A Study in Practical Solutions to Sarcasm Detection with Machine Learning and Knowledge Engineering Techniques

Chia Zheng Lin*, Michal Ptaszynski*, Masui Fumito*, Gniewosz Leliwa, Michal Wroczynski****
*Graduate School of Computer Science, Kitami Institute of Technology, Japan
**Samurai Labs, Poland
{chiazhenglin}@gmail.com, {ptaszynski,f-masui}@cs.kitami-it.ac.jp, {gniewosz.leliwa, michal.wroczynski}@samurailabs.ai

Abstract

In this paper we tackle the problem of sarcasm detection with the use of machine learning and knowledge engineering techniques. Sarcasm detection is considered a complex and challenging task in Natural Language Processing and has been studied by various researchers in the past decade. To get a grasp on the present state of the art in sarcasm detection, we review the important previous research in this field, with a focus on text-based sarcasm detection in English texts. In the proposed method, we compare various dataset preprocessing techniques on the proposed Deep Convolutional Neural Network model. As a result, the most specific, or least preprocessed dataset ranked as the one with the highest performance. However, we observed that some level of data preprocessing could become useful in the task of sarcasm detection.

Introduction

Sarcasm, often used together or interchangeably with irony, is considered an important component of human communication recognized as some of the most prominent and pervasive devices of figurative and creative language widely used from dating back to ancient religious texts to modern times (Ghosh and Veale 2017).

Van Hee (2017) suggested the important implications of irony and sarcasm for Natural Language Processing (NLP) tasks, which aim to explain construct of human language, and the large potential in the domain of text mining. In the recent years, there has been an increasing interest in, especially, automatic sarcasm detection and classification, which have been widely studied as a type of sentiment analysis task (detecting whether a sentence conveys a positive or negative connotation, or in this case: sarcastic or non-sarcastic). Especially, Kumar et al. (2017) surveyed some representative work in the related area and categorized most of the popular approaches into three types, namely, rule-based, statistical, and deep learning-based approaches. We analyse some of that research in the next section.

Researchers' interest in analysing this profound type of figurative and creative use of language grew along with the dramatic increase in the everyday use of social media over the past decade. Especially, Twitter has become one of the most popular venues for people to express their opinions, share their thoughts and report real-time events, etc. Moreover, the huge amount of data has drawn interest of companies for the purpose of studying the opinion of people towards different products, facilities and events. It has been suggested that the nature of tweets makes them the most suitable for studying sarcasm detection approaches (Bouazizi and Otsuki 2016).

However, the lack of empirical investigations into optimal approaches for sarcasm detection is a serious oversight in many related studies carried out throughout the years. Importantly, there have been no studies comparing the differences in the preprocessing and manipulation of the dataset to improve the results of detection.

To contribute to dealing with the above-mentioned problems, in this paper we investigate the variations in sarcasm detection results caused by differences in applied preprocessing techniques typically used in NLP research but not applied before in works focusing on sarcasm detection. To do that most effectively, we firstly review previous related research on text-based sarcasm detection from English tweets, describe the implemented dataset preprocessing techniques, and discuss the results of an experiment performed to compare preprocessing techniques implemented on the dataset. As a result, we managed to observe the impact contributed by hashtags and labels related to sarcasm.

Finally, Ptaszynski et al. (2010) in their research on developing an expert system for Internet Patrol pointed out that, especially with regard to the increased popularity of SNS, sarcasm has been often used in personal attacks, such as cyberbullying and concluded that sarcasm detection is one of the important problems in cyberbullying detection. Therefore, as one of the practical applications, in this research we will verify how effective is sarcasm detection in the detection of cyberbullying.

Research Background

The word sarcasm originates from an Ancient Greek word *sarkasmós* and means "to tear flesh, bite the lip in rage, sneer." According to Oxford dictionary (2019), sarcasm is a

way of using words that are the opposite of what one means in order to be unpleasant to somebody or to make fun of them. They also described irony to be the use of words that say the opposite of what you really mean, often as a joke.

The relationship between irony and sarcasm has been confused in many studies. In the literature, two types of irony are widely considered: verbal irony and situational irony. While situational irony involves an incongruence between two situations, verbal irony, although applying verbal, or semantic incongruence, is a statement in which the meaning that a speaker employs is sharply different from the meaning that is ostensibly expressed. Hence, verbal irony is considered different from situational irony in that it is produced intentionally by the speakers.

