Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches

Piyoros Tungthamthiti, Kiyoaki Shirai, Masnizah Mohd
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi City, Ishikawa, Japan 923-1292
Email: {s1320204,kshirai,masnizah}@jaist.ac.jp

Abstract

Sarcasm is a form of communication that is intended to mock or harass someone by using words with the opposite of their literal meaning. However, identification of sarcasm is somewhat difficult due to the gap between its literal and intended meaning. Recognition of sarcasm is a task that can potentially provide a lot of benefits to other areas of natural language processing. In this research, we propose a new method to identify sarcasm in tweets that focuses on several approaches: 1) sentiment analysis, 2) concept level and common-sense knowledge 3) coherence and 4) machine learning classification. We will use support vector machine (SVM) to classify sarcastic tweet based on our proposed features as well as ordinary N-grams. Our proposed classifier is an ensemble of two SVMs with two different feature sets. The results of the experiment show our method outperforms the baseline method and achieves 80% accuracy.

1 Introduction

Recognition of sarcasm is one of the most difficult tasks in natural language processing (NLP). It is a problem of determining if the actual meaning of a word is intended in a given context. Sarcasm is normally represented in a form of ironic speech in which the speakers convey an implicit message to criticize a particular person. Thus, tone of voice plays a significant role in the communication. There are many communication programs (e.g. Line, Facebook, Twitter), which allow to communicate together through only text characters. It is very difficult to determine the actual meaning by just looking at the text itself. Recognition of sarcasm prevents us from misinterpreting sentences whose meaning are opposite to their literal meaning. It is also a task that is potentially applicable for many other areas of NLP, for example, machine translation, information retrieval, information extraction and knowledge acquisition.

Twitter is an online social networking service that allows users to post and read short messages, called "tweets". However, Twitter allows users to write short messages, i.e. 140 characters per tweet. Also, users usually post a lot of tweets in complex sentence structures. Regarding to these issues, a new method is created to detect sarcasm in tweets.

Sarcasm is known as "the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry" (Macmillan, 2007). According to this definition, we can recognize sarcasm by evaluating the polarity of the sentences. In other words, a sarcastic sentence contains two or more words, which may cause conflict in sentiment polarities (both positive and negative) in a sentence, whereas a normal sentence should contain at most one polarity. Let us consider the example sentence "I love being ignored." The sentence contains both positive ("love") and negative word ("ignored") in a sentence. Therefore, it can be classified as a sarcastic sentence.

In the identification of sarcasm based on the contradiction of the polarity, unknown words in the sentiment lexicon are serious problem. To tackle it, we try to consider the related concepts for each word to identify the sentence polarity. For example, let us consider the tweet "It's Wednesday and it's freezing! It's raining! How better can this day be??" This would be classified as a normal tweet since only

28th Pacific Asia Conference on Language, Information and Computation pages 404–413

the word "better" is recognized as a positive word from the whole tweet. However, our approach can recognize it as a sarcastic tweet by using the concept level knowledge. That is, we can know "bad weather" is one of the related concepts of "raining" from an extra lexical resource, then we have a new concept "bad" (negative) together with the original word "better" (positive) to catch the contradiction in sentiment polarity in the sentence.

In addition, we also consider "coherence"; that is, the relationships across multiple sentences. Generally, sarcastic tweets should contain expressions which clearly show the relationships or references to some words across sentences. For example, in the tweet "And I just found out that my other pap fell and broke his hip. Awesome day thus far", the word "awesome" (positive) refers to the action "fell" and "broke" (both are negative words), that is contradiction of sentiments in the sarcastic tweet. However, when a tweet contains contradiction of sentiment polarity without coherence between them, it could be regarded as non-sarcastic tweet. For example, in the tweet "He likes dogs. She hates cats.", the word "love" (positive) and "hates" (negative) refer to the different subjects in two sentences. Although the tweet contains contradictions in sentiment polarity, the two sentences are not coherent. Therefore, it should not be classified as a sarcastic tweet. In this way, coherence is important for the recognition of sarcasm.

Finally, Support Vector Machine is used to train a classifier that judges if a tweet is sarcastic. Two SVMs will be trained with two different feature sets. One is N-gram, the other is features based on the sentiment score, coherence and punctuation. Then we will combine two SVMs, that is, more reliable judgment between two classifiers are chosen as the final result.

