Twitter Sarcasm Detection Exploiting a Context-Based Model

Zelin Wang[1,2(✉)], Zhijian Wu[1,2], Ruimin Wang[3], and Yafeng Ren[2]

[1] State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China
[2] School of Computer, Wuhan University, Wuhan, China
whwzl@whu.edu.cn
[3] International School of Software, Wuhan University,
Bayi Road, Wuhan 430072, China

Abstract. Automatically detecting sarcasm in twitter is a challenging task because sarcasm transforms the polarity of an apparently positive or negative utterance into its opposite. Previous work focus on feature modeling of the single tweet, which limit the performance of the task. These methods did not leverage contextual information regarding the author or the tweet to improve the performance of sarcasm detection. However, tweets are filtered through streams of posts, so that a wider context, e.g. a conversation or topic, is always available. In this paper, we compared sarcastic utterances in twitter to utterances that express positive or negative attitudes without sarcasm. The sarcasm detection problem is modeled as a sequential classification task over a tweet and his contextual information. A Markovian formulation of the Support Vector Machine discriminative model as embodied by the SVM^{hmm} algorithm has been employed to assign the category label to entire sequence. Experimental results show that sequential classification effectively embodied evidence about the context information and is able to reach a relative increment in detection performance.

Keywords: Sarcasm detection · Sentiment classification · Support vector machine · Sequential classification

1 Introduction

Sentiment analysis in twitter has been one of the most popular research topics in NLP (Natural Language Processing) in the past decade, as shown in several recent surveys [1,2]; The goal of sentiment analysis to automatically detect the polarity of a twitter message, while sarcastic or ironic statement transforms the polarity of an apparently positive or negative utterance into it opposite. So it is very important to differentiate sarcastic utterance from the utterances that express positive or negative attitudes without sarcasm [3]. Sarcasm detection is considered to be an important aspect of language which deserves special attention given its relevance in fields such as sentiment analysis and opinion mining [4].

J. Wang et al. (Eds.): WISE 2015, Part I, LNCS 9418, pp. 77–91, 2015.
DOI: 10.1007/978-3-319-26190-4_6

The issue of automatic sarcasm detection in twitter has been addressed in the past few years, the sarcasm detection is usually considered as a classification problem, previous approaches mostly relied on features modeling to the single tweet. These methods did not leverage contextual information regarding the author or the tweet to improve the performance of the task. However, tweets are filtered through streams of posts, so that a wider context, e.g. a conversation or topic, is always available.

Considering the following tweet from our dataset: "@*syydsand* @*gretchlol this seems like a lie. do you no longer associate with yourself?*" Did the author intend this tweet sarcastic? Without additional context it is difficult to know. But if we peruse the author's conversational context which is shown in Fig. 1, we can reasonably inter that this tweet was intended sarcastically.

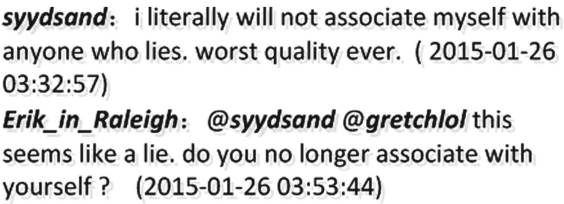

Fig. 1. A tweet and its conversation-based context, the content in parentheses represents the posting time of this tweet

According to the example in Fig. 1, we know the contextual information is benefit to improve the detection performance. This motivated us to detect sarcasm in twitter by using different contextual information.

In this paper, we focus on message-level sarcasm detection on English Twitter using a context-based model along three lines: first, we introduce three different types of contextual information, that are *conversation*, as chains of tweets that are reply to the previous one, *posting history*, also chains of tweets that are come the same author, and *topic*, built based on the same hashtag. Then we focus on feature modeling of tweets, they will also account for contextual information. Finaly, we introduce a more complex classification model that works over an entire tweet sequence and not on one tweet at a time. From a computational perspective, a target tweet and its context are arbitrarily long sequence of messages, ordered according to time with the target tweet being the last. The SVM^{hmm} learning algorithm [5,6] has been employed, as it allow to classify a tweet within an entire sequence. While SVM based classifiers allow to recognize the category label from one specific tweet at a time, the SVM^{hmm} learning algorithm collectively labels all tweet in a sequence.

