Sarcasm Detection Using Emojis

Emily Roller, Alyosha Romanov and Jordyn Allen

Abstract

Text is a crucial part of modern social media platforms and communication. Written text however lacks the emotional cues of face-to-face dialogue and leads to many ambiguities. Many of which are worsened by the informal nature of many social media posts. In social media, sarcasm represents the nuanced form of language that individuals state the opposite of what is implied. The detection of this if this sarcasm however is crucial to understanding the meaning behind messages. The majority of sarcasm detection algorithms that exist today focus exclusively on text emotion. In real scenarios, however, emojis are widely used as signals to emotion, which could greatly aid in detecting sarcasm. Therefore in this paper, we present two versions of a dataset collected from Twitter for a 5-month period and from Facebook for two years from 2015 to 2017 using web scraping Python. The first version of this dataset features the text with no emojis while the second features this dataset with the emojis replaced with their word counterpart. We will be comparing the performance of sarcasm detection between a number baseline classifiers using NLP classification algorithms, including three dummy classifiers, Naive Boyes, and K-neighbors to a deep learning convolutional neural network model.

1 Introduction

In everyday communication, the impact of social media is hard to ignore. Although videos and other forms of media are still common in social media platforms like Twitter and Facebook, text still holds a dominating force. In text however it is commonplace to see Emojis, a visual expression of the message sent. Although communication through solely text may lack non-verbal cues, emojis can provide more expression to solve this issue. Emojis can help readers discern an author's true emotion since emotion is simply a picogram that expresses a real facial expression. One of the most challenging emotions to detect is Sarcasm. The meaning of which is when words are used to express something other than and especially the opposite of the literal meaning of a sentence. It is unfortunately not always easy to figure out if a writer is being sarcastic - particularly as we move ahead in a digital age that involves communication involving texting, emailing, and online comments through social media replacing phone calls or video chats. When delivered in person, sarcasm tends to assume a bitter-cutting tone. But written messages don't always get that attitude across or give enough context. Sarcasm thrives in ambiguous situations – and that's the main issue. Therefore sarcasm detection is an important task as it can significantly improve the quality of online communication by increasing understanding of people's online emotions.

The bulk of algorithms that exist on sarcasm detection focus on text information. These include identifying the characteristics of a user from their past response texts, activities, etc. Most of them have attempted to train deep neural network models using the text to analyze sarcasm. To improve the performance of all of these methods the usage of emojis could be considered to detect sarcasm. Since as we discussed above they help to find the tone of speech using picograms that mirror human facial expressions. For example comments such as, "Bro u r soo handsome 😂 🤢", "Love You 😒", and “Happy Monday morning😭 😩” are examples of sarcastic comments. Without the emoji, these comments could instead imply an emotional meaning different from sarcasm. Especially with the inclusion of words with positive connotations such as “handsome,”, “love” and “happy”. Therefore making the emojis here is essential to figuring out the user’s true intention of sarcasm.

In this paper, we address the problem of detecting sarcasm with the usage of Emojis throughout different social media platforms. To summarize our contributions, we are investigating: 1) how to learn the representation of emojis and text separately; 2) how to take advantage of the emoji signals to improve the performance of detecting sarcasm in text. To solve these challenges we will be comparing the performance of sarcasm detection between five baseline classifiers using NLP classification algorithms to a deep learning neural network model.

2 Related Work

2.1 Sarcasm Detection Using Emojis

Although sarcasm detection is a very narrow research in NLP, it is rapidly growing and there has been some notable works done on the subject. This paper is heavily inspired by Subramanian et al. 2019 [8], who thoroughly explored the effectiveness of creating a new framework ESD (Emoji-based Sarcasm Detection) which simultaneously captures various signals from text and emojis for sarcasm detection. We use their dataset as a baseline for our studies.

2.2 Sarcasm Sentiment Analysis

Sarcasm sentiment analysis is a rapidly growing area of NLP with research ranging from word, phrase and sentence level classification to document and concept level classification.[1] [2] [3] Research is developing in finding ways for efficient analysis of sentiments with better accuracy in written text as well as analyzing humor, irony and sarcasm within social media data. Sarcastic sentiment detection is classified into three categories based on text features used for classification, which are lexical, pragmatic and hyperbolic as shown in Figure 2

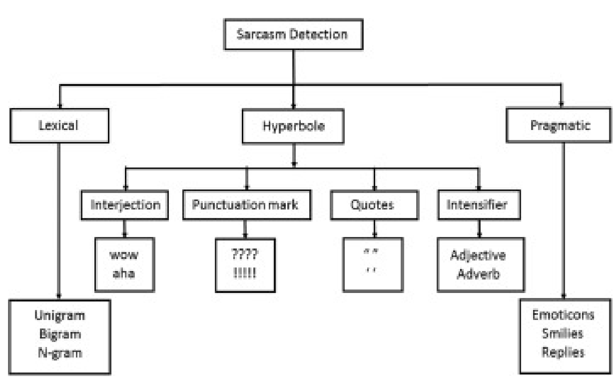

Figure 1: Classification of sarcasm detection based on text features used.

