Sarcasm Detection with Deep Learning

Percival Chen, Ollie Downs
Spring 2020

Abstract

Sarcasm has been considered one of the most difficult challenges in Natural Language Processing. In this paper, we show several methods of categorizing sarcasm, using both machine learning models and deep learning models. We find that of the machine learning models, logistic regression was the most effective, with accuracy rates up to 83% on various datasets. We then explore several deep learning models and find that using convoluted neural networks was the most effective, with accuracy rates consistently up to 72%.
Keywords: sarcasm, deep learning, machine learning

Introduction

Sarcasm is a notoriously difficult concept for non-humans to grasp[1]. We decided to attempt to use neural nets and sentiment analysis to train a model that could identify sarcasm in text with above 50% accuracy. Our second goal was to analyze the results of various algorithms and produce recommendations for future work on the subject.

Identifying sarcasm is important in the processing of text because it can clarify the meaning of ambiguous sentences, allowing the true intent of the speaker to be revealed[2]. The consequence of misunderstanding someone's context is detrimental to understanding the intent of the person, and thus informing future actions; for example, not recognizing sarcasm can be problematic for continuing conversations or following instructions. Understanding one's tone is also important in communication contexts[3]. A piece of news or information presented sarcastically but perceived without context can not only misinform people, but contribute to the (often rapid) spread of misinformation, which has devastating consequences.

In a data science context, the identification of sarcasm can be crucial in streamlining processes and increasing safety. Content moderators' loads may be lightened if content can be pre-screened

[1] Tungthamthiti, Piyoros, et al. "Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches." https://www.aclweb.org/anthology/Y14-1047.pdf

[2] Poria, Soujanya, et al. "A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks." ArXiv:1610.08815 [Cs], July 2017. arXiv.org, http://arxiv.org/abs/1610.08815.

[3] A. D. Dave and N. P. Desai, "A comprehensive study of classification techniques for sarcasm detection on textual data," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 1985-1991.

considering sarcasm as a factor. For example, references to doing something illegal may be allowed when those references are sarcastic, but at this point in computer history, it requires a person to make that distinction. Automating this would allow moderators to focus on more, less clear content, narrowing the amount of information that they would need to process manually on forums.

Identifying sarcasm is also an important accomplishment in Natural Language Processing because it allows us to differentiate language use cases in a new way that would seem counterintuitive to a computer, but which is part of everyday life for *all* humans. Verbal tone and body language clues are very helpful in identifying sarcasm, but necessarily get lost in the transcription of the text. One example of this is shown in the following example about two friends looking to go hang out at the movies:

> **Friend:** I'm waiting at the front. Movie starts in 5.
>
> **You:** I'm on my way now. Should be there in 10.
>
> **Friend:** I'm glad you were watching the clock today.

Depending on the kind of relationship you have with your friend, this could be interpreted as a genuine, sincere gladness or a scathing sense of impatience that your friend has for you because you are a chronically late person.

Some other challenges from this topic include distinguishing literal meaning from the intended meaning. This is not always clear in the text, and there will need to be further investigations into this domain to better distinguish the two. Standard categorization methods struggle with distinguishing them. There is also additional complexity introduced for conclusions that take in a multitude of features in order to establish a sarcastic conclusion. For example, comparing one user's history of content meaning when considering a single piece of content can be helpful (would a known "Bernie bro" honestly say that they "love President Trump"?), but also introduces major complexity and extra steps to such an algorithm.

Background

Sarcasm is crucial to understand for anyone in the business of moderation, interpretation, and communication. The dataset we used first is scraped from Reddit, a website where content moderation is used to prevent hate, violence, the spread of false information and spam, impersonation, invasion of privacy, and certain transactions, but where these activities continue to occur[4].

[4] Content Policy - Reddit. www.redditinc.com, https://www.redditinc.com/policies/content-policy.

Understanding how we can detect sarcasm will allow moderators to distinguish these activities from jokes about them. For example, someone talking about conspiracy theories may be countered by a sarcastic comment, which, without context, could be considered as a conspiracy theory, but would actually be sarcastic, pointing out how ridiculous the original comment was.

Approach and Data

Our approach to this challenge was focused on quantity and diversity. The dataset we used first tested was scraped was from Reddit, using the "/s" handle, and contained over a million rows of data, representing a huge variety of sarcasm and non-sarcasm samples[5] [6]. Our belief was that with more unique examples of sarcasm an algorithm would be better prepared for encounters with novel data.

The Reddit dataset also contains basic information about the user who posted the text, the number of 'upvotes' and 'downvotes' it received from other users, and the text to which the sarcasm is responding, otherwise known as the parent comment.

Later, we would focus our efforts on the quality and cleanliness of data. Our second dataset is much cleaner, better formatted, more consistent and uniform in its presentation, and better-vetted than the first dataset[7]. This data comes from news headlines from sources like *The Onion*, a known producer of sarcastic text, which contrasts with the unpredictable comments produced by individual users of Reddit. However, one caveat of this dataset was that there were far fewer rows of data with only 28,619 rows. The news article data contains both news headlines and a binary classification of sarcasm that was pre-labeled. Some examples of the data can be found in Appendix II. An exploratory look of the Reddit dataset shows that there is not a significant bias towards either the individual user that posts or the word count of the comment.