The contributions of this paper are as follows:

– To the best of our knowledge, the context-based model is proposed to identify sarcasm in twitter for the first time.

– Results show the context-based model can improve the performance for twitter sarcasm detection.
– Results show the history-based context can improve the performance of sentiment analysis in twitter.

2 Related Work

There has recently been a flurry of interesting work on sarcasm detection [7–11]. In these work, verbal irony detection has mostly been treated as a standard text classification task, some innovative approaches specific to detect irony have been proposed.

Carvalho et al. (2009) [12] created an automatic system for detecting sarcasm relying on emoticons and special punctuation, they focused on detection of ironic style in newspaper articles. Veale and Hao (2010) [13] proposed an algorithm for separating ironic from non-ironic similes, detecting common terms used in this ironic comparison. Reyes et al. (2012) [14] have recently proposed a model to detect sarcasm in Twitter, they defined four groups of features: signatures, unexpectedness, style, and emotional scenarios. Moreover, Barbieri and Saggion (2014) proposed a novel linguistically motivated set of features to detect irony in twitter, the features take into account frequency, written/spoken difference, sentiments, ambiguity, intensity, synonymy and structure, experimental results show their model achieves state-of-the-art performance.

There are also a few computational models that detect sarcasm on Twitter and Amazon [3,7,15]. Davidov et al. (2010) proposed a semi-supervised identification, they used 5-fold cross validation on their kNN-like classifiers and obtained 55 % in F-measure on the Twitter dataset. Gonzalez-Ibanez et al. (2011) experimented with Twitter data divided into three categories, they used two classifiers-support vector machine (SVM) with sequential minimal optimization (SMO) and logistic regression, they used various combinations of unigrams, dictionary-based features and pragmatic factors to achieve the better performance. The work of Riloff et al. (2013) detected one type of sarcasm: contrast between a positive sentiment and negative situation. They used a bootstrapping algorithm to acquire lists of positive sentiment phrases and negative situation phrases from sarcastic tweets.

To our knowledge, however, no previous work on sarcasm detection has designed the model which leverages contextual information regarding the author or tweet. But this is very necessary in some cases, some sarcastic utterances can not be recognized by the lack of contextual information. In this paper, we modeled the sarcasm detection problem as a sequential classification task over tweet and his contextual information (one or more tweets, representing conversation, related topic, or posting history). A Markovian formulation of the Support Vector Machine discriminative model as embodied by the SVM^{hmm} algorithm has been employed to assign the category label to entire sequences. Experimental results prove that sequential classification effectively embodied evidence about the contextual information and is able to reach a increment in F1 measure.

3 Dataset

The aim of this paper is to estimate the contribution of the context-based model, existing state-of-the-art approaches neglect the contextual information, so that the datasets with labeled contexts are not available. In this section, we will introduce the dataset used in our method.

3.1 Basic Dataset Construction

In Twitter, people post message of up to 140 characters. Apart from plain text, a tweet may contain references to other users (@user), URLs, and hashtags (#hashtag) which are tags assigned by the user to identify topic or sentiment. Previous work [7,16] also showed that human judge other than the tweets' authors, achieve low levels of accuracy when trying to classify sarcastic tweets. So we argue that using hashtags labeled by their authors of the tweets produces a better quality dataset. In other words, the best judge of whether a tweet is intended to be sarcastic is the author of the tweet. To build the dataset including negative (N), sarcastic (S), and positive (P) tweets, we used a Twitter Streaming API[1] to collect tweets that express sarcasm (#sarcasm, #sarcastic, #irony, #ironic), positive sentiment (e.g. #happy, #joy), and negative sentiment (e.g. #sadness, #angry, #frustrated), respectively. To reduce some noisy tweets, we remove the following tweets:

- we applied automatic filtering to remove retweets, duplicates, quotes, spam, tweets written in language other than English;
- we filtered all tweets where the hashtags were not located at the very end of the message[2].

Finally, we get the 1500 tweets in each of the three categories, sarcastic, positive and negative, which each category includes 500 tweets, respectively. Meanwhile, we remove the hastage which can represent the sarcastic, negative or positive, all 1500 tweets are called basic dataset.