When it comes to sarcasm, Van Hee (2017) defines it to be a form of verbal irony with an aggressive tone, is directed at someone or something, and is used intentionally. Hence the term "irony" and "sarcasm" are used interchangeably in many related studies. In this study, we decided to not focus on distinguishing between sarcasm and irony, and instead implement the general term "sarcasm" throughout the paper.

Previous Research

Tepperman (2006)'s spoken dialogue system used feature extraction approach for sarcasm detection as a subtask in their system, by which they introduced sarcasm detection into the scene of Nature Language Processing. One study by Davidov (2010) utilized tweets and Amazon reviews for text-based sarcasm detection, and Tsur (2010) proposed one of the first attempts to use feature engineering and statistical classifiers to detect sarcasm.

A number of studies have sought to detail the recent trend in sarcasm detection approaches, which can roughly be classified into three parts: rule-based, statistical, and deep-learning approaches (Kumar, Somani, and Bhattacharyya 2017; Barbieri 2017). Rule-based approaches attempt to identify irony through specific evidence which could be captured in terms of rules that rely on indicators of sarcasm. Barberi (2017) argued that rule-based approaches which require no training mostly rely on lexical information and do not perform as well as statistical approaches. Riloff (2013) aimed to recognize positive words in negative sentences while presenting a bootstrapping algorithm that automatically learns the rules from certain situations.

Most of the early works on sarcasm detection applied statistical approaches which varied in terms of features and learning algorithms, basically composed of two phases where data were converted into feature vectors before being classified using machine learning algorithm. Some of the most often used algorithms include Support Vector Machines (SVM), and Naïve Bayes. One of the first attempts in this approach by Tsur (2010) compiled a set of sarcastic patterns composed of common combinations of words extracted from sarcastic examples. Gonzalez-Ibanez (2011) composed a model with three pragmatic features which were positive emoticons, negative emoticons, and users' tagging. Reyes (2013) proposed another model based on four features, signatures, unexpectedness, style and polarity, and emotional scenarios.

Deep Learning approaches have been successfully brought into the scene of sarcasm detection when Amir (2016) used a standard binary classification with Convolutional Neural Network (CNN) while Poria (2016) implemented a combination of CNNs trained on different tasks. Popular Deep Learning algorithms include CNN (LeCun et al. 1998) and Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997). Ghosh and Veale (2017) proposed a network model composed of CNN followed by an LSTM network which outperformed many other models at that time. They utilized CNN to reduce frequency variation through convolutional filters and extract discriminating word sequences as a composite feature map for the LSTM layer. Then the output of the LSTM layer was passed to a fully connected Deep Neural Network (DNN) layer, producing a higher order feature set based on the LSTM output.

Following the Semantic Evaluation 2018 international workshop Task 3: Irony Detection in English Tweets (2018) which received submissions from 43 teams worldwide for the binary classification task A, deep learning algorithms were further explored and optimized for irony detection tasks. The best ranked system submitted by team THU_NGN (2018) consisted of densely connected LSTM network with multi-task learning strategy. Another system from one of the top teams, NTUA-SLP (2018), which used an ensemble of two bi-directional LSTM network-based models, achieved comparable results. The submissions represented a variety of neural network-based approaches and other popular classification algorithms including SVM, Random Forest, and Naïve Bayes (Van Hee, Lefever, and Hoste 2018). Overall, the approaches with ensemble learners were the current trend to tackle the challenges in sarcasm detection.

Proposed Method

Dataset Preprocessing

In the majority of recent studies applying machine learning methods to text classification, the datasets are usually used in their most basic form, namely, represented as tokens (words, punctuation, etc.), despite a wide variety of knowledge-based NLP systems (e.g., stemmers, part-of-speech taggers, etc.) capable of initial preprocessing of datasets, thus providing more informative features to ML algorithms. Therefore in this research we performed additional preprocessing to the dataset to verify usefulness of such knowledge-base systems in ML.

For the implemented dataset, each tweet was first transformed into lowercase and emojis were represented with their corresponding labels (e.g. :smileyface:) using Emoji for Python (2019). All tagged users (e.g. @user123) and URLs (e.g. http://google.com/) appearing in the text were replaced with specific neutral labels, such as "_tagged_" and "_url_." The first dataset preprocessing technique to be used in this study is shown below.

1. Only basic preprocessing.

To verify the depth of dependence of sarcasm detection on hashtags, all of the hashtags (e.g. #sarcasm) in the next 5 versions of the dataset shown below were replaced with a general label, e.g., "_hashtag_."

2. URLs, tagged users and hashtags replaced with labels.

Furthermore, we applied the knowledge-based tools for language processing provided by NLTK (2019).

