IJHSR
ijhighschoolresearch.org
■ RESEARCH ARTICLE

A Hybrid Approach To Solving The Sarcasm Detection Problem: BERT vs GPT-BERT

Ayush Jayanth Dattatri Yajaman

Arthur Phillip High School, 102-116 Macquarie St, Parramatta, New South Wales, 2150, Australia; ayush.yajaman@gmail.com

ABSTRACT: With the boom of Natural Language Processing (NLP) machines like ChatGPT in recent years, the demand for accurate sarcasm detection has surged. However, to the best of our knowledge, there is not an existing approach that addresses the three main challenges of sarcasm detection simultaneously: understanding context, implicit references, and common sense. Hence, we propose a novel Deep Learning (DL) based hybrid approach to tackle these challenges. We utilize Generative Pre-Trained Transformer-3.5 Turbo (GPT-3.5 Turbo) for appending additional information in each dataset example if any historical, cultural, or political reference or renowned figure is identified. This modified dataset is used to train Bidirectional Encoder Representations from Transformers (BERT). We will be using a large, well-labelled News-headlines dataset for training and testing. A current study claims to have obtained 91.47% accuracy on the same dataset using BERT. This will be used as the baseline to compare with our hybrid approach (GPT-BERT) and a BERT-based approach with modified hyperparameters. The GPT-BERT approach resulted in greater precision and lower validation loss of 0.94% each, compared to the BERT-based approach. However, it underperformed in terms of accuracy (by 0.6%), recall (by 2.5%), and F1 score (by 0.78%). Nevertheless, both approaches outperformed the baseline.

KEYWORDS: Robotics and Intelligent Machines; Machine Learning; BERT; Sarcasm Detection; Binary Classification.

■ Introduction

Sarcasm is frequently used to allude to an idea indirectly.[1] It is employed for a whole host of reasons including adding humour, emphasizing, and accentuating critiquing. However, machines often misinterpret it because of the juxtaposition in its literal and figurative meaning.[2] Given the increasing number of human-machine interactions in recent years, machines must detect and understand sarcasm seamlessly. Otherwise, inaccurate feedback analysis, misunderstood communications, poor social media monitoring, and incorrect sentiment analysis will become a common norm. The ambiguous nature of sarcasm, its arbitrary correlation with its lexical polarity (the inherent sentiment of words based on their literal connotations) and the need to understand implicit references to understand its true meaning make it hard for machines to learn.[3,4] In this paper, we propose a GPT- BERT approach to address the sarcasm detection problem.

Various rule-based and statistical approaches to solve the sarcasm detection problem have been proposed. Some of them are as follows: Riloff *et al*. proposed a bootstrap algorithm that learns a set of phrases associated with positive sentiment and negative situations and identifies sarcasm based on an overt contrast from the learned phrases in 2013.[5] Later, in 2015, Joshi *et al*. proposed a model that utilizes context incongruity (both implicit and explicit) and inter-sentential incongruity (inclusion of previous posts in a thread) to get broader context.[4] In the same year, Ghosh *et al*. proposed a Literal/Sarcastic Sense Disambiguation task where the true meaning of a particular word in a sarcastic utterance is identified and paired with the literal meaning.[2] Word embeddings in a modified Support Vector Machine (SVM) kernel were used to identify if the target word was used in a sarcastic or literal sense. A challenge for such approaches arises when they encounter nuanced expressions or unfamiliar phrases like in the statement "Well, isn't this the cat's pyjamas—a software update right when I'm on a tight deadline!". To understand the sarcasm in this statement, the model would have to interpret the idiom 'cat's pajamas' correctly and then predict that it has been used in a sarcastic sense. This would require superior contextual understanding lacking in these models. Thus, employing rule- based and statistical algorithms is not optimal for distinguishing between sarcastic and non-sarcastic text.[6] DL-based approaches, on the other hand, are capable of handling such examples better as they can learn from extensive datasets and capture intricacies and nuances with better contextual understanding.

In recent years, DL-based approaches have been in wide use for sarcasm detection and text categorization in general.[7] BERT[8] has been extensively used for detecting sarcasm due to its bidirectional context understanding nature.[9-12] It has produced outstanding accuracy rates and F1 scores on multiple occasions. However, recognizing the limitation of BERT in processing local information in a text classification task, Zheng & Yang propose that a Convolutional Neural Network (CNN) model is integrated into BERT's task-specific layer to better understand more nuanced local patterns and information in the form of the BERT-CNN model.[7] This model marginally outperformed the baseline BERT model. Then, noting that most sarcasm detection models were trained on noisy Twitter and Reddit datasets that were self-authored, training BERT on a clearly labelled, noise- redundant dataset in the form of a

DOI: 10.36838/v6i9.4

news headlines dataset was used in the hope of getting more accurate classifications.[13] The BERT model used in this experiment outperformed Long Short-Term Memory (LSTM), SVM, and CNN models.