It is worth noting that we can build a classifier to detect sarcasm in twitter based on simple or complex feature modeling. But this paper aim to apply context-based model to improve detection performance. So the classifier (not employing contextual information) built in basic dataset is used to a baseline classifier. Next, we introduce how to get the contextual information and to determine the category label of the context.

3.2 Context Generation

For a tweet, its contextual information is usually embodied by the stream of this tweet, we get the following three contextual information for each tweet in basic dataset using Twitter API:

[1] http://dev.twitter.com/docs/streaming-apis.

[2] To address the concern of Davidov et al. (2010) that tweets with #hashtags are noisy.

- **History-based Context**: An entire tweet sequence can be derived including the multiple tweets preceding the target tweet that are from the same author.
- **Conversation-based Context**: An entire tweet sequence can be derived including the multiple tweets preceding the target tweet that represent the interactive information with other users;
- **Topic-based Context**: An entire tweet sequence can be derived including the multiple tweets preceding the target tweet that contain the same hashtag.

After the extraction of contextual information, we obtain three types of contextual information for each tweet in basic dataset, It is worth noting that not all tweets in basic dataset have contextual information. Finally, we get the contextual information for each tweet in basic dateset, statistical information is showed in Table 1.

Table 1. Basic dataset and contextual information

Category	Basic	History	Conversation	Topic
Negative	500	2224	73	972
Sarcastic	500	2321	267	614
Positive	500	2229	113	1032
Total	1500	6774	453	2618

In Table 1, Basic represents the basic dataset, History represents the history-based context, Conversation represents the conversation-based context and Topic represents the topic-based context. The numbers of tweets are shown in columns 2, 3, 4, and 5, respectively, column 2 represents the basic dataset and column 3–5 represents different contextual information, column 2 (basic dataset) includes 1500 tweets, while column 3–5 represents the subsets of target tweets for which the history-based, conversation-based and topic-based context, respectively, was available. History-based contexts are 6774 tweets (column 3), and topic-based contexts contain 2618 tweets (column 5), while conversation-based contexts only include 453 tweets (column 4).

3.3 Dataset Annotation

To get the dataset with labeled contexts, we need to determine the category label (negative, sarcastic, positive) for each tweet from all contexts. Manual annotation is time-consuming and laborious, so we use a multi-class classifier which is trained based on basic dataset to predict the category label of the tweet in contexts. The disadvantage of this method is that it introduces noise which some contexts will be mislabeled, but it is a realistic solution to determine the category label of contextual information. Experimental results also show the sequential classification get the better performance, though there are some mislabeled tweets in the sequences.

After determining the category label of all tweets from contexts, we can devise the context-based model to detect sarcasm. In the following section, we will prove that sequential classification approach embedding contextual information can get the better performance than multi-class approach (not employing any context). It is worth noting that the dataset (basic dataset and tweets from contexts) used in the paper are automatically constructed without relying on any manually coded resource.

4 The Proposed Approach

This paper proposes a context-based model that exploits the contextual information to detect sarcasm in twitter. Firstly, we formalize three different contextual information which may improve the performance of the task. Secondly, we take consideration of feature modeling of all tweets, we not only use the simple and classical method (Bag of Word, BoW), and use the feature modeling to the closely related nature of social media text. Finally, we introduce the multiple classification approach ($SVM^{multiclass}$) and the sequential classification approach (SVM^{hmm}) to detect sarcasm in twitter.

4.1 Generating Different Contexts

Based on the nature of the twitter, we use the following three types of contextual information.

Conversation-Based Context. In twitter, a target tweet may be the part of the conversation. If we can get the conversational information preceding the target tweet. We are more likely to judge the category label of target tweet with the help of conversational information. Specially, for each tweet $t_i \in \mathcal{T}$, let $r(t_i) : \mathcal{T} \rightarrow \mathcal{T}$ be a function that returns either the tweet to which t_i is a reply to, or *null* if t_i is not a reply. Then, the conversation-based context $\Upsilon_i^{C,l}$ of tweet t_i is the sequence of tweet iteratively built by applying function $r(t_i)$, until l tweets have been selected or $r(t_i) = null$, where l is the number of limiting the size of the context.

