Sarcasm Detection in Product Reviews using Textual Entailment approach

Shubham Sinha
sinhashubham0103@gmail.com
Visvesvaraya National Institute of Technology
Nagpur, Maharashtra, India

Tarun Vijeta
tarunvijeta0104@gmail.com
Visvesvaraya National Institute of Technology
Nagpur, Maharashtra, India

Pratik Kubde
pratikkubde4@gmail.com
Visvesvaraya National Institute of Technology
Nagpur, Maharashtra, India

Ayush Gajbhiye
gajbhiyeayush26@gmail.com
Visvesvaraya National Institute of Technology
Nagpur, Maharashtra, India

Mansi A. Radke
mansi.radke@cse.vnit.ac.in
Assistant Professor, Visvesvaraya National Institute of Technology
Nagpur, Maharashtra, India

Christopher B. Jones
JonesCB2@cardiff.ac.uk
Professor, Cardiff University
Cardiff, Wales, United Kingdom

ABSTRACT

Sarcasm is a form of sentiment characterized by the use of words that express the opposite of what is meant. Sarcasm detection has applications in multiple domains ranging from sentiment analysis in product reviews to user feedback, and online forums. Sarcasm detection is important to understand user opinions and intentions in areas such as sentiment-based classification and opinion mining. This can result in better product development and customer service. Sarcasm detection can be a challenging task because sarcastic sentences may use positive expressions to convey negative meanings or may use negative sentences to convey positive meanings. Also, sarcastic sentences form a very small component of the entire communication. The increasing use of sarcasm in various social media such as Twitter, Reddit, Amazon product reviews, etc. has highlighted the importance of detecting and understanding sarcasm in various contexts. Sarcasm detection is a challenging problem for NLP systems that often rely on statistical models for performing sentiment analysis. In this research, the focus is on the use of a textual entailment approach for detecting sarcasm. Textual entailment is a natural language inference task that involves determining whether one text (hypothesis) can be derived from another text (premise). The underlying assumption behind this approach is that - if there is a contradiction between the premise and hypothesis, we can say that the hypothesis is sarcastic. To test our approach, an annotated corpus of 3000 product reviews was developed methodically from the Amazon Reviews dataset and tested using the textual entailment approach. The proposed approach achieved an F1 score of 0.76 on this dataset. The result is better than the baseline considered which is the BERT binary classifier which gives an F1 score of 0.48 on the same dataset.

NLPIR 2023, December 15–17, 2023, Seoul, Republic of Korea

ACM ISBN 979-8-4007-0922-7/23/12…$15.00
https://doi.org/10.1145/3639233.3639252

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**.

KEYWORDS

Sarcasm Detection, Textual Entailment, Hypothesis, Premise, Natural Language Inference

ACM Reference Format:
Shubham Sinha, Tarun Vijeta, Pratik Kubde, Ayush Gajbhiye, Mansi A. Radke, and Christopher B. Jones. 2023. Sarcasm Detection in Product Reviews using Textual Entailment approach. In *2023 7th International Conference on Natural Language Processing and Information Retrieval (NLPIR 2023), December 15–17, 2023, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3639233.3639252

1 INTRODUCTION

Sarcasm is an ironic statement that is meant to mock some person, product, item, etc or at times to create humour. For example, consider the sentence “Her mobile phone is so good that I won’t recommend it" or “He drove the car very slowly at just 120 km/h". In the second case, the word “slowly" is used in a sarcastic manner. In the above two sentences, we don’t require any context to check whether the sentences are sarcastic or not. However, there are also some sentences that need context to detect sarcasm such as -“Thank you sooooo much for all that you have done on this project" - where we can’t determine whether the above sentence was said by the speaker in order to mock the person or appreciate their efforts. So to detect sarcasm in these type of sentences, we need the context. Sarcasm detection is very important in order to know the correct meaning of the sentence. Nowadays Natural Language Sentence Generation is gaining a lot of popularity as it measures the positivity of words and phrases, which has become more important lately. Sarcasm Detection has a lot of applications in marketing research to understand user opinions, reviews and feedback.