In the same year, Li *et al.* proposed to use common knowledge to solve the sarcasm detection problem. The model uses a pre-trained BERT model for encoding, a Commonsense Transformer (COMET) model for knowledge generation, a comparison for knowledge selection, and a knowledge- text integration module that fuses information received and provides output automatically.[6] This model achieved an F1 score improvement of 1 - 3.6%. Our idea to use GPT 3.5 Turbo comes from the spectacular results of Sentiment GPT.[14] Its optimal prompt engineering utilizes a well-crafted prompt that analyses sentiment and elicits explanations from GPT achieving an accuracy of 97.32%.

However, observing that these approaches fail to tackle at least one challenge in comprehending context, implicit references, or common sense, we propose to use a GPT-BERT model. In our approach, GPT modifies the examples in the news headlines dataset by explicitly stating any cultural, historical, political, and popular figure references, which, in a sense, adds a layer of additional context that might be crucial in detecting sarcasm. Then, BERT is trained on this modified dataset and finally, a sigmoid function classifies data as either sarcastic or non-sarcastic. We believe that this uses the general knowledge of GPT and superior contextual understanding of BERT to good effect. Hence, we hypothesize that this approach will outperform the unmodified headlines and baseline approaches as it has a robust historical, political, cultural, and textual understanding; thus, addressing the major problems of sarcasm detection - understanding context, implicit references, and common sense.

Motivation:

Inspired by the superior performance of BERT in sarcasm detection tasks[7,13] compared to other models like the Bidirectional Long Short Term Memory (BiLSTM) and SVM[9], we decided to utilize a pre-trained BERT model for contextual understanding. Due to computational resource limitations, we used the bert-en-uncased model. While experimenting, we recognised that BERT fails to classify input correctly when some 'commonsense' is required to understand the text. Studying the dataset in use led us to believe that the 'commonsense' required to understand news headlines falls under the following categories: understanding prominent figures and their accomplishments and understanding historical, cultural, and political references. Corroborating this finding, Scola & Segura-Bedmar in 2021, noticed that BERT made several False Negatives due to its lack of common-sense understanding.[13] For example, in the sentence: "Jeff Bezos named Amazon employee of the month" the BERT model was unable to figure out that Jeff Bezos is the founder of Amazon – information that is crucial to identifying if the sentence is sarcastic or not. The phenomenal results of GPT-3.5 Turbo in understanding abbreviations and ambiguities[14] led us to believe that it could play an important role in sarcasm detection as well. Hence, we proposed to use GPT-3.5-Turbo API to add commonsense information in parentheses, next to the word (target word) that requires common sense to be understood.

Our work contributes to the field of sarcasm detection and NLP in general in the following ways:

- By leveraging the diverse capabilities of both GPT and BERT, our approach can correctly classify complex statements like, "That's just as peachy as the smooth sailing of the Titanic" which require a higher level of contextual and historical understanding.
- Our model has been trained on a highly desirable dataset with reduced noise levels (no incorrect grammar, no emoticons, and no slang/informal language) and clearly labeled examples so it is more likely to maintain high accuracy levels.
- Our model can be used to reduce miscommunications caused due to the use of sarcasm in various human-machine interaction tasks and in social media monitoring tasks where sarcasm is often employed by users.

The rest of the paper is organized as follows: The 'Methods' section discusses component-specific details, lists the materials used, and gives an overview of how the experiment was conducted. The 'Results and Discussions' section presents the results obtained, analyzes them in-depth, and lists the challenges faced during the course of the experiment. The 'Conclusion' section concludes the paper along with thoughts for future work.

Figure 1: BERT's encoding technique.