3. Stemming of all words using Porter Stemmer (2019)
4. Stopwords removal with NLTK built-in Stopwords Filtering Tool
5. Stemming of all words after stopwords removal
6. PoS tagging using NLTK Universal Part-of-Speech Tagset

Finally we have our last dataset 7 to have its social media markers such as hashtags, URLs, and tagged users removed instead of being replaced with labels.

7. Tagged users, URLs, and hashtags removed

Below are three examples of a tweet, with hashtags (dataset 1), with hashtags replaced with labels (dataset2), and with hashtags removed (dataset7).

```
monday morning is my favorite! #sarcasm
monday morning is my favorite! _hashtag_
monday morning is my favorite!
```

Feature Weighting

Traditional weight calculation scheme was applied to all versions of the dataset. In particular, we used term frequency with inverse document frequency (tf*idf). Term frequency tf (t,d) refers here to the traditional raw frequency, which is the number of times a term t (word, token) occurs in a document d. Inverse document frequency idf (t,D) is the logarithm of the total number of documents D containing the term t. Finally tf*idf refers to the term frequency multiplied by inverse document frequency as in equation 1.

$$idf(t, D) = log\frac{|D|}{n_t} \quad (1)$$

Applied Classifier

Based on our previous work (Chia, Ptaszynski, and Masui 2019; Ptaszynski, Eronen, and Masui 2017), in this study we propose to use Convolutional Neural Networks (CNN) due to it having the best result for classifying tweets without ironic hashtags when compared to other classifiers.

CNN are a type of feed-forward artificial neural network which is an improved neural network model originally designed for image recognition. CNN performance has been proved useful in various classification tasks including sentence classification and NLP (Kim 2014; Ptaszynski, Eronen, and Masui 2017).

In the proposed CNN we applied Rectified Linear Units (ReLU) as neuron activation function which is a piece-wise linear function that will output the input directly if positive, zero if negative. We also applied dropout regularization. The CNN consisted of two hidden convolutional layers, containing 20 and 100 feature maps, respectively, with both layers having 5x5 patch size and 2x2 max-pooling.

Evaluation Experiment

Dataset Description

The dataset used in this research was the publicly available sarcasm detection dataset collected by Ghosh and Veale (2017) and consists of 51,189 tweets (24,453 sarcastic tweets and 26,736 non-sarcastic tweets) in which sarcastic tweets were automatically collected from Twitter using user's self-declaration of sarcasm/irony with sarcastic and ironic hashtags (e.g. #irony, #sarcasm) and annotated for confirmation. All seven dataset versions were implemented with different data preprocessing methods.

Experiment Setup

All seven separate versions of the dataset (represented with various preprocessing techniques) were analysed in the experiment using the proposed CNN method in the setting of a 10-fold cross validation procedure. The results were calculated using standard balanced F-score (F1) which is the harmonic mean of Precision and Recall.

Results and Discussion

Table 1 shows the summary of all results from the 7 datasets with different preprocessing techniques applied. Dataset 1 which is the dataset with all the hashtags included yielded an F1 score of 0.997. Compared to our previous work (Chia, Ptaszynski, and Masui 2019) which tested on a smaller data set with only 4,618 tweets and attained an F1 score of 0.844 with similar settings (hashtags included), this shows the significant increase in the performance of the CNN model with the increase of the size of the dataset. This suggests that the model is tied to the size of the implemented dataset and the number of extracted features.

The results of dataset 1 (hashtags included) also enhance our understanding of the impact of hashtags, which make a great difference in sarcasm and irony detection, especially in Twitter messages. However, due to the natural characteristics of deliberate sarcastic hashtags in Twitter, classification of tweets with hashtags included does not contribute much to the study of sarcasm detection from linguistic point of view. However, as the results show, hashtags can be a very useful practical mean to handle sarcasm detection with high performance.

While the remaining datasets were stripped of their hashtags (replaced with labels), data set 2 has no further preprocessing while data set 3 to 6 were further processed with different methods. Interestingly, data set 2 still attained the highest F1 score among all the data sets without hashtags included. This discovery highlights the importance of linguistic features in irony detection and shows that increment in data preprocessing does not always provide better results. This is due to the oversimplification of data with many vital and important features manipulated or removed while classification tasks such as irony detection heavily depended on them.

However, further preprocessed data sets have their own value despite attaining lower F1 score. From our observation on the attributes extracted from their confusion matrices in Table 1, their true positive rate is higher than the data set 2 which scored the highest F1 score among the datasets. Data set 5 which implemented both stemming and stop-word removal has obtained the highest true positive rate with only 290 false positive. This shows the implementation of further data preprocessing is crucial to the sensitivity of the data.