In this paper, we propose a new method to utilize several major modules, including 1) sentiment analysis, 2) expansion of concept level and commonsense knowledge 3) coherence identification and 4) machine learning classification. Figure 1 represents the overall process of our method. The method will try to merge our newly introduced features obtained from the module 1, 2 and 3 together with the commonly used features (e.g. N-grams) to enhance the classification performance in sarcastic tweets. Using the data consisting of 50,000 tweets, we will evaluate our results by comparing against two baseline methods derived from definition of sarcasm and supervised learning algorithm based on N-gram features.

2 Related work

Currently, there are several researches related to the recognition of sarcasm. A variety of methods have been proposed based on various kinds of techniques, including statistical models, sentiment analysis, pattern recognition, supervised or unsupervised machine learning. However, the intelligence system and computation process are not sufficient to be relied on for sarcasm recognition. It also requires the development of understanding forms of language in both psychological and linguistic aspects.

According to Stingfellow (1994) and Gibbs et al. (2007), the use of irony and sarcasm is studied to derive a definition and demonstrate some characteristics of sarcasm. Both studies agree on the similar basis that irony and sarcasm arise from the contradictory intentions represented by the opposed meaning of an ironic or sarcastic statement. These studies also discover the theories of verbal irony comprehension 1) that verbal irony requires a violation of expectations, and 2) that it requires violation of felicity conditions for speech acts. Thus, if we observe both contradictory intentions and violation of felicity conditions within a context, we can recognize a sarcastic context.

Tsur et al. (2010) present a semi-supervised learning method to classify sarcastic sentences on Twitter, Amazon and in online product reviews. The method employs two main modules: 1) semi-supervised pattern acquisition and 2) a classification algorithm. It extracts a sequence of high-frequency word (HFWs) and content words (CWs) as a pattern of a sarcastic sentence. Then, it constructs a single feature vector for each pattern. The feature value for each pattern will be calculated based on their similarities comparing to the other extracted patterns. Finally, the method will apply k-nearest neighbours (kNN)-like strategy together with the feature vector to classify the sentences. This method is based on an alternative idea which does not focus on the semantic analysis but on the sequence of HFWs and CWs as sentence

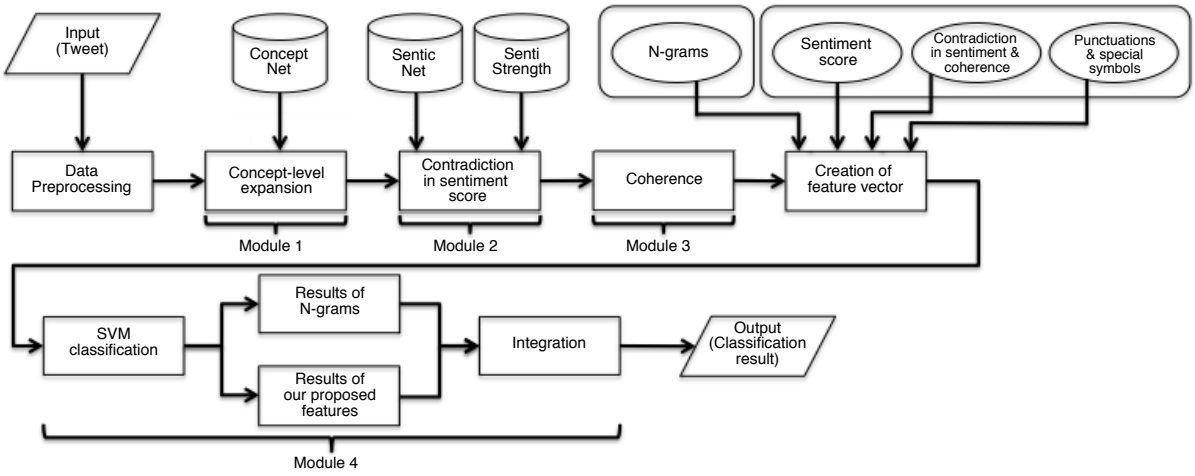

Figure 1: Flowchart of overall process of our method

patterns; this method relies on the syntactic level of natural language processing.

