Sarcasm Detection in News Headlines

Prajwal Chandra Nalla
SBU Computer Science
prajwalchandra.nalla@stonybrook.edu

Sai Pramod Kudapa
SBU Computer Science
saipramod.kudapa@stonybrook.edu

Abstract

Sarcasm plays an important part in human social interaction. Sarcasm detection is of great importance in understanding people's true sentiments and opinions. Application of sarcasm detection can benefit many areas of interest in NLP applications, including marketing research, opinion mining and information categorization. However, detecting sarcasm is difficult for humans and hence it is even more difficult for computers. We tried to detect sarcasm in news headlines using different machine learning models. The input news headlines in the dataset contain a label distinguishing sarcastic and real headlines. The sarcastic headlines are obtained from "The Onion" website and non-sarcastic from "HuffPost" website. We have used accuracy and average F1-score as quantitative evaluation metrics. We have also reported a qualitative evaluation of each model.

1 Introduction

Sarcasm is a form of irony that occurs when there is a discrepancy between the literal meaning of an utterance and its intended meaning. Our goal for this project is to build efficient neural network models for detecting sarcasm in news headlines. Sarcastic comments generally tend to report false information with an intent to invert the sentiment of the expression that they seek to report. Such an inversion of sentiment could be misleading to the unwary people who follow such news posts on their face value. Sarcasm detection is a specific case of sentiment analysis problem. However, unlike sentiment analysis where the sentiment categories are very clearly defined, the borders of sarcasm are not that well defined, making it a difficult task. Sarcasm is a contextual phenomenon and it largely depends on the context, real world knowledge and the tone in which the sentence is written or spoken. Hence, one cannot directly use a set of reference words from the sentence to detect sarcasm.

2 Related Work

Sarcasm detection is a well explored task in the field of NLP. Previously, researchers tried to detect sarcasm from different types of input data like tweets, product reviews, reddit posts and flickr images. Since our project has text as input, we describe below the related work in sarcasm detection on text-based input like tweets, reviews.

To begin with, [5] and [6] tried to detect sarcasm in amazon product reviews using positive and negative words, punctuation, hyperbole, interjection, and bag of words as features. They have tried multiple models like Logistic Regression, Linear SVM, Decision Trees, Random Forest, and Naïve Bayes. [7] used a pattern matching algorithm to find sarcasm in online product reviews.

Authors have explored different features and various machine learning models for detecting sarcasm in tweets. For example, [8] used POS tags, WordNet, Interjections, and punctuations as lexical features; emoticons and user-mentions as pragmatic features to detect sarcasm in tweets. [9] modelled implicit and explicit context incongruity to detect sarcasm in tweets data.

Recent literature on sarcasm detection involve encoding user/author information along with content of the tweet. [10] emphasised on the importance of author context and prior interaction between author and audience in detecting sarcasm. [11] employed behavioural and psychological studies on the users past tweets and constructed a behavioral modeling framework for sarcasm detection. [12]

evaluated user attribute-only models on different social media tasks like stance detection, sentiment analysis and sarcasm detection. Most recently, [13] used BERT classifier to detect sarcasm in tweets and reddit data. [14] also used BERT and GloVe embeddings for detecting sarcasm in tweets.

2.1 Issues and Solutions:

It can be observed that most of the prior work for detecting sarcasm has focused on twitter datasets. But tweets collected using hashtag-based supervision tend to be noisy in terms of language and labels. Also, significant number of tweets in the dataset are just replies to other tweets making it difficult to identify sarcasm without the actual tweet (context). To avoid these issues, we have considered the news headline dataset. Compared to tweets, news headlines have a uniform structure and do not contain any spelling mistakes and informal language. Also, since "The Onion" website publishes only sarcastic news, the labels are accurate, reducing the noise in dataset.

Some authors [15] have used only Convolutional Neural Networks for detecting sarcasm. But it is well known that sarcasm is largely a contextual phenomenon. It is also well known that sequential models like RNN and GRU are good at modelling the context in the text. Hence, to improve the classification we decided to combine the contextual information obtained from sequential models like GRU to the features extracted from CNN layers. Besides this, we also implemented the concept of attention mechanism from [16] which automatically focuses on important semantic information in the sentence.