However, sarcasm detection is not an easy task. We need to know the tone of the speakers, their facial expressions, their body language, their intonation, the context, the pitch of their voice and also the perspective of the speaker to be able to detect sarcasm appropriately. However, when it comes to Natural Language Processing (NLP), the absence of facial expressions, body language,

and pitch make the task very challenging. Another challenge in sarcasm detection can be the unavailability of the context in which the sentence occurs. For example, a statement like "He can buy a car" cannot be determined as sarcastic or non-sarcastic unless we know the context about the person's background and financial conditions. However, thanks to the advancements in Natural Language Processing and Machine Learning, sarcasm detection is being done these days. Primarily, sarcasm is used heavily on social media platforms and various giant companies have come up with their own ways of detecting it. Twitter checks for sentences having sarcasm-related hashtags, emoticons, and words like lol, lmao, hehe etc. Twitter also takes into consideration the polarity score of the instances to detect sarcasm. Other methods can be, to look for all caps letters, consecutive repeated characters like sooooooooo, heeee, raellllllyyyy etc. Additionally, Twitter checks the star ratings and matches them with the comments to see whether they depict the same meaning or not. The rise in expression of people's opinions and views on the social media (where people typically express negative sentiments sarcastically) and use of user feedback in market analysis has made sarcasm detection more crucial. Companies that focus on sarcasm detection can have more valuable data and insights which can be helpful to improve their products.

Sarcasm detection has been quite a popular research topic in NLP. There are existing works on sarcasm detection, however, they have some lacunae which we identified. For example, current works fail to consider enough context for the sentence to be able to predict whether it is sarcastic or not. Another observation was that most of the work was on Reddit and Twitter like datasets and product reviews/ user feedback was not focused on. Therefore, in this work, we propose a scalable and generalised model for sarcasm detection based on a textual entailment approach which we apply on a dataset extracted and built from Amazon product reviews data. We tested the model using precision, recall and F1 measure metrics. We compared the results with a baseline model consisting of a BERT (Bidirectional Encoder Representations from Transformers) [8] binary classifier by fine-tuning it on the created dataset which gave an F1 score of 0.48 whereas the proposed model gave an F1 score of 0.76. Thus, the main contributions of this paper are:

- A novel approach for sarcasm detection using a textual entailment model is proposed
- A dataset is curated and created from the Amazon reviews dataset [16] methodically and can be made available to the research community.
- Meticulous annotation was done for the created dataset to enable the evaluation of the proposed approach
- We obtain the state of the art results on the product reviews dataset which is 0.76 F1. (Note that the results on the same dataset of 0.78 F1 have been reported in the literature but on a very tiny portion of the dataset containing 87 sarcastic and 164 non sarcastic reviews.)
- As there is no work on the exact same dataset in the past, we also provide a baseline for this work which is a BERT binary classifier and demonstrate that the proposed approach beats the baseline effectively with a large margin.

The remainder of the paper is structured as follows. Section 2 presents the detailed literature review on the existing sarcasm detection techniques. Section 3 presents the available datasets and the details of the preliminary experiments performed. It also explains the creation of the dataset used in this work and details the annotation work done by the authors for the same while maintaining acceptable pairwise and average agreement between the annotators as per the Kappa statistics. Section 4 highlights the basic assumption behind the work and explains the proposed idea. This is followed by the results and experiments section namely section 5. Section 6 presents a thorough error analysis of cases where the proposed model fails to detect sarcasm. Section 7 concludes the paper, presenting some interesting insights on the task and data at hand while pointing out some directions for future work.

2 RELATED WORK

Researchers have worked on multiple approaches to identify sarcastic language in text. They have worked on multiple datasets, including Twitter and Reddit dataset and the focus is on social media [3, 4, 6, 11, 20]. Various models for sarcasm detection tasks have been used, ranging from traditional ML models like Support Vector Machines to more advanced models like RNN (Recurrent Neural Network) and BERT [13]. In the paper [18] Parde et al. use Naive Bayes classification on tweets and Amazon product reviews for sarcasm detection. They extract the features that show sarcasm in the different domains as well the general features of sarcasm irrespective of the domain. The reported F1 score is 0.69 on the Twitter dataset and 0.78 on an Amazon Reviews dataset. However, the dataset considered by the authors is very small consisting of 87 sarcastic and 164 non sarcastic sentences. They left out some important features like the exclusion of world knowledge, text normalization, and an enhanced lexicon of sentiment and situational phrases which are the limitations of this approach in addition to the drawback that they did not consider enough context.

