Sarcasm Detection using NLP Techniques

APLN552_01

Koundinya Raghava Nerella
Masters in Data Science
Department of Computer Science
Montclair State University
Montclair, NJ, USA
nerellak1@montclair.edu

Yamini Pathuri
Masters in Data Science
Department of Computer Science
Montclair State University
Montclair, NJ, USA
pathuriy1@montclair.edu

Abstract- Sarcasm detection is a crucial aspect of natural language processing as it affects the performance of various applications such as sentiment analysis, opinion mining, and stance detection. However, current research on sarcasm detection is scattered across different datasets and studies. This paper aims to address this issue by presenting a comprehensive survey of the current state-of-the-art and providing strong baselines for sarcasm detection based on n-grams, logistic regression, and Universal Sentence Encoder pre-trained language models. There are different datasets related to intermediate tasks such as sentiment classification and emotion detection to enhance the model's knowledge of sarcasm, which is often associated with implied negative sentiment and emotions. The experimental results on diverse datasets demonstrate that the Universal Sentence Encoder-based models outperform many previous models, highlighting their effectiveness in detecting sarcasm in natural language text.

Keywords- Sarcasm Detection; n-grams; logistic regression; Universal sentence encoder (USE) model; NLP; machine learning; count vectorizer;

1. Introduction

Sarcasm is a form of communication that conveys the opposite meaning of what is actually said, often using irony, humor, or mockery. Detecting sarcasm is important in many areas, including social media analysis, sentiment analysis, and online customer service. Traditional methods of detecting sarcasm relied on human annotation, which is time-consuming and costly. However, with the advancements in natural language processing, machine learning techniques have become increasingly popular in sarcasm detection.

Among the various machine learning techniques used for sarcasm detection, n-grams, logistic regression, and pretrained models like Universal Sentence Encoder (USE) are commonly employed. N-grams are a type of statistical language model that analyze a sequence of words in a sentence to determine the probability of sarcasm. Logistic regression, on the other hand, is a binary classification algorithm that calculates the probability of a sentence being sarcastic or not based on its features. Pretrained models like USE are neural network models trained on a large corpus of text that can be fine-tuned for a specific task, such as sarcasm detection.

In this era of large-scale online communication, the ability to automatically detect sarcasm is becoming more and more important. With the use of n-grams, logistic regression, and pretrained models like USE, machine learning algorithms are becoming more accurate in detecting sarcasm in text.

2. Related Work

The study presented in this paper [1] involved conducting experiments with various LSTMs and BERT models to detect sarcasm in Discussion Forum data, including both response-only and context-and-response formats. The analysis focused on evaluating the performance of these models, and the findings indicated that utilizing context information can enhance the accuracy of LSTM models for sarcasm detection.

In [2] the transfer learning approach involves utilizing related intermediate tasks such as sentiment classification and emotion detection to enhance the model's knowledge of sarcasm, which is often associated with implied negative sentiment and emotions. The experimental results on diverse datasets demonstrate that the BERT models outperform many previous models, highlighting their effectiveness in detecting sarcasm in natural language text.

This paper [3] addresses the task of multimodal sarcasm detection in tweets that consist of both text and images on Twitter. The proposed approach considers text features, image features, and image attributes as three modalities and utilizes a multi-modal hierarchical fusion model to address this task. The model first extracts image and attribute features, followed by utilizing attribute features and bidirectional LSTM network to extract text features. The features of the three modalities are then reconstructed and fused into a single feature vector for prediction. A multimodal sarcasm detection dataset based on Twitter is also created for evaluation purposes. The results demonstrate the efficacy of the proposed model and the usefulness of the three modalities for detecting sarcasm in multimodal messages.

3. Execution

3.1 Logistic Regression

Logistic regression is a simple yet powerful method for sarcasm detection that can effectively capture the relationship between input features and the probability of sarcasm. It is also computationally efficient and can handle high-dimensional feature spaces.

In logistic regression, the goal is to model the probability that a given input belongs to a particular class, given a set of input features. The model learns the relationship between the input features and the target variable (sarcasm or non-sarcasm) by estimating a set of weights that best fit the training data.