In addition to quantitative evaluation results provided by most authors, we have examined the cases where the models failed and provided a qualitative analysis.

2.2 Outcomes of the project:

1. We have implemented CNN + Attentive BiGRU for the task of sarcasm detection in news headlines.
2. We have also implemented BERT model combined with CNN for the same task.
3. Our evaluation shows BERT + CNN model performed the best both qualitatively and quantitatively.
4. Based on our work, we conclude that enhancing existing state-of-the-art models for different tasks with BERT based representations might be a promising avenue.

3 Sarcasm detection

Sarcasm detection is technically a sentiment classification task. More specifically, our aim is to detect sarcasm in news headlines. Hence, it is a binary classification task classifying a headline as sarcastic or not. The input is a list of sentence and label pairs. Each sentence is a news headline, and the label indicates whether sentence is sarcastic or not. The output is

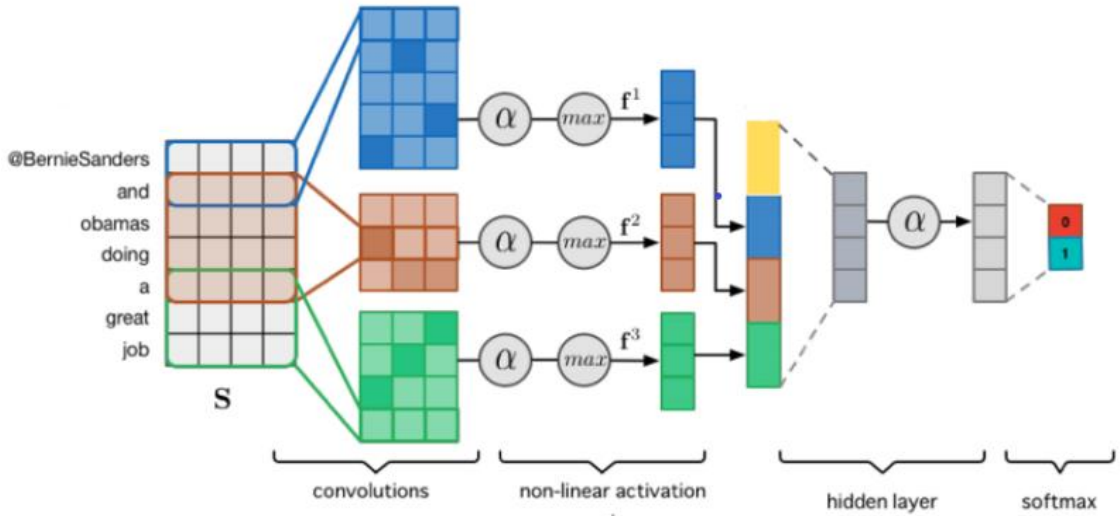

Figure 1. Baseline CNN model (taken from [2])

classification of given sentence as sarcastic or non-sarcastic.

3.1 Baseline Model:

Figure 1 displays our baseline model. We have adopted the CNN architecture from [2] as our baseline model. It is a combination of three CNN layers with different kernel sizes followed by non-linear activation and max pooling for each layer. The three CNN layers have same filter size of 100 but have different kernel size of 4,6, 8, respectively. Each CNN layer is followed by a 'RELU' activation and a 1D max pooling layer. The outputs of three max pooling layers are concatenated and provided as input to a fully connected neural network (dense layer) with "tanh" activation and a dropout layer. This output is again passed to a dense layer with "tanh" activation and a dropout layer. Finally, a classification layer is applied, and the logits obtained are converted to probabilities using softmax.

3.2 Issues:

While the baseline model obtained high accuracy and F1-score, it is still lacking the ability to encode contextual information.

Also, we have used GloVe embeddings to obtain vector representation of words. But studies [17] have shown that word embeddings fail to capture context incongruity especially in the absence of sentiment words i.e., the incongruity patterns captured by word embeddings overly rely on the presence of positive words following a negative situation.

4 Approach

Based on the issues reported above, we tried the below three ideas to improve the baseline model.

4.1 CNN + Attentive BiGRU:

To encode the contextual information from the headlines we have included a Bidirectional GRU layer with attention mechanism. The sentence representation obtained from attentive BiGRU model is concatenated with the output of CNN layers and passed to the dense layers. The rest of the architecture remains same as the baseline model.

