Sarcasm Detection using Hybrid Neural Network

Rishabh Misra
A53205530
University of California San Diego
r1misra@ucsd.edu

Prahal Arora
A53219500
University of California San Diego
prarora@ucsd.edu

Abstract

Sarcasm Detection has enjoyed great interest from the research community, however the task of predicting sarcasm in a text remains an elusive problem for machines. Past studies mostly make use of twitter datasets collected using hashtag based supervision but such datasets are noisy in terms of labels and language. To overcome these shortcoming, we introduce a new dataset which contains news headlines from a sarcastic news website and a real news website. Next, we propose a hybrid Neural Network architecture with attention mechanism which provides insights about what actually makes sentences sarcastic. Through experiments, we show that the proposed model improves upon the baseline by $\sim$ 5% in terms of classification accuracy.

1 Limitations of previous work

(Amir et al., 2016) propose to use a CNN to automatically extract relevant features from tweets and augment them with user embeddings to provide more contextual features during sarcasm detection. However, this work is limited in following aspects:

- Twitter dataset used in the study was collected using hashtag based supervision. As per various studies [(Liebrecht et al., 2013; Joshi et al., 2017)], such datasets have noisy labels. Furthermore, people use very informal language on twitter which introduces sparsity in vocabulary and for many words pre-trained embeddings are not available. Lastly, many tweets are replies to other tweets and detecting sarcasm in these requires the availability of contextual tweets.
- The modeling proposed is quite simplistic. Authors use CNN with one convolutional layer to extract relevant features from text which are then concatenated with (pre-trained) user embeddings to produce the final classification score. However, some studies like (Yin et al., 2017) show that RNNs are more suitable for sequential data. Furthermore, authors propose a separate method to learn the user embeddings which means the model is not trainable end to end.
- Authors do not provide any qualitative analysis from the model to show where the model is performing well and where it is not.
- Upon analysis, we understand that detecting sarcasm requires understanding of common sense knowledge without which the model might not actually understand what sarcasm is and just pick up some discriminative lexical cues. This direction has not been addressed in previous studies to the best of our knowledge.

In section 2, we describe the dataset collected by us to overcome the limitations of Twitter datasets. In section 3, we describe the network architecture of the proposed model. In section 4 and section 5, we provide experiment details, results and analysis. To conclude, we provide few future directions in section 6.

2 Dataset

To overcome the limitations related to noise in Twitter datasets, we collected a new *Headlines* dataset[1] from two news website. *TheOnion*[2] aims at producing sarcastic versions of current events

[1] https://github.com/rishabhmisra/Headlines-Dataset-For-Sarcasm-Detection

[2] https://www.theonion.com/

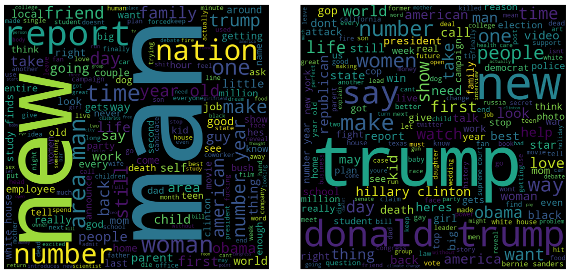


Figure 1: Wordclouds of sarcastic and non-sarcastic headlines respectively.

and we collected all the headlines from News in Brief and News in Photos categories (which are sarcastic). We collect real (and non-sarcastic) news headlines from *HuffPost*[3]. As a basic exploration, we visualize the word clouds in Figure 1 through which we can see the types of words that occur frequently in each category. The general statistics of this dataset along with dataset provided by Semeval challenge[4] are given in Table 1. We can notice that for Headlines dataset, where text is much more formal in language, percentage of words not available in word2vec vocabulary is much less than Semeval dataset.

Statistic/Dataset	**Headlines**	**Semeval**
# Records	26,709	3,000
# Sarcastic records	11,725	2,396
# Non-sarcastic records	14,984	604
% word embeddings not available	23.35	35.53

Table 1: General statistics of datasets.

This new dataset has following advantages over the existing Twitter datasets:

- Since news headlines are written by professionals in a formal manner, there are no spelling mistakes and informal usage. This reduces the sparsity and also increases the chance of finding pre-trained embeddings.

- Furthermore, since the sole purpose of *TheOnion* is to publish sarcastic news, we get high quality labels with much less noise as compared to twitter datasets.

- Unlike tweets which are replies to other tweets, the news headlines we obtained are self-contained. This would help us in teasing apart the real sarcastic elements.

3 Network Architecture

The original architecture of (Amir et al., 2016) takes pre-trained user embeddings (context) and tweets (content) as input and outputs a binary value for sarcasm detection. We tweaked this architecture to remove the user-context modeling path since the mention of sarcasm in this dataset is not dependent on authors but rather on current events and common knowledge. In addition to that, a new LSTM module is added to encode the left (and right) context of the words in a sentence at every time step. This LSTM module is supplemented with an Attention module to reweigh the encoded context at every time step.