Ellen et al. (2013) introduce a method to identify sarcasm in tweets that arises from a contrast between a positive sentiment referring to a negative situation. In order to learn phrases corresponding to positive sentiments and negative situations, this method uses a bootstrapping algorithm that keeps iteration between two steps. The first step is learning negative situation phrases following positive sentiment, where "love" is used as an initial seed word. Then, the second step will learn positive sentiment phrases that occur near negative situation phrases. After multiple iteration processes, the obtained list of negative situations and positive sentiment phrases are used to recognize sarcasm in tweets by identifying contexts that contain a positive sentiment in close proximity (occurring nearby) to a negative situation phrase. This method relies on the assumption that many sarcastic tweets contains the following structure:

$$[+VERB\ PHRASE][-SITUATION\ PHRASE]$$

However, the method has some limitations since it cannot identify sarcasm across multiple sentences.

Coreference resolution is a task in natural language processing to identify multiple words or phrases that refer the same entity such as person, place or thing. Soon et al. (2001) introduce a machine learning approach to link coreferring noun phrases both within and across sentences. They construct a feature vector consisting of 12 features. The features include distance, antecedent pronoun, anaphor-pronoun, string matching, definite noun phrase, demonstrative noun phrase, number agreement, semantic class agreement, gender agreement, both-proper-names, alias and appositive features. Then, a classifier will be trained based on the feature vectors generated from the training documents. C5 (Quinlan, 1993; Quinlan, 2007) is used as the learning algorithm in this study. This research is the first machine-learning based system that offers performance comparable to that of state of the art non-learning based systems on MUC-6 and MUC-7 standard datasets. In this study, a simple coreference resolution method is applied to identify coherence of multiple sentences.

Language can be expressed in many different ways, such as utterance, action, signal and text. According to the definition of sarcasm, we also need to consider violation and aggressiveness of the communication. For utterance, we can easily recognize the emotion through the unsterilized tone of voice (Tepperman et al., 2006). In texts, punctuation plays a vital role in text communication to provide the reader the signals about pause, stop and change of tone of voice. Let us consider an example sentence "That is very annoying!". The exclamation mark (!) can be used to indicate a strong feeling or exaggerates something. Thelwall et at. (2012) aim to assess the sentiment lexicon (SentiStrength) in a va-

riety of different online contexts. One part of this research discusses the usage of punctuations in various contexts. It focuses on the sentence that contains a single punctuation, repetitive punctuation marks, question marks and exclamation marks. Their result shows that punctuation plays a key role to boost the sentiment score.

The characteristic of our method is that we attempt to combine multiple approaches in both psychological and linguistic aspects to develop an innovative strategy. Our method takes various approaches into account, including sentiment analysis, concept level knowledge expansion, coherence and N-gram of words. Tweets are represented by feature vectors based on these methods. Then classifiers for sarcasm identification are trained by supervised machine learning.

3 Data

In this section, the procedures of data collection and data preprocessing will be explained.

3.1 Source

We first prepare a collection of tweets by using Twitter4J[1] as a tool to retrieve tweets data. Tweets are not just simple text data since they contain URL addresses, twitter usernames (mentions) or hashtags. For example, in the tweet "Congrats to @Kelly_clarkson on the birth of her baby GIRL! http://eonli.ne/1vgXVOU #gorgeous", "@Kelly_clarkson" is a username, "http://eonli.ne/1vgXVOU" is an URL and "#gorgeous" is a hashtag. Users can attach an URL to the tweet when they want provide more information or show an image related to the post. Twitter also contains a mention feature (e.g. @<username>), which allows the notification of other users about the tweet. Hashtags (e.g. #<texts>) are used to mark keywords or topics in a tweet. Although the usage of these meta tags is optional, they frequently appeared in a lot of tweet messages.

Two datasets are required in our study: 1) sarcastic tweets and 2) normal tweets. Different query keywords will be used for each datasets. To collect sarcastic tweets, the hashtag "#sarcasm" is used. That is, tweets with #sarcasm are retrieved via Twitter API. Normal tweets are retrieved based on randomly selected keywords from WordNet lexicon (Miller, 1995).

3.2 Preprocessing

Two kinds of preprocessing are performed on tweet datasets: 1) lemmatization and 2) usernames, URLs and hashtags removal. For lemmatization, we use the Standford Lemmatizer[2]. Usernames, URLs and hashtags are removed from tweets as they do not provide any information about the concepts or sentiments of the words and might be noise for the classification process.

4 Proposed method

Below we propose our method based on four major modules. They are the modules to generate a set of classification features or to classify a tweet if it is sarcastic.