Methods

The following are all the methods and materials used to conduct the experiment:

BERT:

Bidirectional Encoder Representation for Transformers, as the name suggests is a stack of identical transformer encoders. Its bidirectional nature (the capacity to understand context from both right-left and left-right) makes it useful in a wide range of NLP tasks including Sentiment Analysis, Text summarization, and paraphrasing. The crux of solving these problems is understanding language and this is what BERT does. It understands language, context, and grammar. It is trained in two phases: Pretraining (for language understanding) and fine-tuning (to downstream the model to perform a certain task).[8] It understands language by training on two unsupervised tasks simultaneously: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). It understands bidirectional context using MLM and it understands cross-sentence context using NSP.[8]

Figure 1 illustrates the encoding technique used by BERT. The classification token (CLS) and separator token (SEP) are used as markers that mark the beginning and the end of an input sequence. More specifically, CLS is used to obtain a

representation of the input and is used as input for the classification layer and SEP is used for separating sentences.[8] A sum of the token embeddings (to capture the meaning of the word), segment embeddings (to represent the sentence number encoded into a vector), and position embedding (to represent the position of the word within that vector) are used as embedding vectors for BERT's input.[15] The segment and position embeddings are used to understand the meaning of the word in that place/context.

In this paper, we utilize a pre-trained base BERT model (bert-en-uncased) due to limited computational resources. Therefore, we will be focusing more on fine-tuning the model to classify text as sarcastic (1) or non-sarcastic (0). Finetuning pre-trained models is relatively a lot faster because model parameters are slightly adjusted and only the output parameters are learned from scratch. The BERT model we used consists of 12 layers, 768 hidden layers, 12 attention heads, and approximately 110 million trainable parameters. The input data was processed using an en-uncased preprocess.

News Headlines Dataset:

Many past papers have used Twitter and Reddit datasets for the sarcasm detection task.[16-18] However, misspelled words are common in social platforms like Twitter and Reddit due to relaxed attitudes towards professionalism, leading to inefficient use of word embeddings. For example, in the tweet "Barley even conscious, talking to my conscious," a model might interpret the spelling mistake (barley) as intended and classify it as sarcastic. Additionally, threads, such as replies to original tweets, require more context (the original tweet) to be understood fully. For example, "@RalphtheBold: But ... football. #sarcasm" doesn't make much sense without the original tweet. Noise is another issue. Some tweets might be mislabelled. This is highly probable as hashtags are self-annotated and what one perceives to be sarcastic might not be sarcastic to another.

Considering the various inconsistencies in those types of datasets, we decided to opt for the news headlines dataset. The News headlines dataset consists of data (news headlines) collected from two different news websites - The Onion and HuffPost. As both sources are newspaper organizations, headlines will be proofread - making the margin of spelling errors extremely thin. Also, the Onion was established to present current affairs in a satirical and sarcastic manner. Hence, the probability of mislabelling is significantly reduced. Furthermore, there is no direct relation between headlines like there is in threads, making utterances require lesser context for the model to understand. Thus, the aforementioned issues pertinent to Twitter and Reddit are effectively tackled.

This dataset approximately has an equivalent number of sarcastic and non-sarcastic headlines making it well-balanced and favorable for comprehensive understanding of language patterns. It has over 28,000 examples and marks all headlines from the Onion as sarcastic and all headlines from HuffPost as non-sarcastic. A link to the article is also given for each headline.

An example of a headline by 'HuffPost' labeled as non-sarcastic: "dem rep. totally nails why congress is falling short on gender, racial equality." Alternatively, an example of a headline by 'The Onion' labeled as sarcastic: "naacp demands less minority representation on upn". The dataset consists of headlines similar to the ones mentioned above. Thus, we can discern that the input data revolves around topics like politics and humanities, both of which potentially require some additional context for the BERT model to understand. This makes the new headlines dataset highly optimal for testing our approach.

GPT:

GPT is a series of Large Language Models (LLM) that is trained on massive amounts of unlabelled data to produce output for prompts provided by a user. GPT models are statistical in nature, meaning that they give output by just predicting the next token that best fits statistically instead of reasoning.[19] Architecturally, they are stacked decoders, meaning that they generate output tokens sequentially. In this paper, we will be using GPT-3.5 Turbo.

We observed that the headlines often used the reader's common knowledge and ability to identify implicit references to give an overarching and concise picture of what the article is about. For example, to understand "Police repeatedly shoot Tim Cook after mistaking iPhone for gun", the model needs to know that Tim Cook is the CEO of Apple. Therefore, we used a well-crafted prompt in GPT-3.5 Turbo Application Programming Interface (API) to explicitly state such information next to the word that needs to be understood to comprehend the true meaning of the sentence. We provided some paradigms for it to follow so that it better understands how to perform the task. Finally, we inputted the headlines in batches of 25 for the model to process. We chose to input 25 examples in each batch because we observed that GPT-3.5 Turbo API was performing well on sets of smaller data. On larger data batches, it used to provide haywire results, often diverging from the instructions provided. Unfortunately, we could not reduce the batch size further due to financial and computational limitations.