	Data set	True Positive	False Positive	False Negative	True Negative	F-score
1	With hashtags	24355	98	72	26664	0.997
2	Without hashtags	24055	398	5068	21668	0.898
3	Stemming applied	24013	440	5172	21564	0.895
4	Stopwords removed	24009	444	5183	21553	0.895
5	Stemming and Stopwords removed	24163	290	5590	21146	0.892
6	PoS Tagging applied	23904	549	5171	21565	0.893
7	Hashtags, URL, tagged users removed	16509	7944	8677	18059	0.665

Table 1: Results from seven datasets with different preprocessing.

Dataset 1	occ	Dataset 2	occ	Dataset 7	occ
#sarcasm	71	_hashtag_	5445	love	1656
sarcasm	60	_tagged_	1639	like	1216
tagged	51	love	413	not	1211
love	22	great	275	good	752
great	8	not	245	great	709
not	8	best	133	hate	488

Table 2: Top 6 error feature occurrences for dataset 1, 2 and 7 (occ = occurrence)

Finally for the last dataset 7 which had all of its social media markers, such as tagged users (e.g. @user123), URLs, and hashtags completely removed, Table 1 shows that the result dropped significantly to an F1 score of 0.665 comparing to other datasets. This case has shown the impact of the labels which were supposed to be neutral to the classification. Comparing to dataset 2 which had the social media markers replaced with labels, the significant increase in false negatives shows that the presence of the labels provides heavy contribution to the precision of the classification.

Error Analysis

Table 2 shows the occurrences of top 6 error features extracted from dataset 1 (with hashtags), dataset 2 (hashtags replaced with labels) and dataset 7 (hashtags, URLs, and tagged users removed) after removing prepositions, conjunctions, and pronouns which do not contribute much to the classifications. For dataset 1, the error feature which occurred the most is the #sarcasm following the word sarcasm. This shows that even the sarcastic hashtags cannot assist the model to achieve 100% sensitivity.

For the dataset 2 results in the second column, the label _hashtag_ appeared 5445 times out of the 5466 misclassified instances (99.62%). Coming up next is the label _tagged_ which appeared 1639 times while the remaining words such as “love”, “great”, “not”, and “best”, which are popular errors in all the 3 implemented datasets. As previously noticed, the supposedly neutral labels, in fact contribute heavily to the precision of the classification. Therefore, removing them does not provide improvement to the results.

The evidences so far provide further support for the hypothesis that deliberate sarcastic hashtags play a significant role in sarcasm detection in tweets. Taken together, these results also suggest that hashtag is the product of authors who understand that their sarcastic phrases alone may not be sufficient for the audience to figure out the intended irony or sarcasm. However, these findings do not completely solve the general sarcasm detection nor do they redefine sarcasm or irony in textual communication especially on social network service.

Application in Automatic Cyberbullying Detection

Although the number of research on sarcasm and irony detection grows each year, practical implementation of such models have not been widely discussed. Ptaszynski et al. (Ptaszynski et al. 2010) mentions, that sarcasm poses a problem in cyberbullying (CB) detection. Therefore, aiming to improve their expert system for automated Internet Patrol, we propose a practical implementation of sarcasm detection in cyberbullying detection.

To quantify the extent to how such model would be useful, we applied the model trained on sarcastic dataset 2 and tested on the cyberbullying detection dataset provided by Ptaszynski et al. (2018) which consists of 12,728 data samples. The result attained an F-score of 0.889 which is comparable to the result of dataset 2 with an F-score of 0.898 above. Interestingly, it was also much higher than models trained on purely cyberbullying-related data (Ptaszynski et al. 2018). This observation shows the prevalence of sarcasm in cyberbullying, and proves the practical applicability of sarcasm detection in other tasks.

Conclusion

In this paper, to find practical solutions for sarcasm detection on Twitter, we compared various dataset preprocessing methods and observed the impact of the preprocessed labels.

We firstly reviewed previous related works on text-based sarcasm detection, where we covered various types of systems, such as rule-based, statistical, or deep learning-based. Next, we compared datasets with various preprocessing on the proposed CNN model.

The first dataset with hashtags included scored an F1 of 0.9965, thus proving the dependence on hashtags in sarcasm detection. Next, the dataset with the least preprocessing ranked the highest among all datasets without hashtags included. However, we observed that data preprocessing is still crucial to the sensitivity of data Lastly, this research serves as a base for future studies on application of sarcasm detection in other tasks, such as cyberbullying detection.

