Sarcasm Detection

Selin Akkaya Sophia Osborne Emily Roest
McGill University, COMP 550

Abstract

Sarcasm can be difficult to detect in regular conversion, and even more by text without verbal, nonverbal and contextual clues. Sentiment analysis in NLP deals with this challenge, but not always perfectly. This paper aims to test which features are most indicative of sarcasm by training our model on the Self-Annotated Reddit Corpus (SARC) containing the \s (sarcasm) tag. Our findings highlight the importance of sentiment scores for sarcasm and suggest that sarcasm can be detected without relying on n-gram matrices providing a more efficient approach to sarcasm detection.

Introduction

Sarcasm is defined as a linguistic device used to convey "the opposite of what is actually spoken, especially in order to criticize or insult someone, show irritation, or be funny" (Payne, 2024). Unless explicitly pointed out, this can be challenging to detect in text or speech with no previous exposure to the device, as it requires fundamental engagement with verbal, non-verbal, and contextual clues that are often refined through repeated encounters. Removing auditory tonality and pitch, detection of written sarcasm makes this task all the more challenging. While emoticons in informal language aid in conveying emotion, features like Twitter's 'Tone Indicators' help readers interpret desired meanings of messages. This serves as a form of textual paralinguistic cues in a text-only environment, especially helpful on such platforms that house sarcastic comments that can be taken out of context (Christanti et al., 2022, p. 6). To indicate the presence of sarcasm using these tone indicators, one simply has to write "\s" at the end of their text, denoting a self-reported instance of sarcasm. Given pre-tagged and self-reported sarcastic and non-sarcastic reddit threads from Princeton's NLP Group, this project aims to decipher text features that are most indicative of sarcasm, most notably, individual entity sentiment, contrasting sentiment with parent threads, presence and use of emoticons, punctuation, capitalization, and presence of common hyperbolic phrases.

Related Work

Sarcasm detection in the field of NLP is far from a novelty. Experiments aimed at creating detection models often rely on corpora rooted in comments on various social media platforms tagged with #sarcasm to decipher optimal machine learning methods. For instance, Bamman & Smith (2015) , Joshi et al. (2015), and Riloff et al. (2013) have all worked on Twitter, differentiating supervised training sets with #sarcasm and have used the methods of binary logistic regression, LibSVM with RBF kernel, and bootstrapping algorithm respectively. This paper is more in line with the work of Kumar et al. (2020) with the use of the Self-Annotated Reddit Corpus (SARC) whereby the authors detect sarcastic threads through implementation of a MHA-BiLSTM network. This study builds on prior work by investigating non-reliance on n-grams to train the classifier.

The code is accessible at https://github.com/fiaosborne/550_final_project.git

Methods

Data

This project employs the use of the SARC with over a million entries. The split of sarcastic and non-sarcastic comments is almost equal, with 505,413 comments denoted as containing sarcasm and 505,412 without (Khodak et al., 2018). Further, the dataset indicates both a labeled target comment and its respective parent comment, providing necessary context to confirm sarcastic categorization.

Features

Punctuation, capitalization, emoticons:

Pragmatic features are used to train the model on sarcasm detection. Those selected are punctuation, capitalization, and emoticons. These were selected as several previous studies have proven the existence of nonverbal cues in text-based communication. Lea and Spears (1992) stated that the use of paralinguistic signs such as ellipsis, comma, quotation marks and exclamation mark can affect the perception of the message recipient in CMC" (Christanti et al., 2022, p. 7). For sarcasm detection specifically, Majumdar et al. (2022) have made a list of elements of sarcasm to be noted within a given text. Those selected for the use of our model are: all caps words that exaggerate a sentiment (ex: GOOD, BAD), overused cases of punctuation marks (['!!', '!?', '??', '?!', '...']), positive emoticons ([':)', ':D', '=)', ';)', ':-)', ';-)', ':-D', '=D', ';P', '=]', '8)', '(:']), and negative emoticons ([":(", ":/", ":‘)", ":‘(", ":-(", "D:", ";(", ":-/", ":|"])

Sentiment Analysis:

One of the hallmarks of sarcasm is sentence sentiment. In order to effectively capture this in pre-processing, Scikit-learn's implementation of the Valence Aware Dictionary and sEntement Reasoner (VADER) is used to assign negative, neutral, positive, and compound scores to each of the reddit threads. VADER is specifically designed for texts containing informal language, making it the ideal tool to use to effectively determine sentence sentiment. Sentiment analysis is applied to both individual comments as well as contrasted against parent comments to explore contextual differences.