As GPT-3.5 Turbo API is a paid service, we made use of the free $5.00 credit upon signup. A secret API key was used to harness the power of GPT-3.5 Turbo API. Appendices 1 and 2 show how we operated with this API.

Hyperparameters:

After experimenting with multiple hyperparameters (configuration settings that govern the way a model is trained), we came to the conclusion that the following are optimal:

i) Adam optimizer for adaptive learning rates

ii) 1e-5 learning rate (the rate at which the model adjusts its weights during training)

iii) 0.2 dropout rate and early dropping to prevent overfitting[20]

iv) sigmoid activation function for optimal binary classification

v) binary cross entropy function for measuring the disparity between true and predicted labels.

Training:

We trained the bert-en-uncased model with a patience of 1 and 2 epochs (the number of times the model goes through the dataset). We kept patience (the number of epochs the model

DOI: 10.36838/v6i9.4

can go over after no improvement in a particular metric being monitored) as 1 because we had only 2 epochs and we wanted to prevent overfitting and train the model efficiently. During training, we were monitoring two metrics – Area Under the Curve (AUC) and accuracy. AUC is a measure of the area under a Receiver Operating Characteristic (ROC) curve where ROC is a True positive rate vs False positive rate graph at different thresholds. Thresholds are cut- off values that determine whether a statement is positive or negative.

As AUC gives a good estimate of the model's binary classification capabilities, we decided to use an early stopping callback based on the validation AUC metric. Finally, a batch size of 16 was used to reduce computational strain.

Materials

Google Collab:

We used Google Collab, a collaborative environment for empirical machine-learning experiments. It is a browser-based host of Jupyter notebooks that gives us access to TPU and GPU runtimes. It runs on a virtual machine (VM) linked to the account being used. We mounted Google Drive and uploaded the news headlines dataset to Google Drive so that we could avoid uploading the file manually to Google Collab every time the runtime got disconnected.

Kaggle:

The bert-en-uncased model and the news headlines dataset were obtained from Kaggle (an online community for ML researchers and data scientists)

Pipeline and Headline Length:

As can be seen from Figure 2, this is how our model flows: GPT-3.5-Turbo API receives a well-crafted prompt and a few paradigms to understand its task. Then, the headlines from the news headlines dataset are fed into it for it to make the necessary modifications. The output of the GPT-3.5-Turbo API is split in a 90% to 10% training-to-validation set ratio and is fed into the BERT model for fine-tuning. The final output classifies the text as either sarcastic or non-sarcastic.

Figure 2: Our GPT-BERT approach flowchart.

The modified news headlines dataset naturally gave us lengthier headlines than the unmodified approach due to the addition of overtly stating references as can be seen in Figure 3. The maximum length of the headline was 153 and 151 respectively. Based on this, we set our sequence length to 160 to avoid truncation and unnecessary padding.

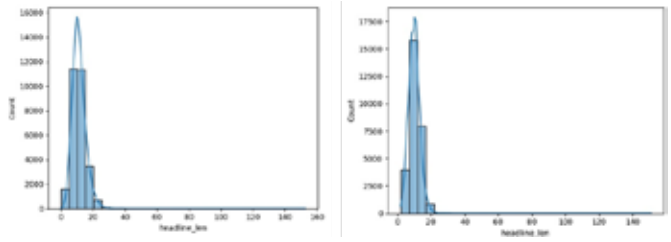

Figure 3: Headline length – modified dataset (left) vs unmodified headline dataset (right).

Results and Discussion

Results:

Both GPT-BERT and BERT approaches yielded results that outperformed the baseline, as can be seen from Table 1. The higher threshold and precision of our approach means that the modified headlines made the model more conservative in predicting sarcastic text. It increased the confidence of positive predictions in doing so, thus minimizing the number of False Positives. This had an inverse effect on recall. This is evident in the considerable disparity of almost 2.5% between our approach and the BERT approach. The effect of this can be visually inferred from the confusion matrix heatmap in Figure 4.

The higher recall yielded by training BERT on an unmodified news headlines dataset shows that an unmodified dataset without explicitly stating historical, cultural, or political references or information about a renowned figure, results in the model being better at capturing positive instances, thus minimizing false negatives. The 0.15 threshold and high recall show that this approach makes the model very sensitive.

The F1 score favors the unmodified approach due to its much higher recall. The BERT approach only marginally outperforms the GPT-BERT approach in terms of accuracy so we cannot infer much from this metric. However, our approach's optimal loss result shows that our approach allows the model to generalize better to unseen data.

