Automatic Sarcasm Detection in Long-Form Forum Comments

Jonathan Zhou

jonathan.zhou@berkeley.edu

04.12.2019

Berkeley MIDS Spring '19 w266 Final Project

ABSTRACT

Automatic sarcasm detection has been difficult for machines because the exact same sentence can be interpreted both literally and sarcastically depending on the context and opinions of the author. It is a task that humans have a lot of difficulty detecting as well, with many commenters commonly using the '\s' tag to directly indicate when they are actually being sarcastic on social media. What typically helps humans identify sarcasm is the context that the statement was made in and any incongruity of the sarcastic statement with its context. In previous work, a dataset of reddit comments on political and religious topics was manually annotated by three humans on whether segments of those comments were sarcastic or not. Because each reddit comment were comprised of multiple segments, these annotated sarcastic segments had additional context of the accompanying segments in the comment to build a model for detecting sarcastic comments. A simple baseline bag-of-words model was built to classify comments, but performed poorly with 38.3% accuracy.

To see if these results could be improved upon in this paper, we did additional preprocessing to the reddit dataset to clean it up and extract additional metadata that can provide additional context. In addition to re-applying bag-of-words, we explored using bidirectional LSTM and CNN deep learning techniques as well in the hopes of improving accuracy.

As a result, we found that LSTM and CNN yielded poorer performance against the baseline simple bag-of-words method, however the updated bag-of-words model with additional metadata and features had significant improved performance. Even so, with a 63.44% accuracy, the model isn't accurate enough to predict sarcasm in reddit comments. A significant challenge was the skewed dataset having more literal comments than sarcastic.

INTRODUCTION

Detecting sarcasm in unstructured text has been a difficult challenge for humans, not to mention machines. This is because the same statement can be intended to be interpreted literally or sarcastically depending on the context and the author's perspective. In fact, the exact same sentence, can be sarcastic or genuine depending on who is stating it and their previous opinions. Because of the

This topic has been explored extensively and Joshi et al [1] have compiled many notable papers and datasets on sarcasm. The vast majority of research focuses on the short-form content found on Twitter, a portion of papers looked at more long-form content, primarily from various reddit threads. No research took into account on classifying sarcasm / irony in more formal unstructured content such as non-fiction and news reports because of the low prevalence of sarcasm in these articles.

While there has been a lot of focus on automatically detecting sarcasm in short-form content, the research into long-form as been relatively lagging. This paper builds on the work of the 2014 paper by Wallace et al [2], who took a portion of reddit comments from political and religious subreddits and manually classified each sentence in a comment as sarcastic or not. Using a simple unigram and bigram bag-of-words based text-classification model to identify sarcastic comments. This work focused on several key markers, such as multiple exclamation points or question marks that previous work has found to be good indicators of sarcasm [6,7]. The conclusion reached was that this approach was not able to accurately identify sarcastic comments.

The motivation of this paper is to improve on the work of Wallace et al and leverage their manually annotated corpus of reddit comments with more modern deep learning techniques to see if improvements can be gained on the accuracy of their work. In addition to analyzing the raw text and identifying key features within the text, additional metadata will be taken into consideration. The methodology used in this paper will be referred to as the the baseline method going forward in this paper.

BACKGROUND

The corpus of data includes comments made from the following subreddits: atheism, politics, Conservative, technology, progressive, Christianity. It includes 10,039 reddit comments, however only 3,550 have been labeled. These labeled comments are further broken down into individual sentences / segments that have been labeled as ironic or not. There are on average 3 segments per comment. Three people independently classified each sentence and provided a confidence indicator of 1-3 of their classification. The independent classifications were fairly well correlated. Only comments that all classifiers identified as sarcastic were classified as sarcastic for data model for building the model. For further information on the data collection methodology, refer to the original paper Wallace et al [2].

Their premise was that although, a segment of a comment was sarcastic, the context provided by the text in the rest of the comment can help more accurately predict the model. Following this logic, a model was built using linear kernel SVM, while maximizing for F1 score, the average F1 score of the model against the manual classification was 0.383. The main conclusion was that even with the additional context, these methods were not able to accurately detect sarcasm.

METHODS

Across other papers focused on sarcasm, the core premise is that additional context around a statement is required to detect sarcasm and that the semantics of the statement itself is rarely sufficient. Incongruity of intention between the context and a statement is usually a good indicator of sarcasm or incongruity of sentiment within the statement itself. Other methods used include identifying users that make a sarcastic statement are more likely to have used it in other authored content as well.

The baseline method uses the accompanying statements around a statement to accomplish this. The baseline bag-of-words was great at generating intrinsic features from the text and recognizing notable tokens of sarcasm (eg. multiple exclamation points). However, it does not properly account for the relationship between words or factor in other available features, such as the reddit metadata on upvotes, downvotes, subreddit and author.