Machine Learning Implementation:

This project implements a binary classifier to determine if text is sarcastic using the features discussed above. The python automated machine learning library automl-jar is used to implement the classifier. This compares five different model frameworks (Random Forest, Neural Network, XGBoost, CatBoost, LightGBM), including doing a hyper parameter search on all models to determine the best possible classification model. Two feature matrices are tested, one which included uni-, bi-, and tri-grams from the training corpus in a TF-IDF matrix as well as metadata on the comments and their parent threads (Feature Matrix 1), and one feature matrix that included just the metadata (Feature Matrix 2). Metadata included sentiment analysis of both the target and parent comment, presence of positive and negative emoticons, presence of

exaggeration punctuation, and presence of hyperbole in the comment and presence of words that are entirely capitalized.

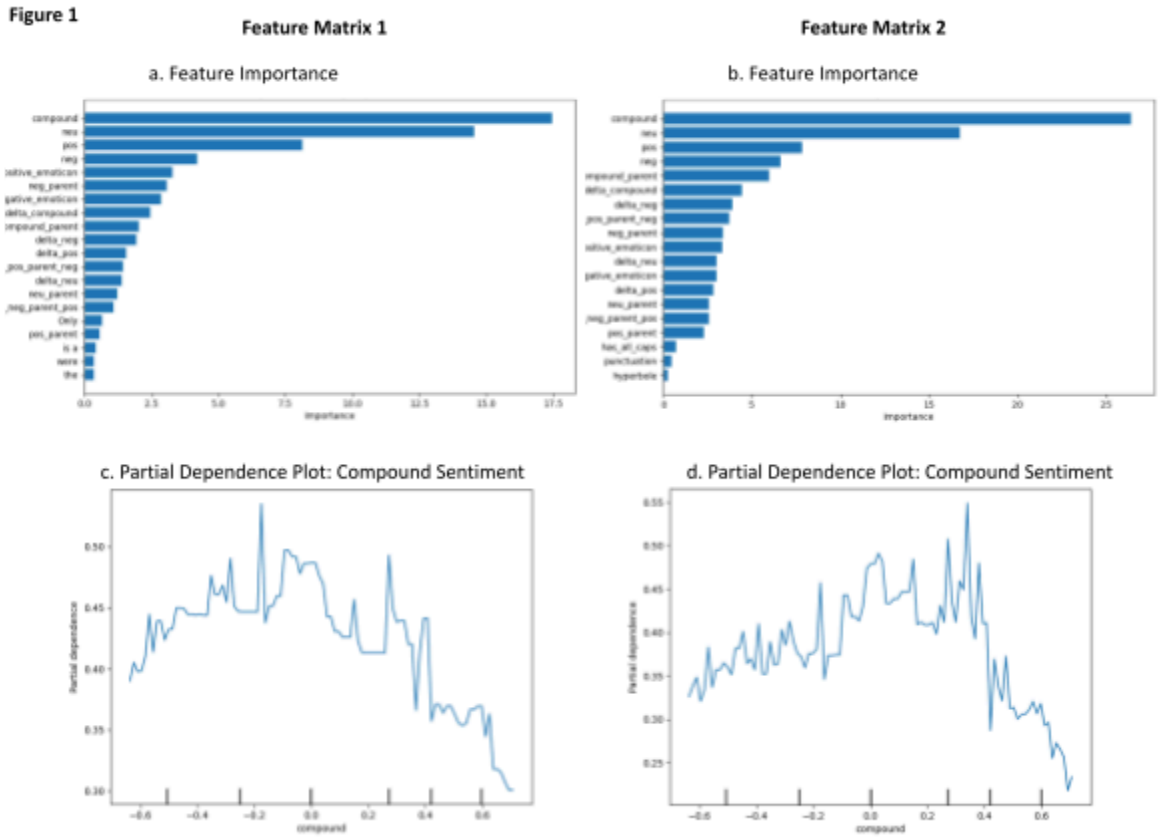
Of all models tested the best performing model was a Catboost Model with a learning rate of 0.2, maximum tree depth of 5 and an rsm of 1. Catboost is a machine learning algorithm that uses gradient boosting on decision trees for classification and regression tasks ("CatBoost."). From the autoML output we were able to visualize which features were the most important predictors in this binary classification task. Using the hyperparameters of the highest performing model as determined by the autoML we re-ran the model using the catboost library directly to generate partial dependence plots that give insight into the directionality of the feature weights as well as capturing the non-linear manner in which feature values impact prediction ("4.1. Partial Dependence and Individual Conditional Expectation Plots.").