Overall, our approach has a higher probability of correctly classifying True Positives as Sarcastic due to its higher precision. However, the significantly lower recall of our model tells us that it is susceptible to making more False Negatives. These characteristics indicate that our approach is better suited for circumstances where misclassifying non-sarcastic comments as sarcastic can be critical. A common task where this can be applicable is company review analysis where often, reviews are literal. If a model classifies serious negative feedback as sarcastic due to its sensitivity (like the BERT approach), it could potentially lead to lower customer return rates. However, it might not be well-suited for situations where capturing true positives takes precedence such as in a chatbot dedicated to having casual conversations with users. Here, the BERT approach would be optimal because people often employ sarcasm in casual interactions.

Table 1: Recall, precision, F1 Score, accuracy, validation loss, hyperparameters for each approach.

Approach	Recall	Precision	F1 Score	Accuracy	Loss	Dataset	Hyperparameters
GPT-BERT	0.9192	**0.9399**	0.9295	0.9336	**0.1860**	Headlines (modified by GPT)	Adam, 1e-5, 2 epochs, 16 batch size
BERT	**0.9442**	0.9305	**0.9373**	**0.9399**	0.1954	Headlines	Adam, 1e-5, 2 epochs, 16 batch size
Scola & Segura-Bedmar (baseline)	0.8938	0.9244	0.9088	0.9147	NA	Headlines	Adam, 2e-6, 10 epochs, 20 batch size

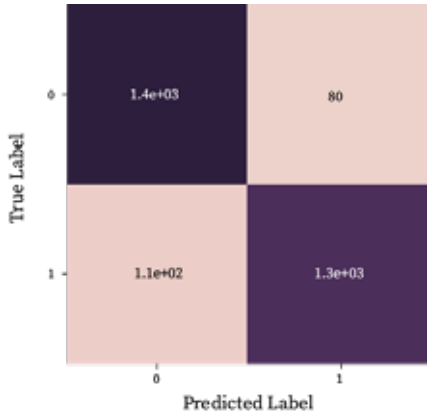

Figure 4: BERT's confusion matrix heatmap on modified headlines dataset.

The hypothesis stated in the introduction, "We hypothesize that this approach will outperform the unmodified headlines and baseline approaches as it has a robust historical, political, cultural, and textual understanding", proved to be only partially correct as our approach, despite outperforming the baseline approach in most aspects, only outperformed the unmodified headlines approach in validation loss retention and precision.

We curated around 160 examples where errors were made and observed the following: Out of the 160 examples, 32 were unique to each approach meaning that both approaches had a considerable overlap. This showed that adding additional context did not necessarily always solve the problem. However, of the 32 unique errors, only 22% of the mistakes made were when the sentence was modified. From this, we can infer that adding additional context does help significantly. Additionally, in the unique false positives made by GPT-BERT, only 11.5% of the False Positives were modified sentences. This corroborates the conservative stance of this approach.

Some False Positives made even after GPT modified the headline are as follows:

- "Donald Trump (former US President) prefers violent football so more black players get hurt: ESPN (sports network) analyst" – The BERT model misclassified this headline as sarcastic whereas it accurately classified "donald trump prefers violent football so more black players get hurt: espn analyst" as non-sarcastic. Considering that the unmodified headline approach led to a much more sensitive BERT model, this is a very curious anomaly. We believe that this misclassification occurred due to the increased complexity caused by the additional information in the parentheses.
- "Elizabeth Warren: 'Donald Trump is a loser' (referring to Elizabeth Warren, a prominent political figure)." – This is an example where GPT inaccurately performs its task. Instead of adding context next to the target word, it adds it at the end of the sentence. This creates contextual incongruity within the sentence and the model might perceive that the 'loser' is Elizabeth Warren due to the text in the parentheses and thus conclude that the text is sarcastic. The original headline, "elizabeth warren: 'donald trump is a loser'", was accurately classified by BERT.

A False Negative made by the GPT-BERT approach includes "Frustrated sycophant (can't figure out what boss wants to hear)" where GPT doesn't adhere to the instruction given again and causes the BERT model to err. Since only linguistic knowledge was required to understand the statement, the addition of parentheses was unnecessary as it added more complexity to the statement. This was probably why BERT correctly classified "frustrated sycophant can't figure out what boss wants to hear" as sarcastic but incorrectly classified "Frustrated sycophant (can't figure out what boss wants to hear)" as non-sarcastic.

As mentioned before in the 'GPT' subsection in 'Methods', we observed that results were significantly more accurate when the number of prompts and tokens inputted to GPT-3.5 Turbo was lesser. Hence, we hypothesize that such anomalies can be reduced by reducing the batch size of batches that are inputted for GPT to process.