History-Based Context. The previous tweets (we called history tweets) about the author of a tweet can reflect author's attitude towards some events or people. History tweets should be useful to improve the detect performance. Specially, for target tweet t_i, an entire tweet sequence can be derived including the l tweets preceding the target tweet t_i that contain the same author. Let $t_i \in \mathcal{T}$ to be a tweet, the history-based context $\Omega_i^{H,l}$ is the sequences of tweets, l is the number of context from the posting history of the author for a target tweet t_i.

Topic-Based Context. In twitter, hashtag represents a topic which can be discussed by other users. We select the tweets with the same hastag in the time window as the third context. Specially, for a target tweet t_i, an entire tweet

sequence can be derived including the l tweets preceding the target tweet t_i that contain the same hashtag set. Let $t_i \in \mathcal{T}$ be a tweet and $t(t_i) : \mathcal{T} \rightarrow \mathcal{P}(\mathcal{H})$ be a function that returns the entire hashtags set $H_i \subseteq \mathcal{H}$ observed into t_i. Then, the topical context $\Gamma_i^{T,l}$ for a tweet t_i is a sequence of the most recent l tweets t_j such that $H_i \cap H_j \neq \emptyset$, i.e. t_j and t_i share at least one hashtag, and t_j has been posted before t_i.

For different contexts, a specific context size l can be imposed by focusing only on the last l tweets of the sequences. According to the above method, we get the three different types of contexts.

4.2 Feature Engineering

For a tweet, different approaches of feature modeling have been used in many work [6,17]. This paper aims at applying the context-based model to detect sarcasm in twitter, we use the following two types of feature modeling methods to represent a tweet, respectively.

Bag of Word. The bag of word (BoW) is the simple method which describes the lexical overlap tweets, thus represented as vectors, whose dimensions corresponding to the different words. Components denote the presence or not of the corresponding word in twitter. Even if it is simple, the BoW model is one of the most informative representations in sentiment analysis and text classification [18].

Word Cluster. The disadvantage of Bow is the sparsity of the word space. Meanwhile, twitter message belongs to social media so that there are many nonstandard word in twitter message, e.g. *be4* (before), *2gether* (together) and *loveee* (love). These nonstandard words make the space more sparse. So we presented another word representations based on word clusters to explore shallow semantic meanings and reduced the sparsity of the word space.

Owoputi et al. (2013) obtained hierarchical word clusters via Brown clustering [19,20] on a large set of unlabeled tweets[3]. The algorithm partitions the words into a base set of 1,000 clusters, and induces a hierarchy among those 1,000 clusters with a series of greedy agglomerative merges that heuristically optimize the likelihood of a hidden Markov model with a one-class-per-lexical type constraint. In their word cluster, many variants of standard words are considered as the same class or closed-class, including pronouns (u = "you") and prepositions (be4 = "before"). if we use this word cluster to represent a tweet, we can get only 1000 dimensions vectors. This word clusters provided by CMU pos-tagging tool[4] were used to represent a tweet. For each tweet we recorded the number of words from each cluster, resulting in 1000 features.

[3] This method is found from Liang (2005), https://github.com/percyliang/brown-cluster.

[4] http://www.ark.cs.cmu.edu/TweetNLP/.

4.3 Modeling Sarcasm Detection as a Sequential Classification Problem

For a tweet and its contexts, once different feature representations are available, a sequential classification approach, based on the SVM^{hmm} [5] will be introduced, as an explicit account of different contexts. To prove the effectiveness of sequential classification approach, we first discuss a multi-classification schema (named $SVM^{multiclass}$) proposed in [21], as the baseline in this paper.

The Multi-class Approach. The $SVM^{multiclass}$ schema [21] is applied to implicitly compare all category labels and select the most likely one, using the multi-class formulation described in [22]. The algorithm acquires a specific function $f_y(x)$ for each category label $y \in \mathcal{Y}$, where $\mathcal{Y} = \{negative, sarcastic, positive\}$. Given a feature vector $x \in \mathcal{X}$ representing a tweet t_i, $SVM^{multiclass}$ allows to predict a specific category label $y^* \in \mathcal{Y}$ by applying the discriminant function $y^* = argmax_{y \in \mathcal{Y}} f_y(x)$, where $f_y(x) = w_y * x$ is a linear classifier associated to each category label y. Given a training set $(x_1, y_1) \ldots (x_n, y_n)$, the learning algorithm determines each classifier parameters w_y by solving the following optimization problem:

$$\begin{aligned} min \quad & \frac{1}{2} \sum_{i=1 \ldots k} ||w_i||^2 + \frac{C}{n} \sum_{i=1 \ldots n} \eta_i \\ s.t. \forall i, \forall y \in \mathcal{Y} : & \; x_i \cdot w_{y_i} \geq x_i \cdot w_y \\ & + 100 \Delta(y_i, y) - \eta_i \end{aligned} \tag{1}$$

where C is a regularization parameter that trades off margin size and training error, while $\Delta(y_i, y)$ is the loss function that returns 0 if $y_i = y$, and 1 otherwise.

The Sequential Classification. The category label prediction of a target tweet can be seen as a sequential classification task over this tweet and its context, and the SVM^{hmm} algorithm can be thus applied. Given an input sequence $\mathbf{x} = (x_1 ... x_n) \subseteq \mathcal{X}$, where $\mathbf{x}$ is a tweet and its context, e.g. the conversation-based, topic-based or history-based context, x_i is a feature vector representing a tweet, the model predicts a label sequence $y = (y_1 ... y_l) \in \mathcal{Y}^+$ after learning a linear discriminant function $F : \mathcal{P}(\mathcal{X}) \times \mathcal{Y}^+ \rightarrow \mathcal{R}$ over input/output pairs. The labeling $f(\mathbf{x})$ is thus defined as: $f(\mathbf{x}) = argmax_{\mathbf{y} \in \mathcal{Y}^+} F(\mathbf{x}, \mathbf{y}, \mathbf{w})$. It is obtained by maximizing F over the response variable $\mathbf{y}$, for a specific given input $\mathbf{x}$. i.e. $F(\mathbf{x}, \mathbf{y}, \mathbf{w}) =< \mathbf{w}.\phi(\mathbf{x}, \mathbf{y}) >$. As ϕ extracts meaningful properties from an observation/label sequence pair $(\mathbf{x}, \mathbf{y})$, in SVM^{hmm}, it is modeled through two types of features: interactions between attributes of the observation vectors x_i and a specific label y_i (i.e. **emissions** of x_i by y_i) as well as interactions between neighboring labels y_i along the chain (**transitions**). In other words, ϕ is defined so that the complete labeling $\mathbf{y} = f(\mathbf{x})$ can be computed efficiently from F, using a Viterbi-like algorithm, according to the linear discriminant function

$$\mathbf{y}^* = argmax_{\mathbf{y}\in\mathcal{Y}^+}\{\sum_{i=1\ldots l}[\sum_{j=1\ldots k}(x_i \cdot w_{y_{i-j}}) + \phi_{tr}(y_{i-j},...,y_i)\cdot w_{tr}]\} \tag{2}$$

In the training phase, giving training examples $(\mathbf{x}^1,\mathbf{y}^1)...(\mathbf{x}^n,\mathbf{y}^n)$ of sequence of feature vectors $\mathbf{x}^j$ with their correct tag sequences $\mathbf{y}^j$, SVM^{hmm} solves the following optimization problem

$$\begin{aligned} min \quad & \frac{1}{2}\sum_{i=1\ldots k}||w_i||^2 + \frac{C}{n}\sum_{i=1\ldots n}\eta_i \\ s.t. \quad & \forall y,n:\{\sum_{i=1\ldots l}(x_i^n \cdot w_{y_i^n}) + \phi_t r(y_{i-1}^n, y_i^n) \\ & \cdot w_{tr}\} \geq \{\sum_{i=1\ldots l}(x_i^n \cdot w_{y_i^n}) + \phi_t r(y_{i-1}^n, y_i^n) \\ & \cdot w_{tr}\} + \triangle(y^n, y) \end{aligned} \tag{3}$$

where $\triangle(y^n, y)$ is the loss function, computed as the number of misclassified labels in the sequence, $(x_i \cdot w_{y_i})$ represents the emissions and $\phi_{tr}(y_{i-1}, y_i)$ represents the transitions, Indeed, through SVM^{hmm} learning the category label for the target tweet is made dependent on its context. The markovian setting thus acquires pattern across tweet sequences to recognize the category label even for truly ambiguous tweets.

