The 2016 Conference on Computational Linguistics and Speech Processing
ROCLING 2016, pp. 299-310

Sarcasm Detection in Chinese Using a Crowdsourced Corpus

林士凱 Shih-Kai Lin
國立臺灣大學語言學研究所
Graduate Institute of Linguistics
National Taiwan University
serenity9078@gmail.com

謝舒凱 Shu-Kai Hsieh
國立臺灣大學語言學研究所
Graduate Institute of Linguistics
National Taiwan University
shukai@gmail.com

Abstract

Based on the assumption that comment with positive sentimental polarity to a negative issue has high probability to be a sarcasm, we propose a simple yet efficient method to collect sarcastic textual data by crowdsourcing with social media and merging *game with a purpose* approach. Taking advantage of Facebook's reaction button, posts triggering strong negative emotion are collected. Next, by using PTT's search engine, we successfully connect PTT's comments to the collected posts in Facebook and build the sarcasm corpus. Based on the corpus data, the performance comparison of sarcasm detection between SVM with naïve features and Convolutional Neural Network models is conducted. An impressive accuracy rate and great potentials of the corpus are demonstrated.

Keywords: sarcasm, PTT, convolutional neural network, support vector machine, crowdsourcing.

1. Introduction

Sentiment analysis is important in automatic interpreting large number of feedbacks from the internet society. However, the usage of sarcasm which typically conveys a negative opinion using positive words could flip the polarity of a message thus interfere the accuracy of the sentiment analysis (Maynard et al. 2014). Therefore, to improve the performance of sentiment analysis model, detection of sarcasm is definitely necessary (Bo et al. 2008).

Linguistically, sarcasm has been regarded as a complicated speech act which utters the opposite of what it literally means, and it distinguishes itself with irony in its intention of making the target the butt of derisive contempt (Ling et al. 2016). Sarcasm can be grammaticalized and lexicalized in various patterns, and often requires context-dependent readings with human involvement. Therefore, the construction of sarcasm corpus providing wider windows as well as training data for predictive model has long been considered as an uneasy task.

However, with the rapid growth of social media platform like Twitter and Facebook, a new solution is provided via crowdsourcing. For instance, a popular method in previous studies, some groups use Twitter's hashtag service to collect tweets with #sarcastic tag and build sarcasm corpus (González-Ibáñez et al. 2011, Reyes et al. 2012, Liebrecht et al. 2013).

Based on the assumption that comment with positive polarity to a negative issue has high probability to be sarcasm, we propose an automatic method to collect sarcastic text data by a two-step algorithm which first takes advantage of Facebook's reaction button then connects to the comments in Gossiping forum of PTT.

Due to the recent progress in machine learning and deep-learning technique, these two method could both handle sarcasm detection as a binary (sarcastic and non-sarcastic utterance) classification problem. However, a performance comparison of sarcasm detection between these two methods has not been conducted before. In this paper, we choose machine learning support vector machine (SVM) and deep-learning convolutional neural network (CNN) to test the difference. Both of them are widely adopted in natural language processing problems (Joachims et al. 1998, Collobert et al. 2008, Kim et al. 2014).

The rest of the paper is structured as follows. Section 2 describes related works on the construction of sarcasm and irony corpus, in Section 3, we describe the procedure of building sarcasm corpus and experimental settings. Results and limitations are discussed in Section 4, and finally, Section 5 draws the conclusion.

2. Related Work

Recently, there have been a great amount of studies in the field of NLP focusing on non-literal semantics such as sarcasm/irony detection. Most of the works exploited various

linguistic features and assembled different (semi-) supervised machine learning models in the task. In the view of language resources for sarcastic expressions, (Filatova et al. 2014) proposes a method in generating a corpus with sarcastic text utterances from Amazon product reviews using MTurkers; (Tang and Chen, 2014) adopt a more rhetoric-linguistic approach in mining ironic patterns and bootstrapping an open irony annotated corpus from microblog in Chinese. (Oraby et al. 2016) use lexico-syntactic cues with crowdsourced annotation to reliably retrieve sarcastic utterances in Dialogue.