4.2 BERT:

To overcome the shortcomings of GloVe embeddings in this task, we decided to try off-the-shelf BERT model. BERT has already shown promising results in several language modelling tasks such as sentiment classification. We have used the CLS token added by the BERT model to classify the sentence. For this network, the BERT model is followed by a fully connected layer with "tanh" activation and dropout layer followed by a final classification layer.

4.3 BERT + CNN:

For this network we have used the same architecture as baseline model but replaced the GloVe embeddings with pre-trained representations obtained from BERT base uncased model. However, the model overfitted heavily due to combination of BERT and three CNN layers. Hence, we have decided to take only one CNN layer.

5 Evaluation

For quantitative evaluation of the model, we have used the standard performance metrics: precision, recall, accuracy, and F1-score. We have provided a comparison between baseline and other models proposed in the project.

5.1 Dataset Details:

For this project, we have used the News Headlines for Sarcasm Detection dataset [1]. The dataset consists of 28,619 headlines, with 13,634 sarcastic headlines obtained from "The Onion" website and 14,985 real headlines from "HuffPost". The average number of headlines is 10 words with standard deviation of 3.4 words. Since the number of samples over 25 words is very small (<0.01%), we truncated the samples to 25 words. The headlines are consistent in length and phrasing and the labels are mutually exclusive. We have split the dataset into training, validation and testing sets with 80%, 10% and 10% respectively.

5.2 Evaluation Measures:

As mentioned earlier, we are using standard performance metrics to evaluate the models. They are defined as follows (assuming sarcastic class as positive class):

Model	Validation accuracy	Test accuracy	F1 score – sarcastic class	Average F1 score
Baseline	85.42	85	84	84
CNN + BiGRU	86.22	86.41	85	85
BERT	91.1	89	89	89
BERT + CNN	91.5	91	91	90

Table 1: Comparison of metrics for different models

$$Accuracy = \frac{\text{Samples correctly classified}}{\text{Total number of samples}}$$

$$Precision = \frac{\text{\# Correctly classified sarcastic}}{\text{\# classified as sarcastic}}$$

$$Recall = \frac{\text{\# Correctly classified sarcastic}}{\text{Total \# of sarcastic instances}}$$

$$F1 = \frac{2^* \text{ Precision } * \text{ Recall}}{\text{Precision } + \text{ Recall}}$$

We can use the above metrics to understand if the model is biased towards one class.

5.3 Baseline Implementation:

The implementation details for baseline models are as follows:
We have used the 100-dimension GloVe embeddings as pre-trained embeddings. We have also used the GloVe common words file which contains 5000 commonly used words. The data is split into batches of 32 and trained for 10 epochs. The filter size for all three CNN layers was taken as 100. The output dimensions for the dense layers are 100 and 50, respectively. Dropout rate is taken as 0.2. We used Adam optimiser for all the models. Similarly, the regularisation lambda is same for the all the models (10^{-4}).

5.4 CNN + BiGRU Implementation:

We used the same hyper parameters as baseline model for the CNN part of this network. For BiGRU, we have taken the hidden size to be 128. Also, we have concatenated the forward and backward representation from the GRU layer.

5.5 BERT Implementation:

We used the BERT base uncased version of the BERT model. The embedding dimension is 768 and it contains 12 layers. For the dense layer, we have taken output dimensions as 100 and "RELU" as activation. We have also taken a dropout rate of 0.2. We have reduced the number of epochs to 4.

5.6 BERT + CNN Implementation:

BERT model implementation is the same as 5.5. For the CNN layer, we have taken output dimensions as 200 and kernel size of 4. This is followed by a 1D max pooling layer. The fully connected layer parameters are the same as reported in 5.5.

5.7 Results:

The comparison between baseline models and other models (5.4, 5.5, 5.6) is shown in Table 1. From Table 1, it can be observed that, BERT + CNN had the best performance among all the models. There is a significant improvement in the metrics when BERT model was used to generate the word embeddings. With very little hyper-parameters tuning, BERT clearly exhibits the benefits of utilizing pretraining models to obtain better results compared to other models.

5.8 Analysis:

We analysed example sentences to determine the probable reasons for which the model has classified the sentences to respective categories.