Code: https://github.com/percivalchen/Sarcasm-Detection-With-NLP/blob/master/EDA.ipynb

Modeling

Our initial plan was to use both sentiment analysis and neural networks to attempt to detect sarcasm. As we progressed with the code, we found that sentiment analysis was not ideal for the scope of this project, as it would require another level of vetting on our end that we felt was not appropriate for this context.

[5] 1 Million Reddit Comments from 40 Subreddits. www.kaggle.com, https://kaggle.com/smagnan/1-million-reddit-comments-from-40-subreddits.

[6] "A Large Self-Annotated Corpus for Sarcasm." https://arxiv.org/abs/1704.05579.

[7] Misra, Rishabh. Rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection. 2018. 2020. GitHub, https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection.

We began the modeling process with the Reddit data, working to clean out null values and group the data into training and testing modules. We decided to remove very small amounts of data from the set that proved to be disruptive and nonsensical. There were only a few null or unusable values in the data, cutting just 53 rows of original 1,010,826 rows of data. We then reduced each word in the dataset to its word stem (otherwise known as stemming the data) using PorterStemmer, transformed it into meaningful numerical representations using TfidfVectorizer, and then modeled it using Linear Support Vector Classifier, Gaussian Naive Bayes, Logistic Regression, and Random Forest models. The results of each model can be found in Appendix III.

After evaluating the results, we were curious to see if our models depended on the quality of the dataset, as they were only slightly better than a random guessing approach. Re-evaluating the data, we decided to find another dataset to use for modeling and then worked with the news headline data. Again, we cleaned, stemmed, transformed, and modeled with the same models as above on this data, and achieved a much higher F1-scores compared to the Reddit dataset. The results of each of these models can be found in Appendix III.

Knowing that we had viable data, we then moved on to neural networks to evaluate sarcasm, utilizing Theano and Keras Python libraries. We eventually settled on using a convoluted neural network to analyze the data for a number of reasons. The primary reason was that it was a much faster way of analyzing the model than other neural nets. The other reason was that we repeatedly had issues where our kernel would crash in the middle of other neural net models. We used two hidden layers for the CNN and optimized the epoch while avoiding the danger of overfitting the data.

One difficult part of the process was completing the analysis on a suitable subset of the data. Even with the CNN, we experienced issues of repeated erroring out of models as our local machine struggled to keep up with the analysis that we were attempting to conduct. We ended up settling with one-tenth of the original size of the new headline dataset, or 28,619 rows, in order to maintain consistency of the dataset and to conserve computing power. This issue also contributed to our selection of CNNs as our primary model, as other algorithms such as BERT would have exceeded our computers' abilities.

Code: https://github.com/percivalchen/Sarcasm-Detection-With-NLP/blob/master/ML.ipynb
Code: https://github.com/percivalchen/Sarcasm-Detection-With-NLP/blob/master/NeuralNets.ipynb

Results
We were able to achieve up to an F1-score of around 0.8302 on the news headline data using logistic regression, with other models and all the Reddit results scoring below that. For the

Reddit dataset, the highest F1-score that we achieved was 0.6929 using CNNs. The numerical scores of all final models can be found in Appendix III.

While implementing CNNs, we mainly tweaked the epochs. Changing other variables, like filter layers, did not seem to have as large of an effect on the resulting probability. Too many epochs led to the overfitting of the data. We found that the model did best with two epochs.

One interesting thing is that there is not really one model that vastly outperforms the others. All of the models turned out to be similar in their predictive capabilities, with the notable exception of the CNN used on the news headlines dataset, which underperformed compared to the other models that we tested it against.

Conclusions

What's interesting about the results is that the scores are much lower on the whole for the Reddit dataset compared to the news headlines dataset. One conjecture that we have for this difference is because of the quality of data. In particular, we noticed that people did not tend to use the correct grammar and wordings for regular phrases in the Reddit dataset. We know that the difference is not a result of the sheer number of comments tested because we were forced to limit the sample size due to computing restraints. Another reason could be because the Reddit dataset contained data that could be conflicting to the model. For instance, we found that the phrase "good for you!" was labeled with both non-sarcastic and sarcastic labels at different points in the dataset. One future improvement could be to scrape the parent thread where the comment originated and try to set up some method to analyze the sentiment of the parent thread(s) to further give context to similar phrases that could be sarcastic given its context.

Some applications of our results include topics such as content moderation and speech-to-text and text-to-speech technologies. Content moderators on sites like Reddit may find the efficient detection of sarcasm helpful in the moderation process; the automation of deciding whether a comment is sarcastic or serious would allow human moderators to focus their time and energy on more borderline or serious cases.