Considering the importance of sustainability and reproductivity of research, in this paper, we aim to propose a non-paid social crowdsourced and naturalistic method for acquiring corpus data with event and affect annotations.

3. Experiment Setup

3.1 Corpus Data

According to the previous research, the miscellaneous pattern of sarcasm makes it's hard to write down the operational definition, and causes the difficulty in automatic collection from large text data. Therefore, instead of analyzing the lexical structure, we detect sarcastic text with the assumption that positive comment to negative issue has large possibility to be sarcasm (Riloff et al. 2013).

To find content that strongly triggers people's negative emotion, we take advantage of Facebook reaction button. Released on 2016/02/26 in Taiwan, users on Facebook could press five kinds of emotion button including ANGRY, SAD, WOW, HAHA, and LOVE to express their attitude toward a post in addition to the original LIKE button. We crawl the reaction data of Apple Daily's Facebook fan page from March to July, and pick out posts that ANGRY has the highest accumulation among every emotion and value larger than 1,000 in each month as the negative content.

In order to gain more naturalistic sarcasm data, we develop an online game called "酸檸檬 (*suan nín méng*)". Using negative posts from Apple Daily's Facebook's fan page as topic, players are told to type sentence that they think has the lowest pH value. The higher sarcastic level, the lower pH value, and it will accumulate after each round. Once the accumulation exceeds 15, the game is over. Such rule could encourage players to contribute sentence with

high sarcastic level for longer survival.

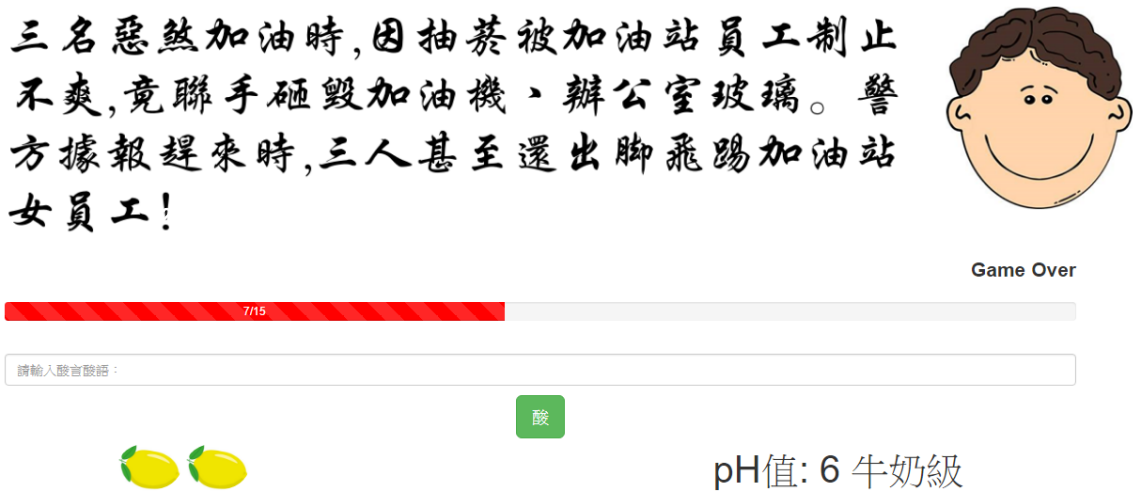

Figure 1. The real game scene of "酸檸檬 (*suan níng méng*)". Players are told to type sarcastic comment to the content above. The pH of each comment will be calculated according to the sentimental polarity analysis.

There are about 400 participants joining the game. According to the design of this game, at least 2~3 sentences should be collected from a single player. However, there's only 300 text data which is much less than expectation and inadequate for machine training. By interviewing with some players, we find that many people decide to close the game after logging because they feel too much effort is needed to come up with a sarcastic sentence.

