Sarcasm Detection In English

Introduction

Sarcasm is a prevalent form of communication in many languages and cultures across the world. Studies suggest that sarcasm is used in day to day conversations in about 1 in every 5 interactions. While the specific expressions of sarcasm may vary across cultures, the underlying concept of using irony or mocking humor to convey meaning remains consistent across different societies. Sarcasm is essentially a way of speaking or writing that says one thing but means the opposite, often in a humorous or mocking way. With the rise of social media platforms, sarcasm has become increasingly prevalent in online communication. However, detecting sarcasm in textual means poses significant challenges due to the absence of vocal cues and contextual information. While sarcasm can be an effective means of humor and social bonding, it can also lead to misunderstandings and conflicts if not interpreted correctly. Studies suggest that misinterpretations of sarcasm can occur in up to 20% of cases. Further, sarcasm in conversation may affect various sentiment analysis tasks to understand people's opinions as the analysis may be biased if people use sarcasm in their statements. Hence, it becomes imperative to appropriately detect sarcasm in real life situations especially in public online forums.

Literature Review

Rathod et al.[1] investigate sarcasm detection in news headlines, departing from the noisy Twitter and Reddit datasets. The work uses neural networks to create sarcasm detectors and study how computers pick up sarcasm's idiomatic patterns. Ghosh et al.[2] redefine sarcasm detection as a word sense disambiguation problem, termed Literal/Sarcastic Sense Disambiguation (LSSD). They tackle two key challenges: selecting target words with dual literal/sarcastic meanings and automatically discerning the word's intended sense in context. Their study explores various distributional semantics methods, ultimately demonstrating that a Support Vector Machines (SVM) classifier is enhanced with word embeddings. Further, in another study Ghosh et al. investigate various Long Short-Term Memory (LSTM) network architectures, emphasizing attention mechanisms. Firstly, they assess the effectiveness of different LSTM configurations in modeling both context and sarcastic responses. Their study sheds light on the significance of contextual information and attention mechanisms in enhancing sarcasm detection performance.

Dataset

The dataset utilized in this study was sourced from the publicly available SemEval 2022. This dataset comprises a substantial collection of tweets, each accompanied by a label indicating whether the tweet is sarcastic or non-sarcastic. Notably, the labels were provided directly by the authors of the tweets themselves. This approach was adopted to mitigate any potential biases or interpretation issues that may arise from using predefined tags or third-party annotators. Moreover, the dataset underwent rigorous verification by linguistic experts to ensure the accurate understanding of the true intent and sarcasm conveyed in the tweets. This meticulous process aimed to enhance the reliability and quality of the dataset for sarcasm detection research.

Methodology

In this section, we outline the methodology adopted for preprocessing the tweet data and selecting the models for sarcasm detection.

Preprocessing

Firstly for the preprocessing of the data we performed normalization on the tokens and the tweets. Normalization of tokens and tweets was essential due to the varied nature of Twitter data, which includes special characters, usernames (@), abbreviations, and informal language usage. This process ensured uniformity and improved the quality of input for the sarcasm detection models. The normalizeTweet function was employed to handle various aspects of normalization, including converting tokens to lowercase, replacing special characters, handling contractions, and standardizing time formats. Additionally, the normalizeToken function was utilized to further normalize individual tokens by converting them to lowercase, replacing specific patterns like usernames and URLs with placeholder tokens, and preserving emojis.

Model Selection

Baseline SVM Model:

A baseline Support Vector Machine (SVM) model was chosen as a traditional machine learning approach for sarcasm detection because of its capacity to effectively learn subtle linguistic cues in high-dimensional feature spaces. By employing SVM as the first model, we aim to establish a robust baseline for sarcasm detection, paving the way for comparison with more complex models while leveraging SVM's strengths in binary classification tasks.

BERTweet-BiLSTM:

The BERTweet-BiLSTM model uses the BERTweet transformer-based model in the encoder layer for contextualized representation learning. The BiLSTM layer acts as the hidden layer, processing the encoded representations from the BERT embeddings layer bidirectionally to capture contextual information. A linear classifier serves as the output layer, taking the concatenated output from the BiLSTM layer and making predictions for the classification task.

BERT-Based Cased and BERTweet with Kim CNN:

These models use Yoon Kim's Kim-CNN architecture, as detailed in the paper "Convolutional Neural Networks for Sentence Classification" by leveraging convolutional layers to capture both local and global context information simultaneously. In the context of sarcasm detection, Kim-CNN's capability to capture linguistic patterns and contextual information makes it a particularly valuable architecture. This model uses 3 convolutional layers (conv1, conv2, conv3) that are instantiated with different kernel sizes to capture different n-gram features from the input text. Further, the CNN layer considers word embeddings as input channels. Lastly a classification layer is used and soft-max probabilities generated. Our model architecture combines the strengths of BERTweet/BERT-based models for contextualized representation learning with the local pattern capturing capabilities of Kim-CNN.