5 Experiments

The aim of this paper is to estimate the contribution of the proposed model in performance based on different scenarios, whereas different contexts (e.g. conversation) are possibly made available or just singleton tweet, with no context, are targeted.

5.1 Experimental Setup

A first experiment has been run to validate the effectiveness of contextual information over tweets. Based on basic dateset and contextual information, the different settings are adopted corresponding to different classification approaches:

- *multi-class*: Based on basic dataset, *multi-class* approach ($SVM^{multiclass}$) is applied, which does not require any context and can be considered as a baseline.
- *conversation*: Based on basic dataset and conversation-based context, *conversation* refers to the sequential tagging classifier (SVM^{hmm}) observing the conversation-based context. The training and test of the classifier is here run with different context sizes (1, 3 or 5), by parameterizing l in $\Upsilon_i^{C,l}$;

– *history*: Based on basic dataset and history-based context, *history* refers to the sequential tagging classifier (SVM^{hmm}) observing the history-based context. Different context sizes (1, 3 or 5) have been considered, by parameterizing l in $\Omega_i^{H,l}$;
– *topic*: Based on basic dataset and topic-based context, *topic* refers to the sequential tagging classifier (SVM^{hmm}) observing the topic-based context. Different context sizes (1, 3 or 5) have been considered, by parameterizing l in $\Gamma_i^{T,l}$.

In our experiment, the performance evaluation is always carried out against one target tweet. We use the 10-fold cross validation to evaluate performance. Performance scores are reported in terms of precision, recall and F-measure.

5.2 Experimental Results

Based on BoW, experimental results of sarcasm detection are showed in Table 2, we can know that *multi-class* (not employing any context) can get 52.67 % in F-measure. We can get the better performance by using sequential classification approaches. When we use the history-based context, the performance of SVM^{hmm} will improve with the increment of the number of sizes, we can get 58.32 % in F-measure when l is set to 5. For the conversation-based context, the performance will be improved 2 % when l is set to 1, 3, or 5. This tells us that the conversation-based context is very effective and stable. If we use the topic-based context, the proposed approach will experience a performance drop in F-measure, but the precision have a big improvement, we will discuss it later.

Table 2. Evaluation results using BoW

Methods	Precision(%)	Recall(%)	F-Value(%)
multi-class	50.16	55.38	52.67
history-1	46.29	64.08	53.75
history-3	45.01	69.64	54.68
history-5	51.46	67.27	**58.32**
conversation-1	42.07	80.14	55.15
conversation-3	41.14	84.97	55.34
conversation-5	40.93	84.86	55.22
topic-1	59.67	39.62	47.62
topic-3	65.79	17.86	28.10
topic-5	62.16	38.74	47.73

Based on word cluster, experimental results of sarcasm detection are showed in Table 3, we can know that *multi-class* (not employing any context) can get 54.54 % in F-measure, there is 2 % improvement than *multi-class* based on Bow.

This tells us that word cluster is very effective. Meanwhile, We can get the better performance by using sequential classification approaches. When we use the history-based context, the performance of SVM^{hmm} will improve with the increment of the number of sizes, we can get 60.32 % in F-measure when l is set to 5. For the conversation-based context, the performance will be improved 2 % when l is set to 1, 3, or 5. This tells us that the conversation-based context is very effective and stable. Like the Bow, the topic-based context can not get the better performance in F-measure, but get the high precision.

Compared with the current best system ***CURRENT*** [23] in which use a complex set of linguistically motivated, easy-to-computer features from the single tweet, context-based model outperforms the current system. The main reason is that our model uses the contextual information which is very useful to detect sarcasm in twitter.

Table 3. Evaluation results using word cluster

Methods	Precision(%)	Recall(%)	F-Value(%)
multi-class	51.85	58.20	54.54
history-1	47.94	65.00	55.18
history-3	46.15	72.80	55.96
history-5	53.68	70.40	**60.32**
conversation-1	43.12	83.00	56.72
conversation-3	42.08	87.20	56.74
conversation-5	41.60	87.80	56.42
topic-1	64.05	38.00	47.56
topic-3	67.90	15.60	25.20
topic-5	62.60	39.80	48.50
CURRENT	52.37	58.63	55.31

5.3 Experimental Analysis

To analyze the impact of different context, we compute the precision of the sequences in which the length is greater than 2. In other words, we only care to the target tweet which have context. This paper aims at detecting sarcasm in twitter, so we analyze the target tweet (500 tweets) which its category label is sarcastic in basic dataset. l is set to 5, their related information are shown in Table 4.