Speech-to-text technologies could be improved with the ability of computers to annotate text with indications of sarcasm. Sarcasm is often lost when spoken words are converted to text, therefore the meaning of the content is also lost. To improve this technology, sarcasm recognition (aided by other cues such as tone and inflection) could be utilized to add annotation, similar to the Reddit "/r" tag, to input text. Similarly, the current abilities of digital voices to convey sarcasm when reading the text aloud are limited to nonexistent. The detection of sarcasm in novel texts could allow developers of digital voices to add tones and inflections to their voices to help a listener understand the true meaning.

Some next steps in our project include improving our models by experimenting with epoch number and other specifications of the deep learning models. Potential other routes we could take include creating and utilizing better data; the irregularities of the Reddit dataset were detrimental to our success. Finally, building a model that could take into consideration other factors, such as a commenter's comment history or other comments in a thread, may be useful.

Appendix I

Sources

References

1. Tungthamthiti, Piyoros, et al. “Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches.” https://www.aclweb.org/anthology/Y14-1047.pdf
2. Poria, Soujanya, et al. “A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks.” ArXiv:1610.08815 [Cs], July 2017. arXiv.org, http://arxiv.org/abs/1610.08815.
3. A. D. Dave and N. P. Desai, "A comprehensive study of classification techniques for sarcasm detection on textual data," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 1985-1991.
4. Content Policy - Reddit. www.redditinc.com, https://www.redditinc.com/policies/content-policy.
5. 1 Million Reddit Comments from 40 Subreddits. www.kaggle.com, https://kaggle.com/smagnan/1-million-reddit-comments-from-40-subreddits.
6. "A Large Self-Annotated Corpus for Sarcasm." https://arxiv.org/abs/1704.05579.
7. Misra, Rishabh. Rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection. 2018. 2020. GitHub, https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection.

Other Sources

- Elena Filatova, "Irony and Sarcasm.” http://www.lrec-conf.org/proceedings/lrec2012/pdf/661_Paper.pdf.
- M. S. M. Suhaimin, M. H. A. Hijazi, R. Alfred and F. Coenen, "Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts," 2017 8th International Conference on Information Technology (ICIT), Amman, 2017, pp. 703-709.
- Zhang, Meishan, et al. "Tweet Sarcasm Detection Using Deep Neural Network - ACL." 11 Dec. 2016, https://www.aclweb.org/anthology/C16-1231.pdf.
- Schifanella, Rossano, et al. “Detecting Sarcasm in Multimodal Social Platforms.” Proceedings of the 24th ACM International Conference on Multimedia, Association for Computing Machinery, 2016, pp. 1136–1145. ACM Digital Library, doi:10.1145/2964284.2964321.

Appendix II

Reddit Comments Dataset

Sample	Label	Comment	Parent Comment
1	0	I think Arcanine would have a decent shot at it with Bite or Fire Fang and Fire Blast.	The issue is being unable to train as the CP difference is much too great without a perfect match up type wise and perfect dodging. Find me a pokemon under 1110 that can train against a 1700 Executor.
2	0	They were underdogs earlier today, but since Gronk's announcement this afternoon, the Vegas line has moved to patriots -1	They're favored to win.
3	0	I couldn't even breathe for a few seconds after reading this because I was laughing so hard.	How does Yui learn guitar so fast? She can barely breathe without Ui's help.
4	0	We didn't say they were sober fans.	... you have fans that say that?
5	1	Rosie O'Donnell, she deserved it, everyone agrees...	Yep, exactly. And the thing about Trump, he never lets go of grudges... No matter how much it hurts him, no matter how much it backfires, no matter how petty it makes him come off. He must respond, no matter what.
6	1	Username checks out.	"It's not less serious, it's one of the bigger tournaments. It just has a different atmosphere about it, unlike the masters. Source:work in golf"
7	1	'Yea, just let the banks fail and America would be fine and the financial crisis would be averted!'	"Well our government is heavily involved in banking.

News Headlines Dataset

Sample	Label	Comment
1	0	the nfl should provide an exemption for medical marijuana
2	0	democrats and republicans agree more than you'd think about kim davis and abortion rights
3	1	'he's a stockbroker,' says woman who finds that exciting
4	1	horrible facebook algorithm accident results in exposure to new ideas

Appendix III

Reddit Comments Dataset Scores (Machine Learning)

Dataset	Model	Precision	Recall	F1-score
Reddit	LSVC	66%	66%	0.6554
Reddit	GNB	58%	58%	0.5770
Reddit	LR	66%	66%	0.6622
Reddit	RFC	64%	64%	0.6404

News Headlines Dataset Scores (Machine Learning)

Dataset	Model	Precision	Recall	F1-score
News	LSVC	82%	82%	0.8218
News	GNB	72%	72%	0.7170
News	LR	83%	83%	0.8302
News	RFC	78%	78%	0.7771

Reddit Comments Dataset Scores (Deep Learning)

Dataset	Model	Precision	Recall	F1-score
Reddit	CNN	72%	67%	0.6931

News Headlines Dataset Scores (Deep Learning)

Dataset	Model	Precision	Recall	F1-score
News	CNN	74%	70%	0.7226