Convolutional Neural Network and Recurrent Neural Network Model Comparisons on Sarcasm Setection on Reddit

Zhiyi Zhao zzhao243@wisc.edu
Jiaxi Zheng jzheng83@wisc.edu
Nicholas Mandal nmandal2@wisc.edu

This counts as plagiarism if you don't put it into quotation marks as it is a literal copying of an existing sentence fragment

1. Introduction

Sarcasm is defined by Websters Dictionary as a mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual. As fields such as Natural Language Processing and sentiment analysis mature and grow in popularity, a yearning to detect and understand the more subtle and sophisticated parts of language is desired in order to perform highly intricate, human-like tasks. Recognizing sarcastic statements can be very useful when it comes to improving automatic sentiment analysis of data collected from social networks as well as helping customer service assistance departments understand the intentions and real opinions of customers when crawling feedback and complaints[1]. For example, a sarcastic remark such as "I love it when I get stuck in traffic" may be construed as having a positive sentiment to a naive detector due to the faint and profound nature of sarcasm, but in fact it is most likely a negative conjecture coming from someone who is frustrated to be in traffic. Sarcasm is so complicated to detect because it is very reliant on the context of a situation and is most noticeable in an individuals tone and ambivalence in their spoken word[3].

Detection and interpretation of sarcasm has proven to be especially difficult for humans in text form because sarcasm thrives in ambiguous situations, and humans are unable to pick up on the signals that go missing as a part of the situational context[10]. As a result, many people will try and alleviate the incurring confusion on social media platforms by tagging their remarks "#sarcastic" on Twitter or s on Reddit. Is there a way to systematically detect or pick up on sarcasm from text or are we doomed to being as naive as a bot that is unable to notice that there is no way that someone could love being in traffic?

While sarcasm detection seems like an audacious feat, our goal is to train and develop a sophisticated neural network model that is able to determine what is and what is not sarcasm without contextual knowledge. There has been a number of different approaches to solving the sarcasm detection problem with varying successes. The social media platform, Twitter, is a goldmine for text data and more researchers have turned to it in order to help refine their sentiment analysis research with tweet classification. One of the more renown implementations of sarcasm detection in tweets is from Maynard and Greenwood who trained their model based off of hashtag tokenization and its effect on sarcasm and sentiment[7]. Liebrecht et al. [6] used the Balanced Winnow Algorithm to classify tweets as sarcastic or not using the aforementioned "#sarcasm" hashtag, with a 75% accuracy. Tsur et al.[2] used a semi-supervised approach to classify sentences in online product reviews into a number of different sarcastic classes, and reported an F-measure of 82.7% on the binary sarcasm detection class. There exists a wide variety of possible approaches and implementation to solving the sarcasm detector problem, however we feel that contrasting two different methods by way of the convolutional neural network and the recurrent neural network will prove the most successful.

Very nice introduction and good background research!

2. Motivation

In recent years, as the improvements of polarity classification slow down, the focus of natural language processing based on online social media and networks has gradually narrowed down to more detailed and challenging sentiment analysis goals[5]. Sarcasm detection is one of the most complex topics, as well as one of the most interesting ones. Sarcasm could change the polarity of the sentence completely. For instance, the meaning of the sentence "That was FANTASTIC." in a sarcastic manner differs completely from a non-sarcastic context. What notes the difference here is the capitalized font. In this way, sarcasm detection matters significantly in analyzing text sentiments which applies to various conditions ranging from products reviews to movie comments. On the other hand, since the validation of sentiment analysis is hard with respect to the statistic that even for humans, they merely agree 79% in the test due to the sarcasm and other subtle emotional factors [9], the research on sarcasm detection is important in this field. In this case, the statistical significance of a well-developed sarcasm detection model could be close to human analysis even though the test accuracy may not as high as the accuracy of other deep learning topics such as image reading

and classifications.

As for the models, we select the convolutional neural networks (CNN) because it is one of the most popular deep learning models and convenient to implement using pytorch. We intend to model it with multiple layers trying different activation functions and different regularization methods. However, CNN is usually applied to visual imageries and known for its accuracy on image recognition, thus, we plan to add a recurrent neural network model as supplement and contrast. RNN makes it convenient to analyze time series data, such as the subreddits in this context, however, it has problems with vanishing gradient descent and exploding gradient. These two models both have their own advantages and disadvantages. Comparing their performance in textual context could be interesting to learn about.

3. Evaluation

Cross-validation is an essential step before formally training neural models, while it divides the Reddit data into training and validation dataset. The training dataset, which consists of around 70% to 80% of the whole dataset, helps the neural model learn how would the significant factors/patterns impact the prediction. On the other hand, Testing dataset is used to measure the efficiency of the learning model through accuracy computation. In contrast to handling the dataset manually, cross-validation randomly splits the dataset into k folds and create more testing epoch. It ensures that the training dataset can represent the distribution of all data without severe bias, which allows us to be more confident on the algorithm.