In the context of sarcasm detection, logistic regression can be used to model the relationship between various input features (such as n-grams, sentiment scores, or syntactic patterns) and the probability that a given text is sarcastic.

Once trained, the logistic regression model can be used to predict the sarcasm label of new input texts. The model takes in the numerical feature representation of the input text and computes a probability that the text is sarcastic. If the probability exceeds a certain threshold, the model predicts that the text is sarcastic; otherwise, it predicts that the text is not sarcastic.

3.1.1 TF-IDF Vectorizer

TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer is a feature extraction technique used in natural language processing, including sarcasm detection. It is a statistical measure that evaluates how important a word is to a document in a collection of documents. In sarcasm detection, TF-IDF vectorizer is used to represent the textual data as a numerical vector by assigning a weight to each word in a sentence based on its frequency and importance.

The TF-IDF vectorizer works by calculating two values: term frequency (TF) and inverse document frequency (IDF). The term frequency (TF) calculates the number of times a word appears in a sentence, while the inverse document frequency (IDF) measures the rarity of the word in the corpus. By combining these two measures, the TF-IDF vectorizer assigns a weight to each word that represents its importance in the sentence.

In sarcasm detection, the TF-IDF vectorizer is used to represent the text data as a numerical vector, which can be used as input to machine learning algorithms for classification. The vectorizer converts the text data into a matrix of numerical values that capture the meaning and importance of each word in the sentence, which can be used to train a classifier to predict sarcasm.

Overall, the use of TF-IDF vectorizer in sarcasm detection is an effective way to represent text data numerically, allowing machine learning algorithms to learn from the data and make accurate predictions.

```
Accuracy: 0.7379454926624738
Precision: 0.7505476451259584
Recall: 0.671813725490196
F1-Score: 0.7090015519917227
AUC-ROC Score: 0.8160299883379893
```

Fig 1. Accuracy for logistic regression

```
Sentence:  nature lovers look united, looking them argue over different topics
Prediction:  yes, it is a sarcastic sentence
```

Fig 2. Sarcasm detection using logistic regression

3.1.2 AUC-ROC curve for logistic regression

In sarcasm detection, the AUC-ROC curve is a graphical representation of the model's true positive rate (TPR) versus its false positive rate (FPR) as the decision threshold is varied.

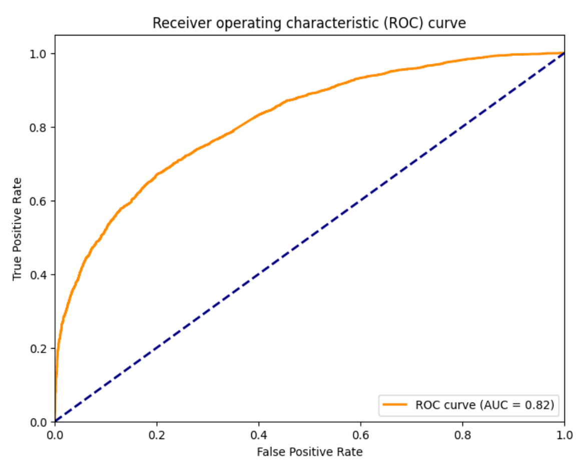

Fig 3. AUC-ROC curve for logistic regression

3.2 n-grams

n-grams are commonly used in sarcasm detection as a feature extraction technique to capture patterns of language use that are typical of sarcastic expressions. n-grams can help detect this gap by analyzing patterns of word usage in a given text. For example, in sarcastic statements, there may be unexpected combinations of words, such as negations or words with opposite meanings. n-grams can be used to extract such patterns and provide a numerical representation of the text that can be used as input to machine learning algorithms for sarcasm detection. We can typically use a combination of different n-grams (such as unigrams, bigrams, and trigrams) and various techniques for feature selection and weighting to improve the accuracy of sarcasm detection models.