In the future, we also plan to further improve the proposed method with more and diverse features and test it on larger datasets, also with other preprocessing techniques. We will also focus on optimizing the feature extraction and the classifier model.

References

Amir, I.; Wallace, B. C.; Lyu, H.; Carvalho, P.; and Silva, M. J. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics.

Barbieri, F. 2017. Machine learning methods for understanding social media communication: Modeling irony and emojis. Department DTIC.

Baziotis, C.; Athanasiou, N.; Papalampidi, P.; Kolovou, A.; Paraskevopoulos, G.; Ellinas, M.; and Potamianos, A. 2018. Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns. In *Proceedings of the 12th International Workshop on Semantic Evaluation(SemEval-2018)*. Association for Computational Linguistics.

Bird, S.; Loper, E.; and Klein, E. 2019. Natural language toolkit. https://www.nltk.org/.

Bouazizi, M., and Otsuki, T. 2016. A pattern-based approach for sarcasm detection on twitter. In *Digital Object*. IEEE Access.

Chia, Z. L.; Ptaszynski, M.; and Masui, F. 2019. Exploring machine learning techniques for irony detection. In *Proceedings of The 33rd Annual Conference of the Japanese Society for Artificial Intelligence (JSAI 2019)*. Japanese Society of Artificial Intelligence.

Davidov, D.; Tsur, O.; and Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the 14th Conference on Computational Natural Language Learning*. Association of Computational Linguistics.

Ghosh, A., and Veale, T. 2017. Fracking sarcasm using neural network. In *Proceedings of NAACL-HLT 2016*. Association for Computational Linguistics.

Gonzalez-Ibanez, R.; Muresan, S.; and Wacholder, N. 2011. Identifyinng sarcasm in twitterl: A closer look. In *Proceedings of the 49th Annual Meeting of the Association For Computational Linguistics*. Assosiation for Computational Linguistics.

Hee, C. V. 2017. Can machine sense irony? exploring automatic irony detection on social media. University Gent.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory.

Kim, T., and Wurster, K. 2019. Emoji for python. https://pypi.org/project/emoji/.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Kumar, L.; Somani, A.; and Bhattacharyya, P. 2017. Approaches for computational sarcasm detection: A survey. ACM CSUR.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. In *Proc of the IEEE*.

Oxford. 2019. Oxford learner's dictionary. https://www.oxfordlearnersdictionaries.com/.

Poria, S.; Cambria, E.; Hazarika, D.; and Vij, P. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. COLING 2016.

Porter, M. 2019. The porter stemming algorithm. https://tartarus.org/martin/PorterStemmer/.

Ptaszynski, M.; Dybala, P.; Matsuba, T.; Masui, F.; Rzepka, R.; Araki, K.; and Momouchi, Y. 2010. In the service of online order: Tackling cyber-bullying with machine learning and affect analysis. In *International Journal of Computational Linguistics Research*. Hokkaido University.

Ptaszynski, M.; Leliwa, G.; Piech, M.; and Smywinski-Pohl, A. 2018. Cyberbullying detection - technical report 2/2018, department of computer science agh, university of science and technology.

Ptaszynski, M.; Eronen, J. K. K.; and Masui, F. 2017. Learning deep on cyberbullying is always better than brute force. In *IJCAI 2017 3rd Workshop on Linguistic and Cognitive Approaches to Dialogue Agents (LaCATODA 2017)*.

Reyes, A.; Rosso, P.; and Veale, T. 2013. A multidimensional approach for detecting irony in twitter. Lang Resources and Evaluation.

Riloff, E.; Qadir, A.; Surve, P.; Silva, L. D.; Gilber, N.; and Huang, R. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceddings of the 2013 Conference on Empirical Methods in Natural Language Processing(EMNLP 2013)*. EMNLP.

Tepperman, J. 2006. Yeah right: Sarcasm recognition for spoken dialogue system. In *Interspeech 2006*. ICSLP.

Tsur, O.; Davidov, D.; and Rappoport, A. 2010. Icwsm – a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*. Association for the Advancement of Artificial Intelligence.

Van Hee, C.; Lefever, E.; and Hoste, V. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*. Association for Computational Linguistic.

Wu, C.; Wu, F.; Wu, S.; Liu, J.; Yuan, Z.; and Huang, Y. 2018. Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning. In *Proceedings of the 12th International Workshop on Semantic Evaluation(SemEval-2018)*. Association for Computational Linguistics.