<td>.7841</td>
<td>.7592</td>
<td>.8268</td>
<td>.7916</td>
<td>.7579</td>
<td>.7440</td>
</tr>
<tr>
<td>Proposed features</td>
<td>.6285</td>
<td>.7138</td>
<td>.6076</td>
<td>.6564</td>
<td>.6041</td>
<td>.6601</td>
</tr>
<tr>
<td>N-gram &amp; proposed features</td>
<td><strong>.8234</strong></td>
<td><strong>.7649</strong></td>
<td><strong>.8533</strong></td>
<td><strong>.8064</strong></td>
<td>.7557</td>
<td>.7256</td>
</tr>
</tbody>
</table>

We implemented this system ourselves, but closely matching the original method, since we deliberately followed the detailed algorithm presented in Riloff et al. (2013).
method made no use of machine learning, but relied only on contradictions of sentiment polarity identified from the sentiment lexicon. Baseline 2 and Baseline 3 were more accurate than Baseline 1, and Baseline 3 performed best among the three baselines against all criteria. This suggested that a machine learning approach is appropriate for the identification of sarcasm.

The last two rows of Table 3 show the results for our proposed methods. The accuracy of the SVM trained only with our proposed features was 0.63 on the ARTK and 0.60 on the SemEval datasets, or approximately 12% lower than Baseline 2. This suggests that N-gram features are informative in the sarcasm identification task. The method in which voting of the SVM classifiers used N-grams and our proposed features achieved the best performance on the ARTK dataset. It outperformed Baseline 3 by 4% in accuracy and 1.5% on the F-measure. However, on the SemEval dataset our method displayed higher precision but lower recall than Baseline 3. On the F-measure and accuracy, Baseline 3 and our method were comparable.

To compare our method with Baseline 3 in finer detail, we divided each dataset into two subsets: a subset consisting of single sentence tweets and a subset comprised of multiple sentences. Table 4 shows the average length of words per tweet and the proportion in these subsets, while Table 5 compares the performance of the two methods on each subset. It was found that,

<table>
<thead>
<tr>
<th>Types of tweets</th>
<th>ARTK</th>
<th>SemEval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg. length (words)</td>
<td>Proportion</td>
</tr>
<tr>
<td>Single sentence</td>
<td>12.24</td>
<td>66%</td>
</tr>
<tr>
<td>Multiple sentences</td>
<td>16.18</td>
<td>34%</td>
</tr>
</tbody>
</table>

Table 4  The average length and percentage of single and multiple sentence tweets in the ARTK and SemEval datasets

<table>
<thead>
<tr>
<th>Types of tweets</th>
<th>ARTK dataset</th>
<th></th>
<th></th>
<th>SemEval dataset</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>R</td>
<td>P</td>
<td>F</td>
<td>A</td>
<td>R</td>
</tr>
<tr>
<td>Single sentence</td>
<td>.8142</td>
<td>.8180</td>
<td>.8396</td>
<td>.8287</td>
<td>.7914</td>
<td>.7987</td>
</tr>
<tr>
<td>N-gram &amp; our proposed features</td>
<td>.8070</td>
<td>.8034</td>
<td>.8174</td>
<td>.8103</td>
<td>.7407</td>
<td>.7180</td>
</tr>
<tr>
<td>Multiple sentences</td>
<td>.7268</td>
<td>.7208</td>
<td>.7586</td>
<td>.7392</td>
<td>.7062</td>
<td>.7038</td>
</tr>
<tr>
<td>N-gram &amp; our proposed features</td>
<td>.8362</td>
<td>.8309</td>
<td>.8467</td>
<td>.8388</td>
<td>.7883</td>
<td>.7853</td>
</tr>
</tbody>
</table>