Some headlines were misclassified by both approaches - even when the headline was modified by GPT appropriately in the GPT-BERT approach. For example, BERT misclassified 'Madonna (renowned singer) just held a surprise concert in NYC to support Hillary Clinton (politician)' as sarcastic. We believe that the modified headline was misclassified because of a skew towards sarcasm in headlines containing musical and political content caused due to the model being trained on the sarcastic headline, "Bruce Springsteen (musician) concert totally changes area man's mind about voting".

The unmodified headlines approach mostly consisted of errors caused due to a lack of common knowledge. For example, BERT misclassified 'Chuck Schumer warns gop not to change the rules to confirm neil gorsuch' most probably not knowing who Chuck Schumer and Neil Gorsuch are whereas it correctly classified "Chuck Schumer (politician) warns GOP not to change the rules to confirm Neil Gorsuch (judge)" as non-sarcastic.

In the bigger picture, the research presented in this paper can help the AI community advance in fields like Natural Language Processing by allowing machines to comprehend and understand sarcasm more 'humanly'. Such advancements can prove to be valuable in tasks like sentiment analysis, social media monitoring, and review analysis where it is a human's utterance that we are trying to classify. By integrating our approach with a model that is downstreamed to performing the above-mentioned tasks, the system will be less susceptible to misclassifications caused by the use of sarcasm. Moreover, since people generally tend to infuse sarcastic comments with implicit references in their reviews and social media comments, our approach is better suited to classify text in these scenarios. For example, consider the review, "Congratulations on channeling your inner Marie Antionette with the pricing!" A traditional sentiment analysis model might misclassify this statement as positive due to its seemingly positive sentiment. But by integrating our approach, the GPT model will provide appropriate context for 'Marie Antionette' and the BERT model can classify the statement as sarcastic. Subsequently the sentiment analysis model can correctly classify it as negative. This ultimately allows an organization that has set up an auto

DOI: 10.36838/v6i9.4

mated system which sends negative feedback for review to receive even this review, notifying them of their high pricings. Therefore, the findings in this paper hold great significance.

Challenges, Observations, and Limitations:

Some challenges and points we thought were noteworthy whilst analyzing the data and conducting the experiment are as follows:

i) ***Contentious labeling:*** Although using the news headlines data did reduce noise in the form of spelling errors, emoticons, and slang words there were some circumstances where the labeling was questionable. Provided that sarcasm is mostly subjective, some sentences like, "Mom wants to know if you'll be free if she visits 14 months from now", which clearly look like an imperative statement with no satirical nature, were labeled as sarcastic just because it was sourced from The Onion. This is where some amount of noise came into play.

ii) ***Obvious errors made by BERT:*** There were some blaring errors made by the model despite the additional context provided. For example, the headline, "Yes, let's just ignore Trump's (former US President) hateful rhetoric and laugh about the guy in the sweater" was classified as non-sarcastic although it is clearly sarcastic. Ideally, the model should be able to understand this statement without knowing who the 'guy in the sweater' is because it is irrelevant to the sarcasm.

iii) ***Bias:*** Misclassifications caused due to biases influenced from the training data as can be seen from "'Madonna (renowned singer) just held a surprise concert in NYC to support Hillary Clinton (politician)' as sarcastic" in the 'Discussions' subsection.

iv) ***Inconsistent GPT:*** Although GPT mostly adhered strictly to instructions, it was non-compliant when modifying some headlines. This in turn led to BERT misclassifying many headlines as can be seen from "Elizabeth Warren: 'Donald Trump is a loser' (referring to Elizabeth Warren, a prominent political figure)" in the 'Discussions' subsection. Moreover, GPT often made erratic errors when working with large amounts of data in a single prompt. Therefore, we had to compress the input into smaller batches. This limited the number of errors but did not eliminate them.

v) ***Runtime errors:*** While using Google Collab, runtimes were terminated due to the dynamic allocation of runtime in free accounts. As a result, sometimes code had to be re-run from the beginning all over again.

vi) ***Financial restrains:*** The limitations of using a free account to run GPT-3.5 Turbo were significant. On several occasions, limitations such as rate limits impeded our progress. Additionally, the API error calls (due to the limitations) often led to us having to be vigilant and observe the code while it is running so that the code just does not spew out error calls and waste resources.