We hypothesize that the sequential information encoded in LSTM module would complement the existing CNN module in the original architecture of (Amir et al., 2016) which captures regular n-gram word patterns throughout the entire length of the sentence. We also hypothesize that attention module can really benefit the task at hand. It can selectively emphasize on incongruent co-occurring word phrases (words with contrasting implied sentiment). For example, in the sentence "majority of nations civic engagement centered around oppressing other people", our attentive model can emphasize on occurrence of 'civic engagement' and 'oppressing other people' to classify this sentence as sarcastic. The detailed architecture of our model is illustrated in figure 2.

The LSTM module with attention is similar to the one used to jointly align and translate in a Neural Machine Translation task (Bahdanau et al., 2014). A BiLSTM consists of forward and backward LSTMs. The forward LSTM calculates a sequence of forward hidden states and the backward LSTM reads the sequence in the reverse order to calculate backward hidden states. We obtain an annotation for each word in the input sentence by concatenating the forward hidden state and the backward one. In this way, the annotation h_j contains the summaries of both the preceding words and the following words. Due to the tendency of LSTMs to better represent recent inputs, the annotation at any time step will be focused on the words around that time step in the input sentence. Each hidden state contains information about the whole input sequence with a strong focus on the

[3] https://www.huffingtonpost.com/
[4] https://competitions.codalab.org/competitions/17468

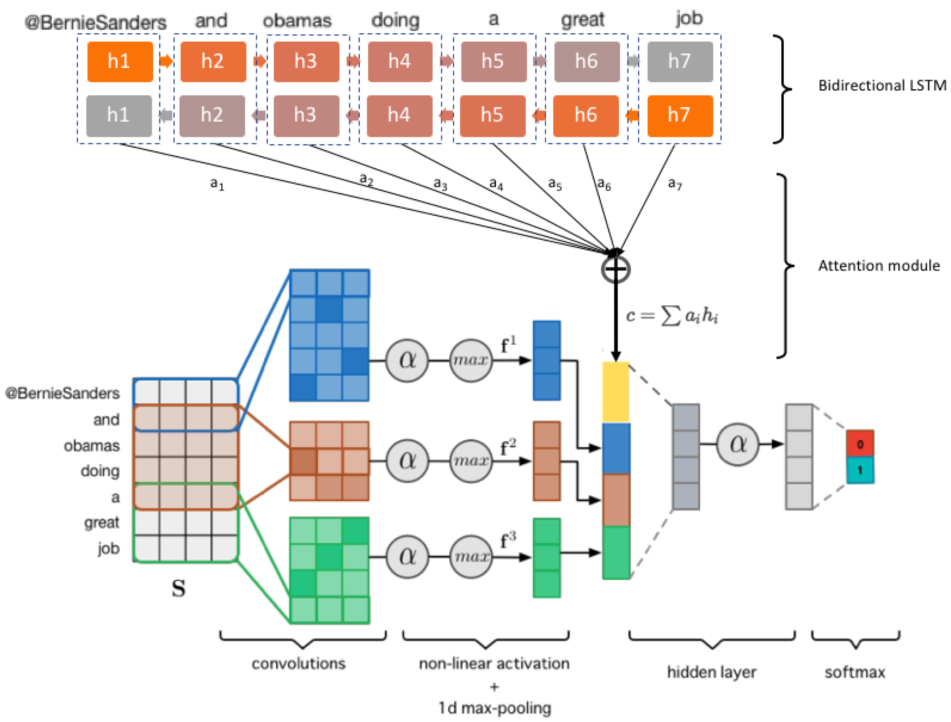


Figure 2: Hybrid Network Architecture

parts surrounding the corresponding input word of the input sequence. The context vector **c** is, then, computed as a weighted sum of these annotations.

$$\mathbf{c} = \sum_{i=1}^{N} \alpha_i h_i$$

Here, α_i is the weight/attention of a hidden state h_i calculated by computing Softmax over scores of each hidden state. The score of each individual h_i is calculated by forwarding h_i through a multi-layer perceptron that outputs a score.

The context vector **c** is finally concatenated to the output of the CNN module. Together, this large feature vector is then fed to an MLP which outputs the binary probability distribution of the sentence being sarcastic/non-sarcastic.

4 Experiments

4.1 Baseline

With new dataset in hand, we tweak the model of (Amir et al., 2016) and consider it as a baseline. We remove the author embedding component because now the sarcasm is independent of authors (it is based on current events and common knowledge). The CNN module remains intact.

4.2 Experimental Setup

To represent the words, we use pre-trained embeddings from word2vec model and initialize the missing words uniformly at random in both the models. These are then tuned during the training process. We create train, validation and test set by splitting data randomly in 80:10:10 ratio. We tune the hyper-parameters like learning rate, regularization constant, output channels, filter width, hidden units and dropout fraction using grid search. The model is trained by minimizing the cross entropy error between the predictions and true labels, the gradients with respect to the network parameters are computed with backpropagation and the model weights are updated with the AdaDelta rule. Code for both the methods is available on GitHub[5].