Results

Model performance:

There was minimal difference in the model performance when trained on the feature matrix that included both n-grams and metadata and the matrix with just the meta-data. The model trained on Feature Matrix 1 performed slightly worse. For the model trained on metadata only the training accuracy is 0.5803. The test accuracy is 0.5830 and the test AUROC is 0.6168. For the model that included n-grams training accuracy is 0.5759. The test accuracy is 0.5781 and the AUROC is 0.6119.

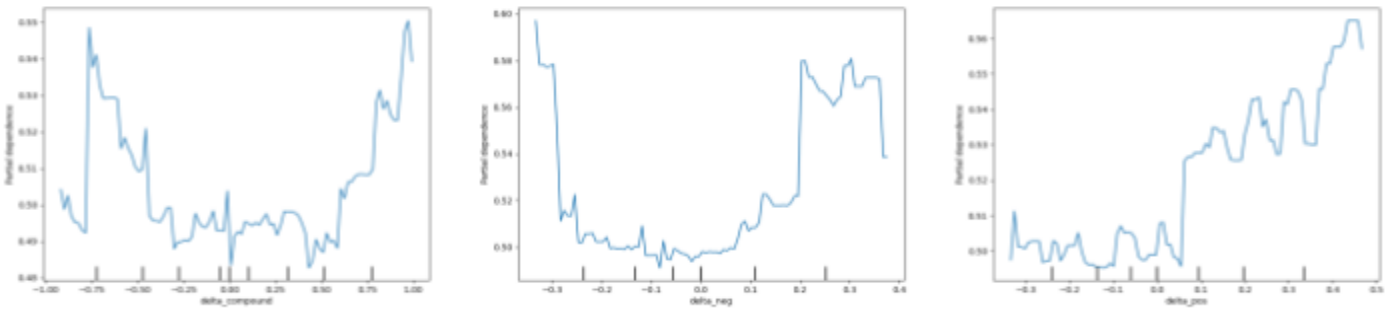
Figure 1



Feature Importance:

Regardless of the training matrix, the top predictors of sarcasm given in the predictor are the sentiment scores on the target string, with the most important polarity being the compound sentiment score (Figure 1a,b). The metric used to evaluate model importance is sk-learn permutation feature importance ("4.2. Permutation Feature Importance."). As seen in Figure 1c,d in the partial dependence plots, there is a non-linear relationship between an increase in compound sentiment score and likelihood of the comment being sarcastic. In the plots for both matrices, there is a positive trend from the most negative values until approximately the zero mark. After this there appears to be a downwards trend from zero to the most positive compound scores. This indicates that comments with a compound comment score closer to zero are more likely to be predicted to be sarcastic as compared to those with a more negative or positive score. While the VADER compound sentiment score was the most important feature in this model, other important features included the compound score of the parent comment as well as delta_compound. From the partial dependence plots looking at the 'delta' features, which are defined as the difference between the sentiment score of the parent and target comment, a higher discrepancy between the sentiment score of the parent and target thread is more predictive of sarcasm (Fig 2). Additionally, it was discerned that a comment having a word in all caps, presence of hyperbole, presence of punctuation increase likelihood of the model predicting the comment to be sarcastic. However from this data set, presence of a positive or negative emoticon increases the likelihood of the comment being predicted as being non-sarcastic.

Figure 2: Partial Dependence Plots



Discussion

A key take away from these results is that when it comes to sarcasm detection techniques, the inclusion of n-grams is not strictly necessary for competitive performance. To build a classifier that is not reliant on a particular corpus of n-grams is helpful as it reduces the computational resources required to perform the task. While we used a Reddit based data set for our training, the functions used to build our feature matrix from this data set could be easily applied to other textual data.

Based on the partial dependence plot for the top most important features in this classifier, sarcastic comments are more likely to take on a neutral tone. This aligns with the idea of sarcasm being a 'dry' humor. Existing research has emphasized the importance of sentiment and

sentiment of the context for sarcasm research (Riloff et al., 2013; Joshi et al., 2015). The findings here reemphasize the importance of sentiment analysis for sarcasm detection among the features commonly used in sarcasm detection literature. The lack of importance on specific n-grams as significant predictors of sarcasm aligns with the theory that sarcasm is identifiable by tone and context rather than a specific pattern of words that can be identified as sarcastic (Kunneman et al., 2015).