Pretrained Helnivan - English Sarcasm Detector

English Sarcasm Detector is a text classification model built to detect sarcasm from news article titles. It is fine-tuned on bert-base-uncased and the training data consists of a ready-made dataset available on Kaggle.

RoBERTa Base with Dense layers

The model combines the power of pre-trained language embeddings (RoBERTa), LSTM for sequential processing, attention mechanism for focusing on important parts of the sequence, and dense layers for non-linear transformations and dimensionality reduction.

Model Architecture:-

1. AttentionLayer Class: This class defines a custom attention mechanism. Attention mechanisms allow the model to focus on different parts of the input sequence which are necessary to understand the sarcasm related nuances in the data with varying degrees of importance. In this case, it calculates attention weights based on the output of an LSTM layer. The attention weights are computed using a linear layer and then softmax is applied to obtain normalized attention scores. Finally, a context vector is calculated by taking a weighted sum of the LSTM output using the attention scores.
2. The main model consists of a pre-trained RoBERTa model, sets up the hidden size based on the RoBERTa configuration, creates an instance of the custom AttentionLayer, and defines a bidirectional LSTM layer. The dense layers take the output of the attention mechanism, apply non-linear transformations, and reduce the dimensionality of the data.
3. Finally the output of the dense layers is fed into a linear layer which produces the final classification scores. In this case, it's a binary classification task since the output size of the final linear layer is two.

The selected models were chosen based on their ability to effectively capture the nuances of sarcasm in tweet data. By combining pre-trained language models with various architectures such as CNNs and LSTMs, the models can leverage both contextualized representations and local patterns present in tweets. Notably, previous research in sarcasm detection has often overlooked the use of attention mechanisms. However, our work incorporates attention mechanisms especially in the BERTweet-BiLSTM model to focus on relevant parts of the conversation context, thereby enhancing sarcasm detection accuracy. This innovative approach distinguishes our work and contributes to the advancement of sarcasm detection methodologies.

Further, we took up an additional task wherein, given a sarcastic text and its non-sarcastic paraphrased version, the trained model determines the sarcastic one.

Siamise RoBERTa

The model architecture used for this task is a Siamese network with the following components:

1. RoBERTa Base Model: The code uses the pre-trained RoBERTa (Robustly Optimized BERT Pretraining Approach) base model as the backbone for encoding the input text sequences.

2. Dropout Layer: A dropout layer with a drop rate of 0.2 is applied to the output of the RoBERTa model to prevent overfitting.

3. Linear Layer: A linear layer with an input size of 768 (the output size of RoBERTa) and an output size of 1 (for binary classification) is used to map the output of the dropout layer to the final prediction.

4. SarcasmClassifier Class: A custom PyTorch module called `SarcasmClassifier` is defined, which inherits from `nn.Module`. It consists of the RoBERTa model, the dropout layer, and the linear layer mentioned above. The forward pass of this module takes the input IDs and attention masks and returns the logits (output of the linear layer).

5. Siamese Network: The overall architecture is a Siamese network, where two instances of the `SarcasmClassifier` are used to encode the more toxic and less toxic input sequences separately. The outputs of these two instances are then passed to a margin ranking loss function (`nn.MarginRankingLoss`) for training.

The model is trained using a margin ranking loss with a margin of 0.5, which encourages the model to assign a higher score to the more toxic sequence compared to the less toxic sequence.

A Siamese network is a type of neural network architecture particularly useful for tasks involving similarity or distance comparisons between two comparable data inputs. One of the tasks, Siamese networks have been successfully applied to is **Duplicate Question Detection**, which is similar to our task and motivated us to use it.

Results

Model	**Accuracy**	**F1-Score**
SVM	**0.619**	**0.53**
Pre-trained Helnivan (English Sarcasm Detector)	**0.793**	**0.46**
BertTweet with BiLSTM	**0.766**	**0.43**
Bert-Based-Uncased with KIM-CNN	**0.723**	**0.63**
Roberta Base with Dense layers & Attention	**0.773**	**0.48**

Classification Reports

SVM

SVM
Accuracy: 0.619
Precision: 0.624
Recall: 0.619
F_score: 0.621

	precision	recall	f1-score	support
0	0.74	0.73	0.73	754
1	0.32	0.33	0.32	287
accuracy			0.62	1041
macro avg	0.53	0.53	0.53	1041
weighted avg	0.62	0.62	0.62	1041

Helnivan

Accuracy: 0.793
Precision: 0.737
Recall: 0.793
F_score: 0.763

	precision	recall	f1-score	support
0	0.85	0.92	0.88	1200
1	0.06	0.03	0.04	200
accuracy			0.79	1400
macro avg	0.45	0.48	0.46	1400
weighted avg	0.74	0.79	0.76	1400

(0.7928571428571428, 0.7374370395307586, 0.7928571428571428, 0.7633111219344445)

BertTweet with BiLSTM

Accuracy: 0.766
Precision: 0.587
Recall: 0.766
F_score: 0.665

	precision	recall	f1-score	support
0	0.77	1.00	0.87	531
1	0.00	0.00	0.00	162
accuracy			0.77	693
macro avg	0.38	0.50	0.43	693
weighted avg	0.59	0.77	0.66	693