Table 5  Accuracy of sarcasm identification on different length of tweet data

11 A single SVM classifier using both N-gram and proposed features was also evaluated. Its performance was worse than the voting of two classifiers, with accuracies of 0.7856 and 0.7193 on the ARTK and SemEval datasets, respectively.
while our method worked well for the multiple sentence subset, Riloff’s method did not. Our proposed method achieved accuracies of 0.83 and 0.78 for multiple sentence tweets, which was approximately 11% higher than Baseline 3 on the ARTK dataset and 8% higher on the SemEval dataset. Let us consider the sarcastic tweet “I had a fever last night, still coughing as if I’m choking. Soon I won’t be able to eat either. Things are going well.” Note that the positive word “well” and three negative words “fever,” “coughing,” and “choking” appear far apart from each other. Since Riloff’s method checks for the existence of a positive sentiment phrase and a negative situation phrase within a five-word window, it fails to find a sentiment contradiction in this example. However, our method is able to correctly classify it as sarcastic. In contrast, our method performed worse than Riloff’s on single sentence tweets. One reason for this is that our method sometimes wrongly identifies a sentiment contradiction in a long sentence. Consider the non-sarcastic tweet “I’m feeling so irritable right now & I just want to go home & not speak to anyone & take a rest.” It contains one positive word “rest” and one negative word “irritable.” Since our method simply checks for the existence of positive and negative words to identify a sentiment contradiction, it misclassifies this tweet as sarcastic. In contrast, since Riloff’s method strictly checks for positive sentiment phrases and negative situation phrases, it successfully classifies it as non-sarcastic. When the sentence is long, our system is more prone to such false identification of sentiment contradiction. Our method performed better for single sentence tweets in the ARTK dataset than the SemEval dataset, since the average length of single sentence tweets in the ARTK dataset is shorter than that of the SemEval dataset. Similarly, our method is superior to Riloff’s method on the ARTK dataset but comparable on the SemEval dataset, again because the single sentence tweets are longer in the SemEval dataset.

Both Riloff’s method and our methods make use of N-gram and sentiment contradiction features, although the way these features are driven is different. However, sentiment scores and punctuation and special symbol features are used only in our method. These were also used in many previous studies (Tsur et al. 2010; Davidov et al. 2010; Reyes and Rosso 2012; Reyes et al. 2013).

4.3.1 Contribution made by the features

To evaluate the effectiveness of our proposed features, classifiers lacking one type of feature were compared with the complete system. Table 6 shows the results of this experiment. The fifth row (− Coherence identification) denotes a system that does not take account of coherence.

12 Note that coherence is not considered for the single sentence tweets in our method.
In this system, the sentiment contradiction feature is always activated if positive and negative words are detected.

Among the proposed features, the sentiment contradiction feature that considered the coherence of the tweet made the greatest contribution. This feature may capture the linguistic aspects of sarcasm. Note that the system that used the sentiment contradiction feature without considering coherence (fifth row of Table 6) performed worse than the system that did not use the sentiment contradiction feature (fourth row). This strongly suggests that the identification of coherence is important in sarcasm identification.

The contribution of the sentiment score feature was also significant. Normally, sarcasm contains special elements that generate violations and aggressiveness in communication, especially in negation terms. Therefore, the strength of the sentiment polarity can be used to estimate the level of violation and aggressiveness, helping to identify sarcasm in the tweet.

In contrast, the punctuation and special symbol feature proved less effective, since accuracy fell by only 0.27% and 0.37% on the two datasets when it was removed. Many previous studies have shown that sarcastic communication often contains strong emotional language, and emoticons and heavy punctuation can be used to indicate this. However, although the punctuation and special symbols feature can capture the strong emotions of the user, such emotional tweets do not always use sarcasm. Nevertheless, the feature did contribute a small gain in performance.

### 4.3.2 Contribution of concept expansion

The contribution made by concept expansion and pruning was also evaluated. Table 7 shows the results for three systems: no concept is expanded, the concepts are expanded but not pruned (the five most related concepts are always expanded), and only the related concepts are obtained by concept expansion and pruning. Table 8 shows the total number of expanded concepts in the test set.