4.1 Concept level and common-sense knowledge

Concept level and common-sense knowledge are the ability to perceive, understand and acknowledge things, which are shared through the common knowledge or facts that can be reasonably realized. In this research, we focus on the semantic analysis of tweets using the semantic network consisting of concepts of words to obtain more affective information. Let us consider an example sarcastic sentence "I love going to work on holidays." The system may misclassify it as a normal sentence due to the lack of sentiment information. From this sentence, only the word "love" has positive sentiment score, while the other words have no polarity. However, using concept level and common-sense knowledge, we can know that the word "work" would refer to a tiring or stressful situation and the word "holiday" would refer to "time for rest". Now a contradiction of the polarity in this sentence could be found since ["love" and "holiday"] and ["work"] are positive and negative words, respectively. The sentence could then, be classified as sarcastic.

In this study, we use a concept lexicon called ConceptNet[3]. ConceptNet is a semantic network

[1] http://twitter4j.org/en/index.html

[2] http://nlp.stanford.edu/software/corenlp.shtml

[3] http://conceptnet5.media.mit.edu

consisting of common-sense knowledge 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" will be parsed into an assertion as "dog/IsA/animal". The assertion consists of two nodes ("dog" and "animal") and one edge ("IsA"). There are also other 31 different types of relationships, such as "PartOf", "UsedFor", "MadeOf", etc. ConceptNet contains more than 800,000 assertions. These assertions are ranked based on the number of votes by users. The number of votes are taken as a score to ensure the quality and the significance of each assertion.

ConceptNet will be used to expand the concepts for the words whose sentiment score is unknown. Thus, the sentiment score of the unknown words can be recognized through their generated concepts and definitions. The concept-level lexicon improves the robustness of our system in terms of calculation of the sentiment scores of tweets. The lexicon also allow the system to recognize sarcasm of the sentence at the concept level.

4.2 Contradiction in the sentiment score

As previously explained, sarcasm often occurs in a contradictory form of communication or the use of words to express something opposite to the intended meaning. In this research, we attempt to use sentiment analysis to find contradiction in sentiment polarity between words in a tweet. Two lexicons are used to check the polarities of words: SentiStrength and SenticNet.

SentiStrength is a sentiment lexicon that uses linguistic information and rules to detect sentiment strength in English text. The lexicon consists of all types of polarity words, including booster words, emotion words, negation words, question words, slang words, idioms and emoticons. SentiStrength provides positive and negative sentiment scores for each word. Both scores are integers from 1 to 5, where 1 signifies weak sentiment and 5 signifies strong sentiment. For example, the sentiment score (1,1) represents a neutral word. Basically, the overall polarity of a word is calculated by subtracting the negative sentiment score from the positive sentiment score.

SenticNet is a resource for opinion mining that aims to create a collection of commonly used common-sense concepts with positive and negative sentiment scores. The sentiment score for each word is scaled from -1 to 1, where -1 signifies strongly negative sentiment, 0 signifies neutral sentiment and 1 signifies strong positive sentiment. In this study, the score is multiplied by 5 so that it corresponds to the scores in SentiStrength.

We calculate the sentiment score of the word w, $w\_score(w)$, as shown in Equation (3). If the word is found in SentiStrength or SenticNet, the sentiment score in the lexicon is used as the $w\_score(w)$. If the word is found in both SentiStrength and SenticNet, the average of the sentiment score of both lexicons is used as the $w\_score(w)$. Otherwise, we obtain the concepts to expand the meaning of the word by choosing the top five ranked concepts from ConceptNet lexicon. Then, we take an average of the sentiment scores of the concepts as $w\_score(w)$. After we obtain the sentiment scores for all words, we will calculate the total score for positive and negative words as shown in Equation (1) and (2), respectively.

If both sum_pos_score and sum_neg_score are greater than 0, we can find contradiction of polarity in the tweet. As we will describe in 4.4.2, the total scores will also be used as weights in the feature vector in the classification process.

$$sum\_pos\_score = \sum_{pos\_w \in TW} w\_score(pos\_w) \tag{1}$$

$$sum\_neg\_score = \sum_{neg\_w \in TW} w\_score(neg\_w) \tag{2}$$

$$w\_score(w) = \begin{cases} polarity\_score(w), & \text{if } w \in SS \text{ or } SN \\ average\_polarity\_score(w), & \text{if } w \in SS \text{ and } SN \\ \frac{1}{|C|} \sum_{c \in C} polarity\_score(c), & \text{otherwise} \end{cases} \tag{3}$$

- TW refers to a tweet.
- pos_w and neg_w refers to the positive and negative words.
- w refers to a word.
- c refers to a concept of a word.
- C refers to the top five ranked concepts of a word.
- sum_pos_score and sum_neg_score are the summation of positive and negative sentiment score.
- SS refers to SentiStrength lexicon.
- SN refers to SenticNet lexicon.