Bert with KIM-CNN

Accuracy: 0.723
Precision: 0.739
Recall: 0.723
F_score: 0.730

	precision	recall	f1-score	support
0	0.84	0.80	0.82	532
1	0.42	0.49	0.45	162
accuracy			0.72	694
macro avg	0.63	0.64	0.63	694
weighted avg	0.74	0.72	0.73	694

Siamese Roberta

Accuracy: 0.741
Precision: 0.819
Recall: 0.741
F_score: 0.725

	precision	recall	f1-score	support
0	0.98	0.49	0.66	174
1	0.66	0.99	0.79	174
accuracy			0.74	348
macro avg	0.82	0.74	0.72	348
weighted avg	0.82	0.74	0.72	348

Roberta Base with Attention

Accuracy: 0.773
Precision: 0.747
Recall: 0.773
F_score: 0.760

	precision	recall	f1-score	support
0	0.85	0.89	0.87	1200
1	0.11	0.09	0.10	200
accuracy			0.77	1400
macro avg	0.48	0.49	0.48	1400
weighted avg	0.75	0.77	0.76	1400

(0.7728571428571429, 0.7474334875650666, 0.7728571428571429, 0.7595970206264323)

Analysis

Task 1: Detection of sarcasm and it's binary classification

(i) We have applied various models for the Binary classification task of sarcasm detection in sentences , to name a few we have used SVM , Pretrained models such Bert-Tweet with Bi-LISTM layers , Bert-base uncased with Kim_CNN , Roberta with dense layers

(ii) Now coming to the performance analysis of the models for the given task

(a) SVM gavc us a accuracy 0.619, among the ML model approaches used by us , but since this wasn't the most optimal , so we resorted to deep learning approaches as mentioned above

(b) Coming to the BERT based approaches , we saw a decent improvement in the performance in terms of accuracy and other metrics , BERT-tweet , and BERT-based-uncased gave us an accuracy of (0.76, 0.72) respectively , and topping the charts was ROBERTA with dense layers with an accuracy (0.79)

(c) Another mention is that of Helnivan , which is a pre trained sarcasm detection model , which is used as a baseline , for transformer based models in sarcasm detection

(iii) Overall we can draw an analysis that Transformer based models are effective in capturing long range dependencies and contextual information , and informational and linguistic cues in various formats which are required for a task like sarcasm detection . The BERT based approaches we have used , at the core of it leverages transformer based architectures , which was the reason of the increase in accuracy compared to the ML based approaches

Task 2: Determine the sarcastic text between a sarcastic text and its non-sarcastic paraphrase

The overall accuracy of 0.741 can be credited to the use of the RoBERTa base model that can effectively capture semantic and contextual information from text, providing a strong foundation for the classification task.

The high recall of 0.99 for the sarcastic class suggests that the Siamese network architecture, with its two parallel branches processing the sarcastic and non-sarcastic text sequences, is effective in learning the distinguishing features of sarcastic texts.

The F1-score of 0.79 for the sarcastic class, while reasonably good, leaves room for improvement. This could be due to the inherent trade-off between precision and recall in the architecture and training approach.

Conclusion and Future Work

(i) Sarcasm detection is an emerging field and work is being done extensively in this field , for sure we can say that transformer based architectures , and fine tuned variants of it are the way forward and will show promising results

(ii) Transformer-based architecture, popularized by models like BERT and GPT, relies on self-attention mechanisms to capture contextual relationships in input sequences, enabling effective modeling of long-range dependencies in tasks such as natural language processing and generation. It employs stacked self-attention layers and feed-forward neural networks, allowing parallel computation and facilitating efficient training on large-scale datasets.

(iii) Text only based sarcasm has its own limitations , as sarcasm is also situation dependent , and other cues are involved , at times even humans find it hard to distinguish between sarcastic and non sarcastic sentiments of a sentence

Future work in this field will include multimodal approaches : which would basically along with text , combine audio (tone and loudness) , visual cues , which would improve accuracy quite a bit , and with additional computational power in the future one can add more systems which would help in sarcasm prediction .

References

[1] Rathod, S., & Kataria, A. Sarcasm Detection Using Natural Language Processing. Available at SSRN 4451909. Ghosh, D., Guo, W., & Muresan, S. (2015, September).

[2]Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In proceedings of the 2015 conference on empirical methods in natural language processing (pp. 1003-1012). Ghosh, D., Fabbri, A. R., & Muresan, S. (2017).

[3]The role of conversation context for sarcasm detection in online interactions. arXiv preprint arXiv:1707.06226.

[4]Chen, Y. (2015). Convolutional neural network for sentence classification (Master's thesis, University of Waterloo).

Project Github Link: https://github.com/hardik21390/Intended-Sarcasm-Detection-in-English.git
Drive Link for Model Checkpoints:
https://drive.google.com/drive/folders/1i57GnQ8Q2izMdpI-t4RgHhj1Pv1CCHT2?usp=sharing