Owing to the inefficiency of the current game framework, we then alternatively turn to the combination of crowdsourcing approach and social mining. Based on the famous culture of frequent usage of sarcasm and highly active discussion about current event (吳承樺, 2014), Gossiping forum of PTT should be the place second to none for building sarcasm corpus. The official released search engine of PTT is used to check whether the negative post from Apple Daily's Facebook fan page is shared in the Gossiping forum or not. If yes, all the comments will be collected and labeled sarcastic or non-sarcastic according to the polarity analysis. The whole procedure is shown in Figure 2.

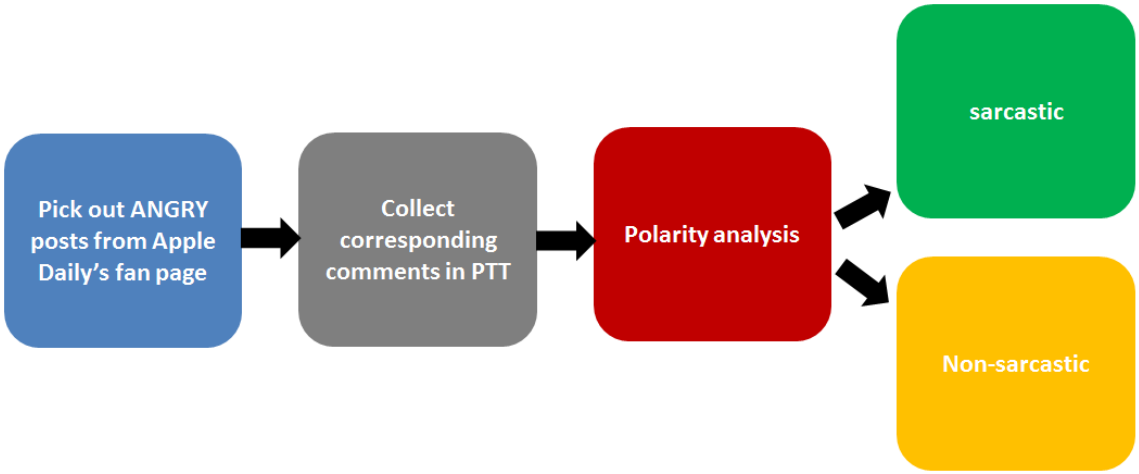

Figure 2. Illustration of the procedure of building a sarcasm corpus.

Lexicon-based approach is adopted for the polarity analysis, which depends on sentimental words appearing in a sentence to determine the polarity. The sentimental word list is built based on the TC-LIWC (黃金蘭 et al. 2012, 林瑋芳 et al. 2014) dictionary file, and two groups are included in the list, positive and negative. Words with posemo/negemo label are categorized into the positive/negative group. In addition, according our observation on PTT's comment, generally used curse words are also included in the negative group.

Simple sum up algorithm then be adopted to examine the polarity of each comment. The polarity score of a comment will add 1 once a word belonging to positive group appears in the sentence, and vice versa. Note that negation and degree terms are also considered for the polarity flipping and strengthening. Comments with polarity >= 0 are labeled as sarcastic, while polarity < 0 are labeled as non-sarcastic.

We observe that comments with negative polarity are mainly composed of curse words because the posts are all related to the extremely ANGRY issues. On the other hand, comments with polarity >= 0 indeed detect lots of sarcasm. However, in addition to the sarcastic comments, some non-sarcastic comments are also included in this category which generally focus on expressing opinion toward the issue rather than be sarcastic or irony to it. According to our observation, it's often to see keywords of the posts be mentioned in such type of comment.

To eliminate these biases, we calculate the term frequency–inverse document frequency

(TF-IDF) of each post, word with value larger than 0.1 as the keyword. If a comment contains any of the keywords, it will be filtered out. There are total 9,373 non-sarcastic and 17,256 sarcastic comments are collected.