5 Results and Analysis

5.1 Quantitative Results

We report the quantitative results of the baseline and the proposed method in terms of classification accuracy, since the dataset is mostly balanced. The final classification accuracy after hyper-parameter tuning is provided in Table 2. As shown, our model improves upon the baseline by $\sim 5\%$ which

[5]https://github.com/rishabhmisra/Sarcasm-Detection-using-CNN

supports our first hypothesis mentioned in section 3. The performance trend of our model is shown in Figure 3.

Implementation	Test Accuracy
Baseline	84.88%
Proposed method	89.7%

Table 2: Performance of baseline and proposed method in terms of classification accuracy

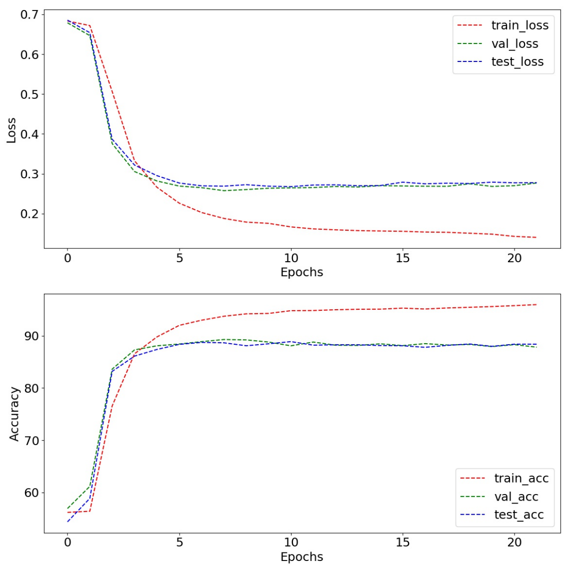
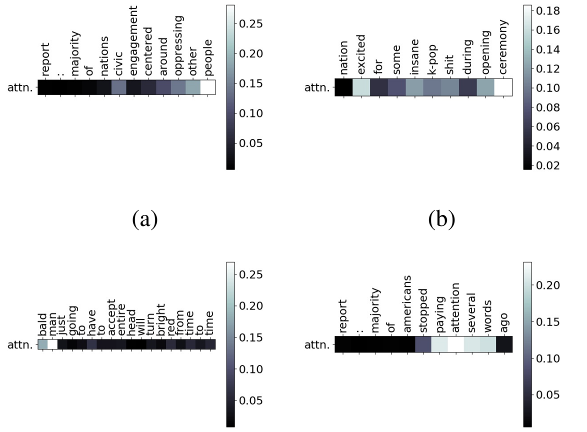


Figure 3: Loss and accuracy trend of the proposed method.

5.2 Qualitative Results

We visualize the attention over some of the sarcastic sentences in the test set that are correctly classified with high confidence scores. This would help us better understand if our hypothesis is correct and provide insights into sarcasm detection process. Figure 4a and Figure 4b show that the attention module emphasizes on co-occurrence of incongruent word phrases within each sentence, such as ‘civic engagement’ & ‘oppressing other people’ in 4a and ‘excited for’ & ‘insane k-pop sh*t during opening ceremony’ in 4b. This incongruency is an important cue for us humans too and supports our second hypothesis mentioned in section 3. This has been extensively studied in (Joshi et al., 2015). Figure 4c shows that presence of ’bald man’ indicates that this news headline is rather insincere probably meant for ridiculing someone. Similarly, ‘stopped paying attention’ in



(a) (b) (c) (d)

Figure 4: Visualizing attention over the entire length of the sarcastic sentences

Figure 4d has more probability to show up in satirical sentence, rather than a sincere news headline.

6 Future Work

Given the time crunch, we are left with several unexplored directions that we would like to work on in future. Some of the important directions are as follows:

- We can do ablation study on our proposed architecture to analyze the contribution of each module.

- The approach proposed in this work could be considered as a pre-computation step and the learned parameters could be tuned further on Semeval dataset. Our intuition behind this direction is that this pre-computation step would allow us to capture the general cues for sarcasm which would be hard to learn on Semeval dataset alone (given its small size). This type of transfer learning is shown to be effective when limited data is available [(Pan and Yang, 2010)].

- Lastly, we observe that detection of sarcasm depends a lot on common knowledge (current events and common sense). Thus, we plan to integrate this knowledge in our network so that our model is able to detect sarcasm based on which sentences deviate from common knowledge. Recently, (Young et al., 2017) integrated such knowledge in dialogue systems and the ideas mentioned could be adapted in our setting as well.

Contributions from Team Members

We did pair programming for this assignment. Both of the team members contributed equally.

References

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)* 50(5):73.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. volume 2, pages 757–762.

Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *WASSA@NAACL-HLT*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923* .

Tom Young, Erik Cambria, Iti Chaturvedi, Minlie Huang, Hao Zhou, and Subham Biswas. 2017. Augmenting end-to-end dialog systems with commonsense knowledge. *arXiv preprint arXiv:1709.05453* .