Overall, n-grams are a valuable tool in sarcasm detection because they help capture the nuances of language use that are characteristic of sarcastic expressions, and can thus help distinguish between sarcastic and non-sarcastic utterances. We have experimented with the different n-grams ranges as follows:

```
# Define a list of different n-gram ranges to try out
ngram_ranges = [(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)]
```

Fig 4. Different n-grams ranges

And the Accuracy and F-1 score is as follows:

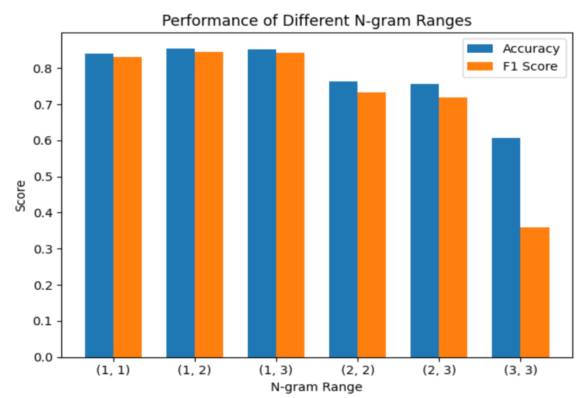

Fig 5. Accuracy and F-1 scores bar chart

3.2.1 CountVectorizer

CountVectorizer is another feature extraction technique used in sarcasm detection. It converts a collection of text documents into a matrix of word counts. It works by counting the frequency of each word in the document and representing it as a numerical vector. In other words, CountVectorizer creates a vocabulary of all unique words in the document corpus and generates a vector for each document that represents the frequency of each word in the vocabulary.

In sarcasm detection, CountVectorizer is often used to create a bag-of-words model, which represents the textual data as a bag of unordered words. This model disregards the order and structure of the sentences, and only considers the frequency of each word in the sentence. By using CountVectorizer to extract features, sarcasm detection algorithms can learn from the frequency and distribution of the words in a sentence.

CountVectorizer is a simple and effective way of representing text data numerically, making it suitable for many machine learning algorithms. However, it may suffer from some limitations, such as the inability to capture semantic relationships between words and the problem of high dimensionality. To overcome these limitations, techniques such as dimensionality reduction and feature selection may be applied.

Overall, CountVectorizer is a useful feature extraction technique in sarcasm detection, providing a numerical representation of the textual data that can be used to train machine learning algorithms.

```
Accuracy: 0.8517353831819241
Precision: 0.8444171779141104
Recall: 0.8433823529411765
F1-Score: 0.8438994481912937
AUC-ROC Score: 0.927233079205939
```

Fig 6. Accuracy for n-grams

3.2.2 AUC-ROC curve for n-grams

The AUC-ROC (Area Under the Curve of the Receiver Operating Characteristic) is a performance metric used to evaluate the effectiveness of a classification model in distinguishing between positive and negative classes, including sarcasm detection.

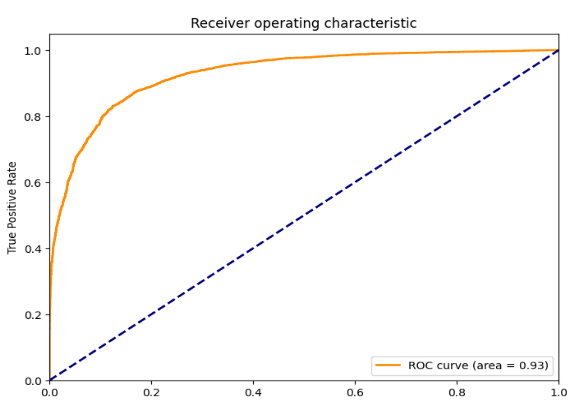

Fig 7. AUC-ROC curve for n-grams

3.3 The Universal Sentence Encoder (USE)

The Universal Sentence Encoder (USE) is a pre-trained model developed by Google that encodes text into fixed-length numerical representations, which can be used as inputs to machine learning models. It has been applied to various natural language processing tasks, including sarcasm detection.

The USE model is trained on a large corpus of text data, making it capable of encoding sentences into high-quality vector representations that capture the meaning and context of the text. Unlike other pre-trained models, the USE model can encode entire sentences or paragraphs, rather than just individual words or phrases. This ability to capture the meaning of a sentence as a whole is particularly useful in sarcasm detection, where the context of the sentence is crucial.