5.8.1 Examples where baseline failed but BERT did not:

1. *{"label": sarcastic (1), "text": "clinton gets full day's relief with one spray of flonase"}*

For this example, baseline model failed possibly due to incapability to capture semantic dissimilarity between full day and

one spray. However, BERT model classified the sentence correctly.

2. *{"label": real (0), "text": "nancy pelosi, paul ryan get mixed marks from their parties"}*

In English language some words will have multiple senses depending on the context. So, baseline model using GloVe embeddings might capture different sense of the word “parties” in the above example leading to incorrect classification. However, since BERT model can detect polysemy in words efficiently, it has accurately predicted the above example as non-sarcastic.

5.8.2 Examples where baseline and BERT both failed:

1. {"label": 1, "text": "miss america pageant adds sweatpants and messy bun competition"}

The above sentence requires real word knowledge about pageant competitions to understand the implicit sarcasm. It is not possible to infer the meaning just from the words in the sentence. Hence, both models failed.

2. {"label": 1, "text": "214 executed in wacky bolivian prison mix-up"}

5.9 Code:

The Github repository link for this project can be found here. All the files including data, code and README can be found in the repository.

6 Conclusions

For this project, we have selected a baseline model for detecting sarcasm in news headlines and tried to improve it by changing the pre-trained embeddings and adding an additional sequential layer. From the quantitative evaluation, we can say that Convolutional networks can demonstratively learn to distinguish sarcastic text from serious text with no context. However, BERT model had shown significant improvements in both quantitative and qualitative analysis. We have also seen that both the CNN and BERT models failed when real world knowledge was required to detect sarcasm. Unfortunately, we were unable to determine the general rules of sarcasm from the network results alone, although more detailed analysis could still be done in the future.

7 References

[1] Rishabh Misra. 2018. News headlines dataset for sarcasm detection. Data retrieved from Kaggle::https://www.kaggle.com/rmisra/newsheadlines-dataset-for-sarcasm-detection.

[2] Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mario J Silva. 2016. Modelling context ´ with user embeddings for sarcasm detection in social media. arXiv preprint arXiv:1607.00976.

[3] Rishabh Misra, Prahal Arora, 2019, “Sarcasm Detection using Hybrid Neural Network”. arXiv:1908.07414

[4] Compilation of literature on Sarcasm Detection::https://github.com/anebz/papers#sarcasm-detection

[5] K. Buschmeier, P. Cimiano and R. Klinger, "An impact analysis of features in a classification approach to irony detection in product reviews," *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 42-49, 2014.

[6] D. Dmitry, O. Tsur and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in twitter and amazon," in Proceedings of the fourteenth conference on computational natural language learning, 2010.

[7] O. Tsur , D. Davidov and A. Rappoport, "A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews," InICWSM , pp. 162-169, 2010.

[8] R. González-Ibánez, S. Muresan and N. 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, vol. 2, pp. 581-586, 2011.

[9] Aditya Joshi, Vinita Sharma, Pushpak Bhattacharyya, “Harnessing Context Incongruity for Sarcasm Detection” Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. DOI: 10.3115/v1/P15-2124.

[10] Silviu Oprea, Walid Magdy, “Exploring Author Context for Detecting Intended vs Perceived Sarcasm”, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2854–2859.

[11] Rajadesingan et al., 2015, “Sarcasm Detection on Twitter: A Behavioral Modeling Approach”, WSDM '15: Proceedings of the Eighth ACM

International Conference on Web Search and Data Mining. February 2015482 pages.
[12] Veronica et al, 2019, “Tweet Classification without the Tweet: An Empirical Examination of User versus Document Attributes”, Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science.
[13] Baruah et al., 2020, “Context-Aware Sarcasm Detection Using BERT”, Proceedings of the Second Workshop on Figurative Language Processing, pages 83–87.
[14] Akshay Khatri, Pranav P, “Sarcasm Detection in Tweets with BERT and GloVe Embeddings”, Proceedings of the Second Workshop on Figurative Language Processing, pages 56–60.
[15] Poria et al., 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:1610.08815
[16] Zhou et al., 2016. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 207-212.
[17] Joshi et al., “Are Word Embedding-based Features Useful for Sarcasm Detection?” arXiv:1610.00883.