In Table 4, NUMBER represents the number of the context, SEQUENCE represents the number of the sequences which are accurately predicted, TARGET represents the number of sequences in which the target tweet is accurately predicted, P1 represents the proportion of S in N, and P2 is the proportion of T in N.

Table 4. Evaluation results of the sarcastic tweets including the contexts

Context	NUMBER	SEQUENCE	TARGET	P1(%)	P2(%)
history	498	66	355	13 %	71 %
conversation	172	92	172	53 %	100 %
topic	131	2	35	2 %	27 %

For a sarcastic tweet t_i with contextual information. Based on Table 4, we can know that the tweet t_i can be predicted to sarcastic with 71 % probability if this tweet has history-based context. The tweet t_i will be predicted as sarcastic with 100 % probability if this tweet has conversation-based context. The tweet t_i will be predicted as sarcastic with 27 % probability if this tweet has topic-based context. Based on these analysis, this can explain the reason that topic-based context can not improve the detection performance. Meanwhile, we can know that the conversation-based context is the best effective to detect sarcasm in twitter. In our previous experiment, the reason that the performance from history-based context is better than the conversation-based context dues to because the number of sarcastic tweets having history-based context is 498 in all 500 tweets, but the number of sarcastic tweets having conversation-based context is only 172 in all 500 tweets.

5.4 Parameter Sensibility

In Tables 2 and 3, the history-based context can improve the detection performance. With the increment of l, the performance will be improved, so we need find the best l for this type of context, based on this type of context, we experiment the performance of the proposed approach about l from 1 to 20. The result shows in Fig. 2, we can know that our model can get the best performance when l is set to about 5.

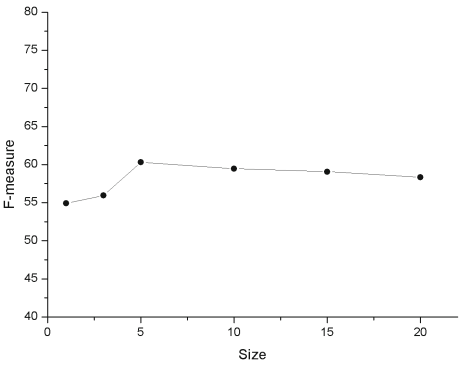

Fig. 2. The performance of the model on different context sizes

5.5 Experimental Results About Sentiment Analysis in Twitter

In our context-based model, the history-based context can get the best performance for sarcasm detection in twitter. For sentiment analysis in twitter, previous work [6] has not been developed the model to exploit the history-based context. In this section, we discuss the impact of the proposed model to the performance of sentiment analysis in twitter.

Table 5. Evaluation results of sentiment analysis in twitter

Methods	Negative			Positive			Macro-F(%)
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	
multi-class	67.34	69.20	68.28	68.68	66.40	67.33	67.70
history-1	68.68	72.60	70.34	70.57	64.40	67.00	68.67
history-3	68.77	73.20	70.28	70.84	64.60	67.45	**69.13**
history-5	69.99	69.60	69.56	68.88	67.40	67.90	68.73

We delete all tweets in basic dataset which its category label is sarcastic, meanwhile, we delete all contextual tweets for sarcastic tweets. All other settings are same to sarcasm detection. There is a two classification problem (negative and positive) of sentiment analysis in twitter. Based on feature modeling of word cluster, experimental results are showed in Table 5. The *multi-class* approach (not employing any context) can get 67.70 % in Macro-F. The SVM^{hmm} can get the better performance (69.13 %) than *multi-class* approach. Results show that the history-based context can improve the performance of sentiment analysis in twitter.

6 Conclusion

In this paper, the role of contextual information in sarcasm detection over Twitter is investigated. We modeled the sarcasm detection problem as a sequential classification task over target tweet and its context. A Markovian formulation of the Support Vector Machine discriminative model as embodied by the SVM^{hmm} algorithm has been employed to assign the category label to entire sequence. Results show that sequential classification effectively embodied evidence about the contextual information and is able to reach a relative increment in detection performance. It is worth noting that our proposed approach does not require manually coded resources.