Concept expansion showed improvement against all evaluation criteria on both datasets by a
maximum of 2.4%, with the exception of accuracy on the SemEval dataset. On the ARTK and SemEval datasets, 1.9 and 0.75 concepts per tweet were obtained, respectively. These could be used to identify the polarity of unknown words or slang in the sentiment lexicon. Concept pruning reduced the number of expanded concepts by 33% and 32%, contributing a further improvement. Our pruning method was shown to successfully remove irrelevant concepts.

4.3.3 Contribution of the clustering-based adaptive genetic algorithm

The CAGA also played a part in enhancing the accuracy of the method. Table 9 shows the results when the weights of the feature vector were determined by the CAGA, and when they were not. When the CAGA was not applied, all feature weights were represented as binary values. The CAGA increased accuracy by 1.78% for the ARTK dataset and 0.98% for the SemEval dataset. These results demonstrate that the CAGA played a significant role in predicting the optimized weight for each feature in the clustering of coherent and incoherent tweets. Our experiments showed that the features proper names and demonstrative noun phrase were less important, while semantic class agreement and definite noun phrase were significant features.

Table 7  Effectiveness of concept expansion and pruning

<table>
<thead>
<tr>
<th>Method</th>
<th>ARTK</th>
<th>SemEval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>R</td>
</tr>
<tr>
<td>no concept expansion</td>
<td>0.7951</td>
<td>0.7419</td>
</tr>
<tr>
<td>concept expansion only</td>
<td>0.8187</td>
<td>0.7483</td>
</tr>
<tr>
<td>concept expansion and pruning</td>
<td>0.8234</td>
<td>0.7649</td>
</tr>
</tbody>
</table>

Table 8  Number of expanded concepts

<table>
<thead>
<tr>
<th>Method</th>
<th>ARTK</th>
<th>SemEval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no concept expansion</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>concept expansion only</td>
<td>576,881</td>
<td>11,374</td>
</tr>
<tr>
<td>concept expansion and pruning</td>
<td>385,602</td>
<td>7,722</td>
</tr>
</tbody>
</table>

Table 9  Effectiveness of feature weighting by GA

<table>
<thead>
<tr>
<th>Method</th>
<th>ARTK</th>
<th>SemEval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>R</td>
</tr>
<tr>
<td>w/o feature weighting by GA</td>
<td>0.8056</td>
<td>0.7341</td>
</tr>
<tr>
<td>with feature weighting by GA</td>
<td>0.8234</td>
<td>0.7649</td>
</tr>
</tbody>
</table>
4.4 Limitations

Through error analysis of the experimental results, some limitations in our method were found. First, in sentiment identification and concept expansion many words were misinterpreted, which produced misclassification of sarcastic tweets. Consider the following example tweet: “I had a terrible fight with my friend in the garden. I beat him around the bush.” Our concept expansion module obtained one positive concept “come out better in a competition race” and one negative concept “give a spanking to” from the word “beat.” Since the intensity of the positive concept is stronger than the negative concept in this example, our system will recognize “beat” as a positive word. Thus sentiment contradiction is wrongly identified, causing misclassification of this tweet as sarcastic. Sometimes, inappropriate concepts remain after concept pruning. Second, implicit sentiment in the tweet sometimes produces error. The tweet “I just love how you tweet all these other girls” uses only the positive word “love” and no contradiction of sentiment is detected. The system therefore misclassifies this as a non-sarcastic tweet. However, the phrase “how you tweet all these other girls” shows jealousy on the part of the user and the implication is negative. Since neither the sentiment lexicon nor our concept expansion module is able to detect the sentiment in this phrase, the system cannot identify this tweet as sarcastic. Finally, the method of coherence identification also needs further refinement. Consider as an example the tweet “cold, sad & sleepy. that’s a perfect combo. goodnight.” This tweet contains both a positive word (“perfect”) and negative words (“cold” and “sad”). However, our coherence identification method failed to identify coherence in the tweet, causing a false-negative error. Recall that the demonstrative noun phrase feature is used in our method. The demonstrative “that” in this tweet may indicate coherence between the first and second sentences. However, the weight of this feature from training by the CAGA is too small to identify coherence.