4.3 Sentence coherence

Since our study focuses on contradiction in the sentiment score, coherence is another issue that we need to consider. Assume that a tweet consists of multiple sentences with sentiment contradiction. If all sentences are independent on each other, it is not obvious to say that the tweet is sarcastic. Therefore, we introduce a set of heuristic rules to identify coherence across multiple sentences.

In this study, coherence between two sentences is identified by simply checking coreference between subjects or objects of sentences. Let us suppose that sentence s_1 precedes s_2, and word w_1 and w_2 are the subject (or object) of s_1 and s_2, respectively. If w_1 is an antecedent of w_2, we regard the two sentences as coherent. We created the following five rules to check coreference between w_1 and w_2:

1. Pronoun match feature - w_1 and w_2 are identical pronouns, including reflexive pronouns, personal pronouns and possessive pronouns.

2. String match feature - w_1 and w_2 are identical. Note that stopwords are ignored in string matching.

3. Definite noun phrase feature - w_2 starts with the word "the".

4. Demonstrative noun phrase feature - w_2 starts with the "this", "that", "these" and "those".

5. Both proper names feature - w_1 and w_2 are both named entities.

Two sentences are regarded as coherent if they fulfill one of the above rules. If one pair of w_1 and w_2 satisfies our rules among all combination of w_1 and w_2 in multiple sentences in a tweet, we regard the overall tweet as coherent.

Obviously our method is too simple to identify coherence within sentences. In future, a more sophisticated method should be incorporated into our coherence identification module.

4.4 Creation of feature vector

In this section, we will explain how to represent a tweet as a feature vector to train a classifier for sarcasm identification.

4.4.1 N-grams feature

N-gram refers to a sequence of words within a tweet, where N indicates the size (number of words) of a sequence. The common used sizes of N-gram are uni-gram ($N = 1$), bi-gram ($N = 2$) and tri-gram ($N = 3$).

In our dataset, we will divide each tweet into a single word, a sequence of two words and a sequence of three words. They will be used as features. The weights of N-gram features are binary: 1 if N-gram is present in a tweet, 0 if absent.

4.4.2 Contradiction feature

As discussed earlier, contradiction in the sentiment score and coherent within multiple sentences are useful for sarcasm identification. Therefore, we introduce two new binary features, $contra$ and $contra + coher$, considering contradiction of polarity and coherence in the tweet. The feature $contra$ is activated if (1) the tweet consists of one sentence and (2) contradiction of the sentiment score is found by the method described in Subsection 4.2. $contra + coher$ is activated if (1) the tweet consists of two or more sentences, (2) contradiction of polarity is detected and (3) the tweet is judged as coherent by the method described in Subsection 4.3.

4.4.3 Sentiment feature

We also provide sentiment score features for both positive and negative sentiment phrases. In this case, we use three classes (low, $medium$ and $high$) to indicate the degree of positive and negative polarity of the tweet. After conducting a preliminary experiment to find the optimum range of sentiment scores, three positive sentiment features are defined as follows:

pos_low: activated if $sum\_pos\_score \leq -1$

pos_medium: activated if $0 \leq sum\_pos\_score \leq 1$

pos_high: activated if $sum\_pos\_score \geq 2$

neg_low, neg_medium and neg_high are defined in the same way. Note that weights of these 6 sentiment features are binary.