3.2 Model Selection

3.2.1 Supporting Vector Machine

For SVM, determination of features mainly depends on human's observation, which is a highly empirical experience (Taira et al. 1999). However, the advantage is that features included in the model training could clearly attribute the importance to the classification result.

The conduction of SVM calculation is based on Python library scikit-learn (Pedregosa et al. 2011). According to the previous study (Mathieu, 2014), n-gram is a very effective feature for sarcasm detection, thus we choose to use bigram, trigram and tetragram of sarcastic comments as the feature for SVM model training. We only keep n-gram whose term frequency is higher than 3, and the total number of feature is 26,751. All the comments are encoded into a binary sparse matrix. An element of the matrix will be assign as 1 when the corresponding feature is included, and 0 vice versa. Linear kernel is used. The parameter C and gamma are both set to the default value.

3.2.2 Convolutional Neural Network

Due to the achievement of good text classification performance and the similarity of using short sentence data (Kim et al. 2014), CNN is selected as the representation of deep learning model for the sarcasm detection task. Figure 3 shows the structure of CNN used in our experiment.

For CNN, features are automatically extracted from the corpus through the filter, pooling algorithm and the complex neural network structure. Although deep-learning model could include features more thoroughly, one could not trace back the actual contribution from each feature.

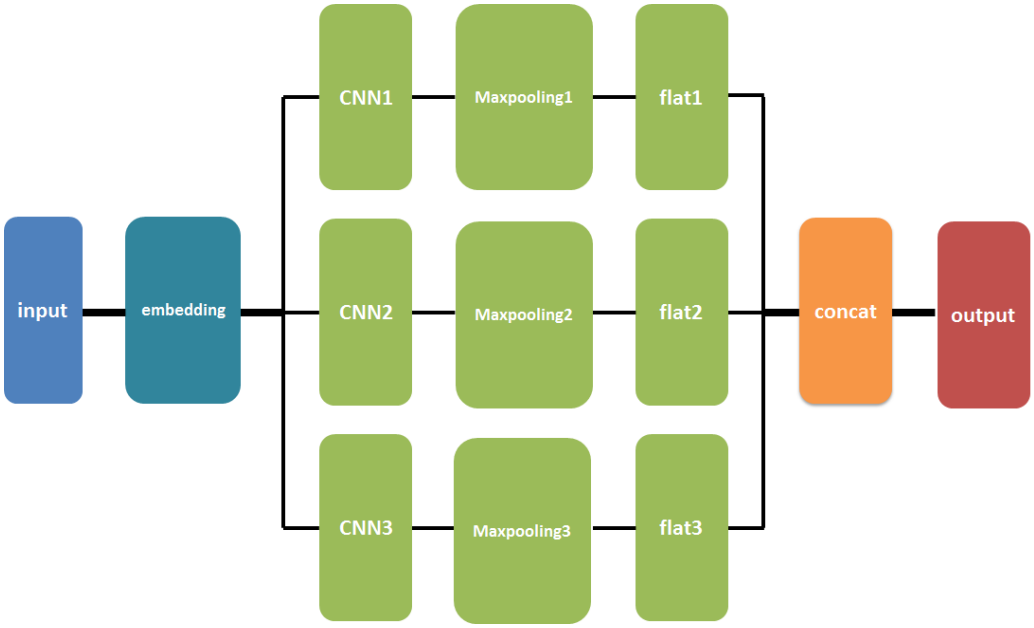

Figure 3. Illustration of the CNN structure. Input data will project into virtual space via word2vec in embedding layer. Three convolution layers with filter window size 2, 5, and 10 are used to extract features from the data.

Because the comments are collected from the Gossiping forum of PTT, we use Python Chinese Word Segmentation library JSEG developed by our lab which include PTT corpus to tokenize our data. Only top 20,000 frequently used tokens are remained as our input. The max length of a comment is defines as 20 words. If the length of a comment is less than 20, zero padding is adopted.