Other general causes of errors were found. The #sarcasm hashtag is sometimes used to indicate the sarcasm of other, previously posted tweets. Consider the tweet “@john4768 That was #sarcasm.” In this example, it is not the tweet itself but a previous tweet by the user “@John” that is identified as sarcastic. In such cases, #sarcasm is not an indicator that the tweet is sarcastic. In addition, our datasets contain many tweets that provide absolutely no clue that sarcasm is intended, other than the hashtag #sarcasm, for example, “I feel great #sarcasm.”

5 Conclusions

In this paper we proposed a novel method for identifying sarcasm in a tweet. Word N-grams, sentiment scores, sentiment contradictions, and punctuation and special symbol features were
used in a supervised machine learning approach. Coherence in the tweet was used to derive the sentiment contradiction feature, which was identified by a clustering-based adaptive genetic algorithm. Concept expansion and pruning were used to enhance the features derived from the sentiment analysis.

The contributions of the paper are as follows. 1) A novel feature is proposed that is able to capture sentiment contradiction in a coherent tweet. Our experimental results demonstrated the significant contribution that coherence identification made to improvements in accuracy. 2) A semi-supervised approach was used to identify coherence in a tweet. This was based on the unsupervised clustering and optimization of the weights of the feature vectors by an adaptive genetic algorithm. 3) An alternative way of tackling the data sparseness of public sentiment lexicons was introduced. This utilizes ConceptNet as a source of additional external knowledge. The effectiveness of our concept expansion and pruning were empirically demonstrated.

In future work, we will extend our approach from recognition of sarcasm to other areas of practical sentiment analysis. More specifically, we will explore ways of estimating the polarity and/or the intensity of sentiment from both sarcastic and non-sarcastic sentences in product reviews and news articles. Our initial work towards this goal was reported in Tungthamthiti, Santus, Xu, Huang, and Shirai (2015). In addition, we intend to apply personalization to the sentiment analysis task. Clearly, the intensity of sentiment will reflect the personality and characteristics of each individual user. Personalization is a promising way of improving the accuracy of sentiment analysis in sarcastic sentences. A more minor research goal is to run our CAGA without the use of coherence annotated data, as discussed in subsection 3.4, and to empirically evaluate the effectiveness of such an approach.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments.

Reference


Piyoros Tungthamthiti: received the B.S. and M.E. from Sirindhorn International Institute of Technology (SIIT), Thammasat University in 2010 and 2012, respectively. He is currently a Ph.D. student at School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology. His research interests are in the area of natural language processing (NLP), human computer interaction (HCI) and knowledge management.

Kiyoaki Shirai: received the B.E., M.E. and Dr. Eng. from Tokyo Institute of Technology in 1993, 1995 and 1998, respectively. He was an assistant at the Graduate School of Information Science and Engineering, Tokyo Institute of Technology from 1998 to 2001. He is currently an associate professor at School of Information Science, Japan Advanced Institute of Science and Technology. His current research interests include natural language processing, especially corpus-based methods and their applications. He is a member of the Association for Natural Language Processing, the Japanese Society of Artificial Intelligence, the Information Processing Society of Japan, and the Institute of Electronics, Information and Communication Engineers.

Masnizah Mohd: received her Ph.D. in Computer and Information Sciences from the University of Strathclyde, United Kingdom; and M.IT (2002) and B.IT (1999) degrees in Information Science from the Universiti Kebangsaan Malaysia. She was a postdoctoral researcher at the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST) from 2014 to 2016. She is currently an associate professor at the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. Her main research interests are in the areas of Information Retrieval, Topic Detection and Tracking, and Natural Language Processing. She is interested in theoretical and practical information retrieval problems. Her research has focused on retrieval models, query/document representations, term weighting, user interaction, tasks and evaluation.

(Received December 23, 2015)
(Revised April 8, 2016)
(Rerevised June 16, 2016)
(Accepted August 9, 2016)