4.4.4 Punctuation and special symbols feature

We also consider punctuation as one of the main features in this study. Many studies have shown that punctuation has a lot of influence in text classification, especially in the area of sentiment analysis. We consider the following 7 indicators to introduce punctuation features:

P_1. Number of emoticons

P_2. Number of repetitive sequence of punctua-

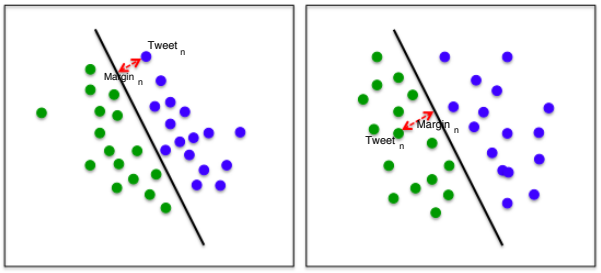

Figure 2: Example of margin based SVM classification approach

tions

P_3. Number of repetitive sequence of characters
P_4. Number of capitalized word
P_5. Number of slang and booster words[4]
P_6. Number of exclamation marks
P_7. Number of idioms[5]

We use low, $medium$ and $high$ as features to indicate the range of number of punctuation and symbols. Through our preliminary experiment to check various range of values from 0 to 7, we found the optimum range to be:
P_i_low: activated if $number = 0$
P_i_medium: activated if $1 \leq number \leq 3$
P_i_high: activated if $number \geq 4$
Thus we introduce 7 $\times$ 3 = 21 new features. Note that these features are binary.

4.5 Classification algorithm

A machine learning algorithm based on the feature vectors generated from the tweets data was used to train a classifier. The classification algorithm used is support vector machine (SVM) due to its simplicity and effectiveness in binary classification. We use the linear kernel to perform the classification task because it does not consume as much time and resources on a large amount of data as polynomial kernel.

To combine our features described from 4.4.2 to 4.4.4 with N-gram feature, we choose an approach in which two feature sets are used separately to train two different SVMs and combine them to get final decision. First, we perform the classification task twice (once for n-grams and once for our features) and obtain two sets of results. Then, we determine the final result by comparing the classification outputs of all data. For each tweet, if the judgments of two SVMs agree, it simply becomes the final result. However, if they do not agree, we need to consider the classification margin for each classifier. Figure 2 demonstrates a situation where two classifiers obtain different classification results for the same tweet. In this case, we need to compare the margin (distance between the data and separate hyperplane) of both classifiers. Usually, the higher the margin, the more reliable the output. Therefore, we take the output from the classifier with higher margin as the final result.

5 Experiment

In our experiment, we retrieved 50,000 tweets from Twitter for our datasets. 25,000 tweets were randomly selected as normal tweets, whereas the other 25,000 tweets are sarcastic tweets. Then, we classified the tweets based on variety of features, including N-grams and our proposed features. The results of the proposed method are compared against two baseline methods. The first baseline is based on the definition of sarcasm. The second baseline uses N-gram features to train an SVM classifier for sarcasm identification.

5.1 Baseline 1

Since sarcasm normally emerges in a sentence that expresses the meaning opposite to the intended meaning, we will consider tweets where both positive and negative scores (Equation (1) and (2)) are greater than 0 to be sarcastic.

5.2 Baseline 2

The other baseline is SVM trained with N-gram features. We prepare two baseline systems: one is SVM with uni-gram features, the other is SVM with uni-gram, bi-gram and tri-gram features.

5.3 Evaluation procedure

We evaluate the two proposed methods: 1) an SVM trained with our proposed feature sets, 2) a classifier combining SVMs with our proposed features and N-gram. Our proposed methods as well as Baseline 2 are evaluated by 10-fold cross validation on our

[4]SentiStrength is used as a lexicon of slang and booster words.

[5]http://www.englishcurrent.com/idioms/esl-idioms-intermediate-advanced/

tweet dataset. Recall, precision, F-measure and accuracy are measured to evaluate the performance of sarcasm identification.

6 Results and discussion

Table 1 shows the results of Baseline 1. The performance is relatively high, although Baseline 1 does not rely on supervised machine learning, but on the sentiment lexicon only. Table 2 reveals results of single SVMs with our proposed features (contradiction, sentiment and punctuation features) and Baseline 2. The accuracy of our proposed method is 63.42%, which is better than Baseline 1 but worse than Baseline 2. We found that N-gram features were still powerful for classification of sarcasm. Table 3 shows the results of the combination of two SVMs. In this table, our individual features are combined with uni-gram separately to evaluate the effectiveness of each feature. The fifth row in Table 3 is the system where coherence in a tweet is not considered[6], while the sixth row indicates the system where ConceptNet is not used for concept expansion. We can find that the combination of N-gram features and all our proposed features improve the accuracy 3% against Baseline 2 with N-grams. It indicates that several sarcastic tweets can be found by our approach but not by N-gram features. Examples of such sarcastic tweets are shown below, where polarity words are in bold:

1. I am **thrilling.** The **storm** in my area
2. A **nice** sunny day to go **pay** some **bills**.....
3. It's **brilliant** to realize when your **best** asset **screw** everything up
4. I really **enjoy** running on the treadmill. So **exhausted**!!
5. It has been **freezing** and **snowing** all week. The weather is so **gorgeous**

Although polarity words in these tweets are effective features, they do not frequently appear in the training data. SVM trained with N-gram features fails to classify them as sarcastic due to data sparseness. Our sentiment, contradiction and punctuation features are rather abstract and appear many times in the training data. Therefore, our method can classify these sarcastic tweets correctly.