2.3 Emoji Analysis of Social Media

There has been developing research in regards to the study of emojis and how they relate to sentiment classification and text understanding. Hu et al. [4] incorporates emotion signals in an unsupervised framework for sentiment classification. Kelly et al. [6] shows that emojis can be used as appropriations which can help facilitate communications using interview data. Eisner et al.[7] shows using emoji along with word embedding from Google News can improve sentiment analysis. Hallsmar et al. [5] runs a study using an emoji training heuristic for multi-class sentiment analysis on Twitter with a Multinomial Naive Bayes Classifier

3 Approach

3.1 Dataset

The first step to our approach was to find an appropriate dataset for analysis. We searched through many pre-made Twitter datasets that included emojis, but many of these were too small, unannotated, or had other issues. The dataset that we chose in the end was from the work performed by Subramanian et al. [citation number], which consists of over 16,000 tweets including hashtags and emojis, each of which is labeled with a 1 or 0 to indicate sarcasm or lack of sarcasm, respectively. The dataset is publicly available and can be found at [github link].

3.2 Data Preprocessing

We began by importing the aforementioned tweet sarcasm database and its corresponding labels. We then removed links, twitter images, punctuation and converted hashtags to plain terms to create the first “basic” dataset. We then created two additional datasets, one where we transformed the emojis into plain text using the emoji library and removing the term “face” from the created text (the “text emoji” dataset), and one where emojis were removed outright (the “no emoji” dataset). We then tokenized and compared each token against an English dictionary removing any terms that were deemed not English, slang, or misspelled. We further removed any tweets that did not have at least 5 meaningful tokens without emojis. This reduced the database from an original size of 16000 to around 9000 tweets, now tokenized and more accurate for our goal.

3.3 Baseline Models

We implemented five different baseline classifiers: three dummy classifiers, a Naive Bayes classifier, and a k-Nearest-Neighbors classifier. The dummy classifiers, which were implemented using the SKLearn DummyClassifier classifier, each used a different strategy, either predicting the most frequent label, predicting uniformly at random, or predicting by respecting the training set’s class distribution. The Naive Bayes classifier was implemented using the SKLearn Naive Bayes classification model trained on ngram-level TF-IDF vectors, with an ngram range of n = 2 to n = 4. Finally, the k-Nearest-Neighbors was implemented with k = 5.

3.4 CNN Model

Our CNN model was built using the Keras library similar to Figure 2 as shown above. We used an input layer with shape (70,), an embedding layer using embedding and spatial dropout, a pooling layer, and two dense output layers with relu and sigmoid activations respectively. We also used the Adam library optimizer function and binary cross-entropy loss function. Additionally, the model makes use of the fastText pre-trained word-embedding vectors, which consists of 1 million word vectors trained on articles from Wikipedia 2017.

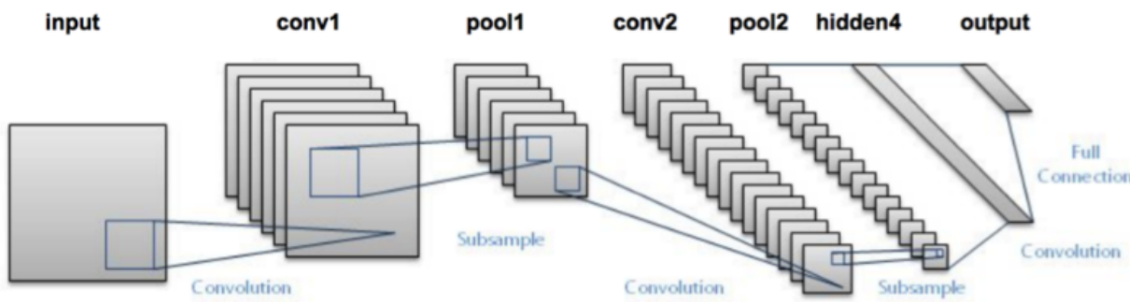

Figure 2: Classification of sarcasm detection based on text features used.