To improve upon this initial work, several additional features that were previously ignored will be included as well.

DATA PREPARATION

The baseline method performed very little data preprocessing. The only preprocessing was around identifying markers that were previously correlated with sarcastic comments such as question marks, exclamation points, emoticons and all caps words. Special keywords were appended to the comment when identified.

For our work, we performed additional preprocessing on the comments including:

- Convert all text to lowercase
- Removing numbers
- Removing punctuation

Apart from the preprocessing the text, metadata was vectorized and preprocessed as well and captured separately from the vectorized text. The metadata collected includes vectors identifying:

- Presence of exclamation marks
- Presence of question marks
- Presence of multiple adjoining exclamation and question marks
- Presence of all uppercase words
- Vectorized representation of the comment redditor
- Vectorized representation of the subreddit topic
- Upvote counts
- Downvote counts

MODELING

Three methods were explored to see if these modeling techniques can perform better than the baseline method. For all methods, training was done with a 80/20 split. These techniques are:

1. Bidirectional LSTM
2. CNN
3. Improved BOW with preprocessed text and metadata

For LSTM technique, the vectorized text is first processed through an embedding layer, bidirectional LSTM layer with l2 regularization, affine and activation transformations. Then, the hidden layer outputs are concatenated with the metadata before an additional affine and activation layer to result in the output.

For CNN technique, the vectorized text is processed through an embedding layer, 3 1D convolutional layers with kernel size of 5, affine and activation transformations. Concatenation of the hidden layer outputs with the metadata before an additional affine and activation layer to result in the output.

For the improved bag-of-words model, the metadata and vectorized input text are concatenated before using stochastic gradient descent and optimized for accuracy. Five-fold cross-validation was performed calculate a mean accuracy.

RESULTS AND DISCUSSIONS

After training models using these three methods, it was found that improved bag-of-words performed better than LSTM and CNN. Below are the accuracies found for each model:

Model	Accuracy
Baseline BOW	38.3%
Improved BOW	63.44%
Bidirectional LSTM	32.04%
CNN	32.04%

Figure 1. Reported accuracy from modeling techniques used

Comparing the accuracy exhibited by the above, the deep learning techniques both performed poorly against bag-of-words. With the additional text preprocessing and metadata extraction, we were able to significantly improve the performance from using the bag-of-words technique.

Looking at the results from the improved bag-of-words model, the incorrectly detected comments were typically much shorter in length. For these shorter comments, 5 out of 6 were false positives. For comments incorrectly predicted as false negatives, they were more likely to be longer comments as well. This is an expected outcome since our dataset is heavily skewed to be literal comments with 4 out of 5 comments being literal.

CONCLUSION

Although we were able to significantly improve performance from the baseline by 25.1%, the model was still not accurate enough to be confidently detecting sarcasm.

There are several considerations for further improvements and exploration to see if we can build a better automatic sarcasm detection model.

The first consideration is the relatively small size of the dataset, at only 3,550 comments. For example, was not enough data to get a proper impression of whether a redditor had a pattern of making sarcastic comments, with an average of 1.58 comments made per redditor in the dataset. The dataset was also skewed towards literal comments, with relatively fewer sarcastic ones.

Other considerations to be considered in the future include techniques found in other long-form automatic sarcasm detection papers also relied on capturing sentiment and extracted entities features [1].

It is difficult enough for humans to detect sarcasm in many cases, not to mention machines. While there are challenges with skewed datasets and poor annotations, there is significant room for further exploration on automatic sarcasm detection.

REFERENCES

1. Aditya Joshi, Pushpak Bhattacharyya, Mark J Carman, 2017. Automatic Sarcasm Detection: A Survey
2. Byron C Wallace, Laura Kertz Do Kook Choe, and Eugene Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). 512–516.
3. Do Kook Choe Wallace, Byron C and Eugene Charniak. 2015. Sparse, Contextually Informed Models for Irony Detection: Exploiting User Communities, Entities and Sentiment. In ACL.
4. Amir Silvio, Byron C Wallace, Hao Lyu, and Paula Carvalho Mario J Silva. 2016. Modelling Context with User ´ Embeddings for Sarcasm Detection in Social Media. CoNLL 2016 (2016), 167.
5. Aniruddha Ghosh and Tony Veale. 2016. Fracking Sarcasm using Neural Network. WASSA NAACL 2016 (2016).
6. Davidov, O Tsur, and A Rappoport. 2010. Semisupervised recognition of sarcastic sentences in twitter and amazon. pages 107–116
7. Burfoot and T Baldwin. 2009. Automatic satire detection: are you having a laugh? In ACL-IJCNLP, pages 161–164. ACL.