[6] $contra + coher$ feature is activated even when coherence in a tweet is not confirmed.

6.1 Contribution of our proposed features

In this subsection, we further discuss the contribution of each proposed feature.

6.1.1 Punctuations and special symbols

As can be seen from Table 3, punctuations and special symbols contribute only a slight improvement. The accuracy is increased by only 0.1% when they are combined with uni-gram. This may be because punctuations and special symbols are also incorporated in uni-gram feature set, that is, our proposed feature is partially duplicated with uni-gram. Nevertheless, the feature provides some improvement to the overall result.

6.1.2 Concept level knowledge expansion

The results show that concept level knowledge expansion can enhance the quality of the sentiment score features from 75.48% to 76.35%. Tweets are unstructured and context free data. There are a lot of unknown words and slang that are very difficult to handle. From this reason, concept level and common sense knowledge can be applied to improve our method.

6.1.3 Effectiveness of coherent identification

As explained in 4.4.2, coherence in the tweet is required to be considered in order to detect contradiction of polarity more precisely. Next we will discuss the contribution of coherence feature. The accuracy decreased by 1% (from 76.35% to 75.48%) when coherence is ignored as shown in Table 3. It is clear that contradiction in the sentiment score with coherence feature has an impact on the improvement of the result. Let us consider a non-sarcastic tweet in our dataset "My gf's mac failed three times and I had to reboot twice. Windows are WAY simpler." Suppose that we ignore coherence when constructing the feature vector. This tweet would be misclassified as a sarcastic tweet since it contains contradiction in the sentiment score of both positive ("simpler") and negative ("fail") words in two different sentences. However, when coherence in the tweet is checked, our method will recognize that the words "My gf's mac", "I" and "Windows" are not related to each other. In other words, coherence does not exist within the tweet. Now it can be correctly classified as a non-sarcastic tweet. As shown in this example,

Table 1: The result of contradiction in sentiment score approach

Methods	Recall	Precision	F-measure	Accuracy
Contradiction in sentiment score (Baseline 1)	0.55	0.56	0.56	57.14%

Table 2: The result of SVM classification based on various features

Methods	Recall	Precision	F-measure	Accuracy
Our proposed features	0.64	0.63	0.63	63.42%
Uni-gram features (Baseline 2)	0.72	0.73	0.73	73.81%
Uni-gram, bi-gram and tri-gram features (Baseline 2)	**0.76**	**0.76**	**0.76**	**76.40%**

Table 3: The result of marjority vote and margin based SVM classification

Methods	Recall	Precision	F-measure	Accuracy
uni-gram and contradiction	0.72	0.72	0.72	72.83%
uni-gram and sentiment score	0.75	0.75	0.75	75.64%
uni-gram and punctuations + special symbols	0.72	0.73	0.73	73.91%
uni-gram and our proposed features without coherence	0.75	0.75	0.75	75.72%
uni-gram and our proposed features without concept level knowledge generation	0.74	0.75	0.75	75.48%
uni-gram and all our proposed features	0.76	0.77	0.76	76.35%
uni-gram, bi-gram, tri-gram and all our proposed features	**0.79**	**0.78**	**0.79**	**79.43%**

contradiction of polarity in an incoherent tweet does not indicate sarcasm.

6.2 Limitation of our approaches

There are some limitations in our method. First, there are a lot of ambiguous words in concept knowledge expansion, which may lead to misclassification of sarcastic tweets. Inappropriate concept expansion causes erroneous detection of contradiction in the sentiment score. For example, the sentence "I love when its raining." contains a positive sentiment word "love" and also negative situation word "rain" whose concept is "bad weather". However, it is not always true that the word "rain" refers to a negative situation. It may cause misclassification. Second, in our dataset, some normal sentences retrieved by random sampling are actually sarcastic although there is no hashtag "#sarcasm". It is rather difficult to prevent it. It means that our collection of tweets is noisy data. Finally, there are a lot of sarcastic sentences, which provide absolutely no clues. An illustrative example is "I feel great #sarcasm". Without "#sarcasm" hashtag, there is no way that we can realize it as a sarcastic tweet.