Python library Keras is used to do the CNN calculation (P.W.D. Charles et al. 2013). The embedding layer will conduct word2vec transformation projecting the input into a virtual space with 100 dimensions, and the basis of this virtual space is uninterpretable. Three different filters are used and all with number 200. These filters will slide through the virtual space created by embedding layer with stride size of 1 and extract fragments of the matrix. Rectified linear unit (ReLU) is employed as the activation function. All these fragments will go through a successive max pooling algorithm to generate lots of features.

From Figure 3, we can see that features from the three different CNN layer will concatenate together and feed into hidden layer with 50 neurons. Dropout rate 0.5 is used. Softmax, cross entropy and Adam are used as the activation function, loss function and optimization algorithm of the output layer.

4. Discussion

Both of the models are trained with a balanced data set, composing 9,373 non-sarcastic and 9,373 sarcastic data randomly sampled from the 17,256 data. Here we use Gaussian Naïve Bayes model as the baseline result to compare with. The same n-gram feature for SVM and default parameter setting are adopted. Although the n-gram has included as features for both Naïve Bayes and SVM, the average accuracy of 5-fold cross validation only reaches 57.9% and 55.4% respectively, which is just slightly higher than a random guess.

From the bigram, trigram and tetragram of the comments with polarity >= 0, we observe some specially used word appearing with high frequency. Some of them are topic-oriented, like "三寶 (*san bǎo*)" usually relates to car accident. The others are globally showed under different topic, like "不意外 (*bú yì wài*)". Users on PTT are used to using words like these to make comment sarcastic.

Table 1. Average accuracy of 5-fold cross validation

model	average accuracy
Naïve Bayes	57.9%
SVM	55.4%
CNN	87.1%

In contrast, CNN gets impressive 87.1% accuracy of 5-fold cross validation without human involvement in the feature engineering. The 1-D convolution layer collects features by filters with different size sliding through the semantic space, and the successive max pooling algorithm. It's unable to clearly interpret the meaning of the feature get from neural network, however, the result shows that such algorithm seems to include the sarcastic pattern more precisely than the n-gram feature in this pilot study.

However, it is noted that in the current study, it is not our intention to employ/discover/evaluate the most reliable linguistic features that signal the presence of sarcastic utterance in Chinese, such as those identified in English and other languages: emoticons and onomatopoeic expressions for laughter; heavy punctuation marks; quotation marks; positive interjections, or pragmatic features like smiley and frown that have been used

as discriminating features in the classification tasks. Gaining insights from linguistics, psychology and cognitive science, we argue that since there is no common agreement on the operational definition of sarcasm and related linguistic phenomena, any one-size-fits-all methodological attempt will run the risk of overfitting and over-generation.

The real challenges of sarcasm detection in texts involves not only linguistic knowledge represented in lexical, semantic-pragmatic, discourse levels, but also common-sense knowledge which is contextualized, situation-anchored and highly individuated. That is, most cases of sarcastic text utterances can only be understood when an individual/a social group placed within a broader context in responding to a certain situation. It is thus more urgent at this stage to build language resources for the exploration of influential factors and social ontologies for situated machine learning models on this task.

5. Conclusion

In this paper, based on the assumption that comment with positive polarity to a negative issue has high probability to be sarcastic, we propose an automatic method to build a sarcasm corpus that is advantageous of its situation-driven architecture and potentials for real-time processing. Start from the concept of crowdsourcing, we first make use of Facebook's reaction button to collect posts related to negative issue, finding comments to these posts from PTT, and finally label these comments sarcastic or not based on the sentimental polarity analysis.

Using the comments as training data, we compare the sarcasm detection performance of machine learning SVM and deep-learning CNN. The result shows that the difference in the feature engineering has great impact on the classification accuracy. Both trained by balanced model, CNN model could reach about 87% accuracy, which is far better than the 55% accuracy got from SVM. Although previous studies show that n-gram features have great importance in sarcasm detection, the automatic feature extraction from neural network seems to have more information in distinguishing a comment is sarcastic or not.