To use the USE model for sarcasm detection, the text data is first preprocessed and then passed through the model, which generates fixed-length vector representations for each sentence. These vector representations can then be used as inputs to machine learning models, such as logistic regression or neural networks, for classification.

The advantage of using the USE model for sarcasm detection is that it requires no additional training, making it easy to use and deploy in real-world applications. Additionally, the model has been shown to outperform other state-of-the-art models in sarcasm detection, achieving high accuracy on various datasets.

In summary, the USE model is a powerful tool for sarcasm detection, providing high-quality vector representations that capture the meaning and context of a sentence. Its ease of use and high performance make it a popular choice for researchers and practitioners in the natural language processing community. Using this pretrained model the accuracy, F-1 score will be:

```
Accuracy:  0.8044491031912415
Precision:  0.7960542540073983
Recall:  0.7911764705882353
F1 score:  0.7936078672403195
AUC-ROC:  0.8871452986432035
```

Fig 8. Accuracy for USE

3.3.1 AUC-ROC curve for USE

The AUC-ROC score ranges from 0 to 1, where 0.5 indicates a random classifier, and 1.0

represents a perfect classifier. A higher AUC-ROC score indicates better model performance in correctly classifying sarcasm instances.

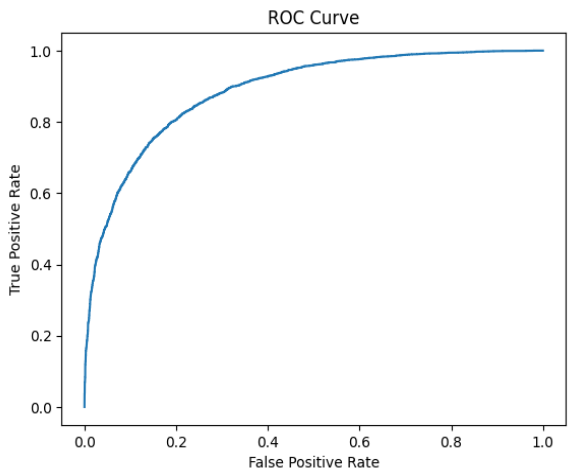

Fig 9. AUC-ROC curve for USE

4. Results

	Accuracy	Precision	Recall	F1-score	AUC-ROC score
n-grams	0.851	0.844	0.843	0.843	0.927
Logistic regression	0.737	0.750	0.671	0.709	0.816
Universal Sentence Encoder	0.804	0.796	0.791	0.793	0.887

Table 1. Results

5. Conclusion

By comparing the techniques logistic regression, n-grams, and pretrained Universal Sentence Encoder (USE) model, n-grams has the highest accuracy while comparing to the other two models.

6. Future Work

Improving model accuracy: While the USE model is a good starting point, there is always room for improvement in terms of model accuracy. We could explore using more advanced machine learning algorithms, such as deep neural networks, to improve performance.

Multi-lingual sarcasm detection: The current model is trained on English data, but sarcasm is prevalent in many other languages as well. We could explore building models for sarcasm detection in other languages.

Real-time detection: In many settings, such as social media, sarcasm detection needs to happen in real-time. We could explore building models that can detect sarcasm in real-time.

Sarcasm generation: While sarcasm detection is an important problem, it's also interesting to explore generating sarcastic text. We could explore building models that can generate sarcastic text, which could have applications in fields like natural language generation and chatbots etc.,

Acknowledgements

Assistance in this project was provided by Dr. Srinivas Bangalore, Department of Computational Linguistics at Montclair State University with assisted materials and guidance through this project. Python programming language was used to implement this project.

References

1. Lydia Xu le and Vera Xu, Sarcasm Detection. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15791781.pdf.

2. Savini, E.; Caragea, C. Intermediate-Task Transfer Learning with BERT for Sarcasm Detection. Mathematics 2022, 10, 844. https://doi.org/10.3390/math10050844.

3. Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2506–2515, Florence, Italy. Association for Computational Linguistics.