4 Experiment

For each model, we performed three experiments: in the first, we used the dataset preprocessed with the non-emoji sets (the "basic" dataset) for training/testing; in the second, we used the dataset preprocessed by replacing emojis with their text descriptions (the "text emoji" dataset); in the third, we used the dataset preprocessed by removing any and all emojis (the "no emoji" dataset). We ran each of the six models (five baselines plus one CNN model) on each of these three datasets and calculated the prediction accuracies to obtain the results given in the section below.

5 Results

5.1 Accuracy of Sacarsm Detection

Accuracy for Text Data			
Approach	Basic Data	Text Emoji Data	No Emoji Data
Dummy classifier, uniform:	0.514	0.501	0.522
Dummy classifier, stratified:	0.503	0.472	0.504
Dummy classifier, freqency:	0.512	0.497	0.500
Naive-Bayes:	0.769	0.897	0.771
K-Neighbors:	0.924	0.959	0.956
CNN:	0.986	0.991	0.979

5.2 Performance Comparison

We compare our CNN model with the five other baseline classifiers. The metrics for evaluation are based on accuracy compared across three approaches from our dataset: text with the included emojis, text with emojis removed, and text with the emojis converted to a word counterpart. Overall our CNN outperformed the other models, reporting a higher accuracy in every scenario. This was closely followed by the K-Neighbors model and Naive-Bayes N-Gram Vectors. Lastly, our three baseline classifiers shared a similar accuracy way below that of the machine learning algorithms. Now that we've compared our models the question of if emojis truly make a difference in the accuracy of sarcasm detection must be considered. When comparing the three approaches, the fact the highest accuracies overall were reported from the text with emojis proves our hypothesis correct that emojis do in fact lead to more accurate sarcasm detection when converted to a text descriptor.

6 Conclusion

Emojis have undoubtedly added a new dimension of expression to everyday communication. Through this study and previous it is safe to say that they have the potential to significantly impact and improve sentiment detection. Specifically in our case, the detection of sarcasm in various forms of text. We developed a neural network trained on a dataset where emojis were replaced with their text counterparts. Since our CNN outperformed the other baseline models by a significant amount this allowed us to contribute an effective deep learning model for sarcasm detection. If we had the opportunity to further our research we would consider the possibility of further optimizing the performance of sarcasm detection. This could be done by designing a new deep learning model with features that drastically improve sentiment detection. For example Subramanian et al. proposed a new deep learning model by introducing an attention layer which helps to model the text and emojis simultaneously for sarcasm detection. We could also learn from other state-of-the-art sarcasm detection models like FSNN [9] and CASCADE [10] to implement more novel features as well.

References

[1] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, R. Huang, Sarcasm as contrast between a positive sentiment and negative situation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013, pp. 704–714

[2] R. González-Ibánez, S. Muresan, N. Wacholder, Identifying sarcasm in twitter: a closer look, in: Proceedings of the 49th Annual Meeting on Human Language Technologies, ACL, 2011, pp. 581–586

[3] S.K. Bharti, K.S. Babu, S.K. Jena, Parsing-based sarcasm sentiment recognition in twitter data, in: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), ACM, 2015, pp. 1373–1380.

[4] Hu, X., Tang, J., Gao, H., Liu, H.: Unsupervised sentiment analysis with emotionalsignals. In: Proceedings of the 22nd international conference on World Wide Web.pp. 607–618. ACM (2013)

[5] Hallsmar, F., Palm, J.: Multi-class Sentiment Classification on Twitter using anEmoji Training Heuristic. Master's thesis, KTH Royal Institute of TechnologySchool of Computer Science and Communication (05 2016)

[6] Kelly, R., Watts, L.: Characterising the inventive appropriation of emoji as relationally meaningful in mediated close personal relationships (2015)

[7] Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., Riedel, S.: emoji2vec: Learn-ing emoji representations from their description. arXiv preprint arXiv:1609.08359(2016)

[8] Subramanian, Jayashree, et al. "Exploiting emojis for sarcasm detection." International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. Springer, Cham, 2019.

[9] Ghosh, A., Veale, T.: Fracking sarcasm using neural network. In: Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis. pp. 161–169 (2016)

[10] Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., Mihalcea, R.: Cascade: Contextual sarcasm detection in online discussion forums. arXiv preprint arXiv:1805,06413 (2018)