Conclusion

To the best of our knowledge, this is the first paper to use a GPT-BERT approach to solving the sarcasm detection problem. In our approach, GPT-3.5 Turbo was used to identify and explicitly state any historical, cultural, or political reference or renowned figure and BERT was used for better contextual understanding. The BERT model trained on the modified headlines (GPT-BERT approach) was more conservative, generalized more efficiently to unseen data, and showed higher precision. However, it underperformed in certain metrics like accuracy and F1 score compared to the more sensitive BERT model trained without modifying the news headlines. This meant that our approach holds promise in fields where misclassifying non-sarcastic statements as sarcastic can be critical, such as in company review analysis. Nevertheless, both these approaches outperformed the baseline approach which trained a BERT model on the same news headlines dataset but with different hyperparameters.

Building on these compelling findings, there are numerous avenues for further investigation. Research efforts could focus on mitigating biases obtained from training data. This would mean that output is not incorrectly skewed due to resemblance in the given utterance and an utterance previously trained on. Additionally, superior models of GPT like GPT-4 could be leveraged for optimal dataset modifications. This would greatly minimize errors while adding additional context to an utterance. Finally, exploring multimodal data instead of unimodal data could offer various streams of data to analyze, leading to more accurate sarcasm detection.

Acknowledgment

I would like to thank my mentor, Dr. Eric Sakk, for his continuous guidance and feedback throughout the research project.

References

1. Abu Farha, I.; Magdy, W. From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*; Al-Khalifa, H., Magdy, W., Darwish, K., Elsayed, T., Mubarak, H., Eds.; European Language Resource Association: Marseille, France, 2020; pp 32–39.
2. Ghosh, D.; Guo, W.; Muresan, S. 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*; Màrquez, L., Callison-Burch, C., Su, J., Eds.; Association for Computational Linguistics: Lisbon, Portugal, 2015; pp 1003–1012. https://doi.org/10.18653/v1/D15-1116.
3. Chaudhari, P.; Chandankhede, C. Literature Survey of Sarcasm Detection. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*; 2017; pp 2041–2046. https://doi.org/10.1109/WiSPNET.2017.8300120.
4. Joshi, A.; Sharma, V.; Bhattacharyya, P. Harnessing Context Incongruity for Sarcasm Detection. In *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)*; Zong, C., Strube, M., Eds.; Association for Computational Linguistics: Beijing, China, 2015; pp 757– 762. https://doi.org/10.3115/v1/P15-2124.
5. Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; Huang, R. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*; Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., Bethard, S., Eds.; Association for Computational Linguistics: Seattle, Washington, USA, 2013; pp 704–714.
6. Li, J.; Pan, H.; Lin, Z.; Fu, P.; Wang, W. Sarcasm Detection with Commonsense Knowledge. *IEEEACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3192–3201. https://doi. org/10. 1109/TASLP.

2021.3120601.
7. Zheng, S.; Yang, M. A New Method of Improving BERT for Text Classification. In *Intelligence Science and Big Data Engineering. Big Data and Machine Learning*; Cui, Z., Pan, J., Zhang, S., Xiao, L., Yang, J., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2019; pp 442–452. https://doi.org/10.1007/978-3-030-36204-1_37.
8. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv May 24, 2019. https://doi.org/10.48550/arXiv.1810.04805.
9. Baruah, A.; Das, K.; Barbhuiya, F.; Dey, K. Context-Aware Sarcasm Detection Using BERT. In *Proceedings of the Second Workshop on Figurative Language Processing*; Klebanov, B. B., Shutova, E., Lichtenstein, P., Muresan, S., Wee, C., Feldman, A., Ghosh, D., Eds.; Association for Computational Linguistics: Online, 2020; pp 83–87. https://doi.org/10.18653/v1/2020.figlang-1.12.
10. Khatri, A.; P, P.; M, D. A. K. *Sarcasm Detection in Tweets with BERT and GloVe Embeddings*. arXiv.org. https://arxiv.org/abs/2006.11512v1 (accessed 2024-01-03).
11. Parameswaran, P.; Trotman, A.; Liesaputra, V.; Eyers, D. BERT's The Word : Sarcasm Target Detection Using BERT. In *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*; Rahimi, A., Lane, W., Zuccon, G., Eds.; Australasian Language Technology Association: Online, 2021; pp 185–191.
12. Meng, J.; Zhu, Y.; Sun, S.; Zhao, D. Sarcasm Detection Based on BERT and Attention Mechanism. *Multimed. Tools Appl.* **2023**. https://doi.org/10.1007/s11042-023-16797-6.
13. Scola, E.; Segura-Bedmar, I. Sarcasm Detection with BERT. *Proces. Leng. Nat.* **2021**, No. 67, 13–25. https://doi.org/10.26342/2021-67-1.
14. Kheiri, K.; Karimi, H. SentimentGPT: Exploiting GPT for Advanced Sentiment Analysis and Its Departure from Current Machine Learning. arXiv July 23, 2023. https://doi.org/10.48550/arXiv.2307.10234.
15. Mickus, T.; Paperno, D.; Constant, M.; van Deemter, K. What Do You Mean, BERT? In *Proceedings of the Society for Computation in Linguistics 2020*; Ettinger, A., Jarosz, G., Pater, J., Eds.; Association for Computational Linguistics: New York, New York, 2020; pp 279–290.
16. Gupta, R.; Kumar, J.; Agrawal, H.; Kunal. A Statistical Approach for Sarcasm Detection Using Twitter Data. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*; 2020; pp 633–638. https://doi.org/10.1109/ICICCS48265.2020.9120917.
17. Lemmens, J.; Burtenshaw, B.; Lotfi, E.; Markov, I.; Daelemans, W. Sarcasm Detection Using an Ensemble Approach. In *Proceedings of the Second Workshop on Figurative Language Processing*; Kebanov, B. B., Shutova, E., Lichtenstein, P., Muresan, S., Wee, C., Feldman, A., Ghosh, D., Eds.; Association for Computational Linguistics: Online, 2020; pp 264–269. https://doi.org/10.18653/v1/2020.figlang-1.36.
18. Pawar, N.; Bhingarkar, S. Machine Learning Based Sarcasm Detection on Twitter Data. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*; 2020; pp 957–961. https://doi.org/10.1109/ICCES48766.2020.9137924.
19. Floridi, L.; Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.* **2020**, *30* (4), 681–694. https://doi.org/10.1007/s11023-020-09548-1.
20. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting.