After gathering our training and testing dataset through cross validation, in each testing epoch, we compute the accuracy (efficacy) of the two learning models (that said, two models will use the homogeneous unit to minimize sample-to-sample variation). The false positive, false negative, true positive, true negative rate and eventually the overall accuracy will be computed. Accuracy is equal to the number of currency prediction (includes TRUE positive and FALSE negative) divided by the total number of observations. The figure 1 below illustrates the idea.

We aim to use the two-sample z test (hypothesis testing) to examine the difference between the overall accuracy, false positive, false negative, true positive, true negative rate of two models in m epochs. We will set up a null hypothesis (H0: Difference between the the variables above of convolutional and recurrent model are the same), and three alternative hypotheses (Ha: convolutional ones is not equal to/ bigger/smaller than the one of recurrent model) in general. The significance level is alpha= 0.05. Confidence interval will also be considered.

The MAIN success outcome of our project will be the existence of a clear gap between two models overall accuracy.

	TRUE (experiment)	FALSE (experiment)
TRUE/Sarcastic (actual)	# of 1 in experiment given that the actual result is 1/ # of actual 1	# of 0 in experiment given that the actual result is 1/ # of actual 1
FALSE/non-Sarcastic (actual)	# of 1 in experiment given that the actual result is 0/ # of actual 0	# of 0 in experiment given that the actual result is 0/ # of actual 0

Figure 1.

Nevertheless, we expect to see if any model favors a specific of sarcastic pattern, and that is the reason why we introduce the hypothesis testing of false positive, false negative, true positive, true negative rate. These four hypotheses would be the second significant factor we would like to explore in the project.

Besides the difference between the efficiency of these two neural models, we would like to make sure the test accuracy is good enough without overfitting(eg. training accuracy is significantly bigger than the validation one). We hope to get around 80% average accuracy of the test dataset.

4. Resources

The corpus we will use are gathered by Khodak et al.[4] and updated in 2018. It contains 1.3 million sarcastic comments from Reddit. Each sarcastic comment is labeled #s (label = 1) by the Redditor itself, avoiding any independent classifier bias. It also contains numerous non-sarcastic comments for doing both balanced and unbalanced designs. The actual datasets we will use for experiment are shuffled and divided into train, and test subsets by Ofer[8] on Kaggle with both balanced and unbalanced versions. Other influential factors such as author, date, and subreddits are also included.

As for computational tools: even though most papers about sarcasm detection use Tensorflow, pytorch is our first choice as it has a higher adoption of deep learning community compared to TensorFlow. We can take advantage of its strong Management of computation that similar to Numpy, but with a stronger hardware, such as GPU. However, using pytorch to generate plots might need much hard work the whole process might become not as efficient as doing it using R. Thereby,We might consider replace pytorch with R when generating summaries, plots or even regression analysis if necessary.

In addition to software technique, we are also aware of the computational hardware. Because of the relatively large size of the dataset (>500 megabytes), CPU is not sufficient

for the computation. To use all the data we acquired, we therefore prefer using GPU instead, which is cloud computing (eg. google).

5. Contributions

For the proposal, we prompted up ideas and selected the topic based on our research and discussion together. We also looked through numerous datasets together and the finalized version was found by Nick. Introduction was ~~wrote~~ written by Nick, motivation was by Zhiyi, evaluation was by Jiaxi, resources was by Jiaxi and Zhiyi, contribution and formatting were by Zhiyi. For the experiment, we expect the coding tasks will be done together with approximately 40% by Nick, 30% each for Jiaxi and Zhiyi. The project writing will be equally distributed, the PowerPoint will be made 37.5% by Zhiyi, 37.5% by Jiaxi and 25% by Nick. The estimated number of work contribution of experiment and following tasks could be changed according to actual conditions.

References

[1] M. Bouazizi and T. Ohtsuki. A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4:1–1, 01 2016.

[2] D. Davidov, O. Tsur, and A. Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. *CoNLL 2010 - Fourteenth Conference on Computational Natural Language Learning, Proceedings of the Conference*, 01 2010.

[3] D. V. G and S. Chandra Sekharan. A comprehensive study on sarcasm detection techniques in sentiment analysis, 06 2018.

[4] M. Khodak, N. Saunshi, and K. Vodrahalli. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*, 2017.

[5] S. Le Hoang, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network.

[6] C. Liebrecht, F. Kunneman, and A. Van den Bosch. The perfect solution for detecting sarcasm in tweets not. pages 29–37, 06 2013.

[7] D. Maynard and M. A. Greenwood. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. 2014.

[8] D. Ofer. Sarcasm on reddit - 1.3 million labelled comments from reddit. 2018.

[9] M. Ogneva. How companies can use sentiment analysis to improve their business. *Retrieved August*, 30, 2010.

[10] S. Peters. Why is sarcasm so difficult to detect in texts and emails?, 03 2018.

Very nice report overall.