Limitations:

While the results suggest that a text classifier can learn important features of sarcasm, there are some limitations to this overall approach. Firstly, the data set used in this study utilized users' self labeling of sarcasm to create the gold standard sarcasm class. Given this, it seems plausible that there are comments within the non-sarcasm class that were intended as sarcasm but were not flagged as such given that the user did not explicitly mark the comment as sarcastic. Additionally, there is a study based on French and Dutch tweets that finds that users use the '\s' sarcasm marker as the extralingual marker of sarcasm and given this do not actually utilize other extralinguistic markers of sarcasm such as capitalizations, punctuations or standard sarcasm structure (Kunneman et al., 2015). Consequently, without the marker the comment would not otherwise be flagged as sarcastic by a reader or machine learning model. Finally, though a broad range of features were used in this sarcasm classifier that were rooted in linguistic theory, this approach does still require assumptions made about what linguistic features would be relevant for this task rather than being truly data driven.

Conclusion

In conclusion, this study demonstrates that the most indicative features of sarcasm are sentiment analysis and pragmatic aspects, and classification can be achieved without relying on high-dimensional n-gram models. This approach presents computational advantages in both space and time complexity, while providing key insights as to the tools to detect sarcasm in informal textual language.

Statement of Contribution

Sophia consolidated the feature matrices and did the machine learning implementation. Selin implemented the pragmatic features (punctuation, capitalization, emoticons). Emily implemented the file reader and sentiment analysis functions. Writing the report was a combined effort with each member focusing on writing sections relevant to their code execution.

Comments:

Unfortunately we were unable to put this document in the appropriate ACL format within the time frame due to time constraints and technical issues.

References (APA)

1. “4.1. Partial Dependence and Individual Conditional Expectation Plots.” *Scikit*, scikit-learn.org/stable/modules/partial_dependence.html. Accessed 18 Dec. 2024.
2. “4.2. Permutation Feature Importance.” *Scikit*, scikit-learn.org/dev/modules/permutation_importance.html. Accessed 18 Dec. 2024.
3. Bamman, D., & Smith, N. (2015). Contextualized Sarcasm Detection on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, *9*(1), Article 1. https://doi.org/10.1609/icwsm.v9i1.14655
4. “CatBoost.” *Documentation*, catboost.ai/docs/en/. Accessed 18 Dec. 2024.
5. Christanti, M. F., Mardani, P. B., & Fadhila, K. A. (2022). Analysing The Meaning Of Tone Indicators By Neurodivergent Community in Twitter. *International Journal of Social Science Research and Review*, *5*(1), Article 1. https://doi.org/10.47814/ijssrr.v5i1.118
6. Joshi, A., Sharma, V., & Bhattacharyya, P. (2015). Harnessing Context Incongruity for Sarcasm Detection. In C. Zong & M. Strube (Eds.), *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)* (pp. 757–762). Association for Computational Linguistics. https://doi.org/10.3115/v1/P15-2124
7. Khodak, M., Saunshi, N., & Vodrahalli, K. (2018). *A Large Self-Annotated Corpus for Sarcasm* (arXiv:1704.05579). arXiv. https://doi.org/10.48550/arXiv.1704.05579
8. Kumar, A., Narapareddy, V. T., Aditya Srikanth, V., Malapati, A., & Neti, L. B. M. (2020). Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. *IEEE Access*, *8*, 6388–6397. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2963630
9. Kunneman, F., Liebrecht, C., van Mulken, M., & van den Bosch, A. (2015). Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management*, *51*(4), 500–509. https://doi.org/10.1016/j.ipm.2014.07.006
10. Majumdar, S., Datta, D., Deyasi, A., Mukherjee, S., & Acharya, A. (2022). Sarcasm Analysis and Mood Retention Using NLP Techniques. *International Journal of Information Retrieval Research*, *12*, 23. https://doi.org/10.4018/IJIRR.289952

11. Payne, L. (2024, November 13). sarcasm. Encyclopedia Britannica. https://www.britannica.com/topic/sarcasm
12. Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013). Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, & S. Bethard (Eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 704–714). Association for Computational Linguistics. https://aclanthology.org/D13-1066
13. Verma, P., Shukla, N., & Shukla, A. P. (2021). Techniques of Sarcasm Detection: A Review. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 968–972. https://doi.org/10.1109/ICACITE51222.2021.9404585