References

1. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 415–463. Springer, New York (2012)

2. Tsytsarau, M., Palpanas, T.: Survey on mining subjective data on the Web. Data Min. Knowl. Discov. **24**(3), 478–514 (2012)
3. Davidov, D., Tsur, O., Rappoport, A.: Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In: Processdings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL 2010), Uppsala, Sweden, pp. 107–116 (2010)
4. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retireval **2**(1–2), 1–135 (2008)
5. Altun, Y., Tsochantaridis, I., Hofmann, T.: Hidden Markov support vector machines. In: Processdings of the International Conference on Machine Learning (ICML 2003), Washington, pp. 3–10 (2003)
6. Vanzo, A., Crose, D., Basili, R.: A context-based model for sentiment analysis in Twitter. In: Processdings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING 2014), Dublin, Ireland, pp. 2345–2354 (2014)
7. Gonzalez-Ibanez, R., Muresan, S., Wacholder, N.: Identifying sarcasm in Twitter: a closer look. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011), Portland, Oregon, pp. 581–586 (2011)
8. Filatova, E.: Irony and Sarcasm: corpus generation and analysis using crowdsourcing. In: Language Resources and Evaluation, pp. 392–398 (2012)
9. Burfoot, C., Baldwin, T.: Automatic satire detection: are you having a laugh? In: Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009), Singapore, pp. 161–164 (2009)
10. Tepperman, J., Traum, D., Narayanan, S.: "Yeah Right": Sarcasm recognitio for spoken dialogue Systems. In: Proceedings of the 9th International Conference on Spoken Language Processing, Antwerp, Belgium (2006)
11. Tsur, O., Davidov, D., Rappoport, A.: Semi-supervised recognition of sarcastic sentences in online product reviews. In: AAAI Conference on Weblogs and Social Media, Atlanta, Georgia, pp. 107–116 (2010)
12. Carvalho, P., Sarmento, L., Silva, M.J., de Oliveira, E.: Clues for detecting irony in user-generated contents: oh...!! it's so easy; -). In: Proceedings of the 1st CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, New York, NY, pp. 53–56 (2009)
13. Veale, T., Hao, Y.: Detecting ironic intent in creative comparisons. In: Proceedings of the 19th Europen Conference on Artificial Intelligence (ECAI 2010), The Netherlands, Amsterdam, pp. 765–770 (2010)
14. Reyes, A., Rosso, P., Veale, T.: A multidimensional approach for detecting irony in Twitter. Lang. Resour. Eval. **47**(1), 239–368 (2012)
15. Riloff, E., Qadir, A., Surve, P., Silva, L.D., Gilbert, N., Huang, R.: Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), Seattle, USA, pp. 704–714 (2013)
16. Wallace, B.C., Choe, D.K., Kertz, L., Charniak, E.: Humans require context to infer ironic intent (so computers probably do, too). In: Processdings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Baltimore, Maryland, USA, pp. 512–516 (2014)

17. Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., Wilson, T.: Semeval-2013 task 2: sentiment analysis in Twitter. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation(SemEval 2013), Atlanta, Georgia, USA, vol. 2, pp. 312–330 (2013)
18. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Preceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), Philadelphia, pp. 79–86 (2002)
19. Owoputi, O., O'Conor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N.A.: Improving part-of-speech tagging for online conversational text with word clusters. In: The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013), Atlanta, pp. 380–390 (2013)
20. Brown, P.F., deSouza, P.V., Mercer, R.L., Della Pietra, V.J., Lai, J.C.: Class-based n-gram models of natural language. Comput. Linguist. **18**(4), 467–479 (1992)
21. Joachims, T., Finley, T., Chun-Nam, Y.: Cutting-plane training of structural SVMs. Mach. Learn. **77**(1), 27–59 (2009)
22. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-vector machines. J. Mach. Learn. Res. **2**, 265–292 (2001)
23. Barbieri, F., Saggion, H.: Modelling irony in Twitter. In: Proceedings of the Student Research Workshop at the 14th Conference of the Europen Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, pp. 56–64 (2014)