7 Conclusion

In this research, we present a new method for recognition of sarcasm in tweets. The method is based on a variety of approaches, including sentiment analysis, concept level knowledge expansion, coherence of sentences and machine learning classification. Sentiment scores of words are used as features for the classification. We also use the common-sense concept to find the sentiment score for the word with unknown sentiment score. Then, we consider coherence in a tweet to ensure that the tweets with contradiction in the sentiment score have dependent relationships across multiple sentences. Finally, we construct the feature vector to train an SVM classifier based on our proposed features. N-gram and our proposed features are used to train separate classi-

fiers, then a more reliable judgment between them is chosen as the final result.

We compared our results against two strong baselines. One of them is derived based on the definition of sarcasm and the other is SVM trained with N-gram features. The results show that our method has the greatest accuracy, when we combine our proposed features with N-gram. Although the model with the proposed features achieves only 63.42% accuracy, the results clearly show that our features can help to classify some tweets that the model using only N-gram features cannot identify.

Even for human, it is not easy to identify sarcasm in tweets because sarcasm often depends on common-sense knowledge associated with the context of tweets. It makes automatic identification of sarcasm difficult. We think that about 80% accuracy could be considered a satisfying result.

7.1 Future work

For the future work, we plan to improve the efficiency of our method based on three major issues: 1) coherence and 2) word sense disambiguation 3) evaluation using real data.

In this research, we have provided some heuristic rules to determine coherence within multiple sentences. Coherence may have a lot of influence in the classification, however, the improvement by coherence scheme was not so great in our experiment. We should investigate a better way to identify and incorporate the coherence feature in our model.

Word sense disambiguation is another issue that we need to consider. In our method, we always expand five concepts for each word, that does not exist in SentiStrength or SenticNet lexicon. However, some expanded concepts may be irrelevant with the context of the tweet. Therefore, if we can obtain only the suitable concepts for each word, the performance of our method might increase.

Finally, we tested our system on balanced datasets, where the number of sarcastic and non-sarcastic tweets are equal. However, this situation rarely occurs in a real situation, since the number of non-sarcastic tweets may be much higher than the number of sarcastic tweets. We also need to evaluate our method on an unbalanced dataset and a real dataset.

References

Ellen R., Ashequl Q., Prafulla S., Lalindra S., Gilbert D. S., Gilbert N., Ruihong H. 2013 Sarcasm as Contrast between a Positive Sentiment and Negative Situation *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 704–714, Seattle, Washington

Gibbs, R. W., Colston, H. L. 2007 *Irony in Language and Thought: A Cognitive Science Reader* Lawrence Erlbaum Associates, eds. 2007.

Macmillan, E. D. 2007. *Macmillan English Dictionary*, Macmillan Education, 2 edition.

Miller A. G. 1995 WordNet: A Lexical Database for English *Communications of the ACM*, Vol. 38, No. 11: 39-41.

Quinlan, R. J. 1993. *C4.5: Programs for machine learning*, Morgan Kaufmann San Francisco, CA

Quinlan, R. J. 2007. *C5* Available: http://rulequest.com

Soon W. M., Ng H. T, Lim D. C. Y. 2001 A Machine Learning Approach to Coreference Resolution of Noun Phrases *Computational Linguistics*, pages 521–544, Cambridge, MA, USA

Stringfellow, F. J. 1994. *The Meaning of Irony* NewYork: State University of NY.

Tepperman, J., Traum, D., Narayanan, S. 2006 Sarcasm Recognition for Spoken Dialogue Systems *Interspeech 2006*, Pittsburgh, PA, USA

Thelwall, M., Buckley, K., Paltoglou, G. 2012 Sentiment Strength Detection For The Social Web *Journal of the American society for information science and technology*, 63(1):163–173

Tsur O., Davidov D. 2010 Icwsm - a great catchy name: Semi-supervised recognition of sarcastic sentences in product reviews *In International AAAI Conference on Weblogs and Social*

Tsur O., Davidov D., Rappoport A. 2010 Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala, Sweden