Author

Ayush Jayanth Dattatri Yajaman is a year 11 student studying at Arthur Phillip High School, Sydney, Australia. He is passionate about mathematics and technology and has a keen interest in AI. He looks forward to studying courses at university that would help him pursue a career in AI.

Appendices

Appendix 1:

GPT prompt: "I am a statement processor. If the sentence is understandable with only English knowledge, return it unchanged. Otherwise, add relevant information in brackets '()' next to the word requiring additional context, limited to 7 words. Modifications are exclusive to historical, cultural, political, or renowned figures' knowledge. Information should only be added next to the word, not at the sentence end. In case of error, corrections are made for repeated sets of sentences."

Paradigms given to GPT:

Input: "Jeff Bezos named Amazon employee of the month"
Output: "Jeff Bezos(founder of Amazon) named Amazon employee of the month"
Input2: "leave no person with disabilities behind"
Output2: "leave no person with disabilities behind"

Appendix 2:

```
Pip install openai
pip install --upgrade openai
import time
from openai import OpenAI
api_key = #your secret API key
client = OpenAI(api_key=api_key)
# Insert your file name in file_name placeholder
with open(file_name, 'r') as file:
    headlines = file.readlines()
# Define the step size for indexing. Change 'z' according to your needs. We used 25.
step_size = z
# Iterate over the range of indices with the specified step size. Modify 'x' and 'y' according to your needs.
for start_index in range(x, y, step_size):
end_index = start_index + step_size
user_messages = headlines[start_index:end_index]
conversation = [  {"role" : "system", "content":
                   "Instruction"},
                   {"role" : "user", "content": "Example
                   problem "},
                   {"role" : "assistant", "content":
                   "Solution"},
                   {"role" : "user", "content":
                   "Example problem-2"},
                   {"role" : "assistant", "content":
                   "Solution"},
                   ]
# Add user messages to the conversation
user_message = {"role" : "user", "content" : ' '.join([f'"{i + 1}){statement}"' for i, statement in enumerate(user_messages)])}
conversation.append(user_message)
```


DOI: 10.36838/v6i9.4

```
   # Create a chat completion. Modify a,b,c according to your
needs. We used 2000, 1, and 0 respectively.
try:
   chat_completion= client.chat.completions.create(
       model="gpt-3.5-turbo",
       messages=conversation,
       max_tokens=a,
       n=b,
       temperature=c)
   with open('output-final-mod.txt', 'a', encoding='utf-8') as
   output_file:
   output_file.write('\n'.join([choice.message.content for
   choice in chat_completion.choices]) + '\n')
   # Print the content of all assistant's
   responses for choice in chat_completion.choices:
      print(choice.message.content)
   except Exception as e:
       print(f"Error in API
       call:{e}")
   time.sleep(1)
```