In summary, we propose a social crowdsourcing-based sarcasm corpus generation procedure which could efficiently collect sarcastic comments from PTT together with their original situations, which can be used for a closer look at the nature of sarcastic expressions, and the training data for different machine learning models as well. A preliminary experimental result

shows that deep-learning CNN has much stronger ability in detecting sarcasm than SVM.

We are planning to improve our online game "酸檸檬 (*suan níng méng*)" from tedious typing to providing dropping menu for selecting the most sarcastic comment collected from PTT. The complete pipeline from Facebook fan page negative posts identification to PTT comments collection and polarity analysis is ongoing. Players no longer need to figure out sarcastic comments by themselves, rather they just need to select out the most sarcastic PTT comments toward to a specific issue. We believe such improvement could largely decrease the effort to play the game, and could enhance the intention to contribute annotation data.

By making use of such data, we could further filter out the biases in the sarcastic corpus, and develop the original sarcasm classification into sarcastic level regression problems which will facilitate and shed new light on a more realistic and individuated sarcastic computing.

.

Reference

[1] Maynard, Diana, and Mark A. Greenwood. "Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis." LREC. 2014.

[2] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis."Foundations and trends in information retrieval vol. 2.1-2, pp. 1-135, 2008.

[3] Ling, Jennifer, and Roman Klinger. "An Empirical, Quantitative Analysis of the Differences between Sarcasm and Irony." Semantic Sentiment and Emotion Workshop, ESWC, Crete. Greece. 2016.

[4] González-Ibánez, Roberto, Smaranda Muresan, and Nina Wacholder. "Identifying sarcasm in Twitter: a closer look." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2. Association for Computational Linguistics, 2011.

[5] Reyes, Antonio, Paolo Rosso, and Davide Buscaldi. "From humor recognition to irony detection: The figurative language of social media." Data & Knowledge Engineering, vol. 74, pp. 1-12, 2012.

[6] Liebrecht, C. C., F. A. Kunneman, and A. P. J. van den Bosch. "The perfect solution for detecting sarcasm in tweets# not." Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2013, pp. 29-37.

[7] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." European conference on machine learning. Springer Berlin Heidelberg, vol. 1398, pp. 137-142, 1998.

[8] Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." Proceedings of the 25th international conference on Machine learning, 2008, pp. 160-167.

[9] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 2014.

[10] Filatova, Elena. "Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing." LREC. 2012.

[11] Tang, Yi-jie, and Hsin-Hsi Chen. "Chinese Irony Corpus Construction and Ironic Structure Analysis." COLING, 2014, pp. 1269-1278.

[12] Oraby, S., Harrison, V., Hernandez, E., Reed, L., Riloff, E., and Walker, M. “Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue.” Proceedings of the 17th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL), 2016.

[13] Riloff, Ellen, et al. "Sarcasm as Contrast between a Positive Sentiment and Negative Situation." EMNLP, vol. 13, pp 704-714, 2013.

[14] 吳承樺. "網路匿名 酸民文化" .http://hdl.handle.net/11536/37301. 2014.

[15] 黃金蘭、Chung, C. K.、Hui, N.、林以正、謝亦泰、程威詮、Lam, B.、Bond. M.，及 Pennebaker, J. W. 中文版語文探索與字詞計算字典之建立。中華心理學刊，vol. 54, pp 185-201, 2012.

[16] 林瑋芳、黃金蘭、林以正. 從 LIWC 到 C-LIWC：電腦化中文字詞分析的潛力。台灣諮商心理學報，vol. 1, pp 97-111, 2014.

[17] Taira, Hirotoshi, and Masahiko Haruno. "Feature selection in SVM text categorization." AAAI/IAAI. 1999.

[18] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[19] Mathieu , The sarcasm detector, http://www.thesarcasmdetector.com, 2014

[20] P.W.D. Charles, Project Title, GitHub repository, https://github.com/charlespwd/project-title, 2013