In [22], the authors introduce SCUBA (Sarcasm Classification Using a Behavioral modeling Approach), a behavioral modeling framework for sarcasm detection. Different forms of sarcasm are discussed, and relevant features are constructed for representation on Twitter. SCUBA utilizes historical user information and psychological aspects to effectively detect sarcastic tweets, making it suitable for real-time applications with computational constraints. It can be extended to other social media sites, providing a valuable tool for consumer assistance teams to respond appropriately to sarcastic tweets and avoid potential PR issues. The highest reported accurcay is 0.94 for a data split of 90:10 while experimenting. In [5] the authors Baruah et al. use BERT and BiLSTM (Bidirectional long short-term memory)classifier for sarcasm detection on the Twitter and Reddit datasets. They fined-tune the classifiers using grid search on the datasets and report an F1 score of 0.743 on Twitter dataset and 0.658 on Reddit dataset. Sarcasm detection has been quite a popular research topic in NLP. The authors have not performed any experiments or reported any results on any of the product reviews dataset like Amazon, eBay etc.

A BERT-based method to identify and detect sarcasm in a conversion context using BERT is described in the work [2] by Kalaivani et al. The datasets used to detect sarcasm are those from Twitter

and Reddit and were provided through the Figurative Language Processing 2020 shared challenge on sarcasm identification. Using contextualized word embeddings produced by the BERT model, the model obtained an F1 score of 0.738 on the Twitter dataset and an F1 score of 0.743 on the Reddit dataset. The authors suggest leveraging context as a future direction of their work. In [14], Misra et al. present relatively large-scale and high-quality dataset for the task of sarcasm detection as well as showcase through training a Hybrid Neural Network with attention mechanism that deep learning models can reliably learn sarcastic cues from the text in an expressive manner. This paper uses News Headline Dataset and an accuracy of 89.7% is reported. However, the F1 score or precision recall is not mentioned in the paper. In [12], Joshi et al. present a sarcasm detection system using context incongruity as a basis. It incorporates lexical, pragmatic, explicit, and implicit incongruity features. The highest precision of 0.81, a recall of 0.97, and F1 Score of 0.88 is reported on a Twitter dataset. Evaluation of tweets and discussion forum posts shows a 40% improvement over a rule-based algorithm is what they conclude. The system also introduces inter-sentential incongruity, resulting in a significant improvement in precision.

To the best of our knowledge, there is no work on sarcasm detection in product reviews and therefore we focus on this particular challenge using a novel textual entailment approach.

3 AVAILABLE DATASETS AND PRELIMINARY EXPERIMENTS

In this section, we list the available standard datasets of NLP and detail out some of the preliminary experimentation done on the them. Some datasets used for Natural language inference are SNLI (Stanford Natural Language Inference) [7], ANLI (Adversarial Natural Language Inference) [17] and DocNLI (Document-level Natural Langugage Inference)[24].

3.1 The News Headlines Dataset

As per the studies in the available literature, the Twitter dataset is often used for sarcasm detection. The dataset is collected using hashtag-based supervision and is found to be noisy in terms of labels and language. In addition to this, the sarcasm detection in tweets requires contextual tweets as they might be the replies to other tweets. Due to these limitations, we decided to use the News Headline dataset,which is made up of records collected from two news websites, namely The Onion and HuffPost. The Onion has sarcastic news of the current events, and HuffPost has original, non-sarcastic news. Compared to current Twitter datasets, this dataset has a number of advantages. Because news headlines are written in a formal, professional manner, there are fewer spelling mistakes and instances of casual language, which lowers sparsity thereby enhancing the likelihood of discovering pre-trained embeddings. The Onion also produces high-quality labels with less noise than Twitter datasets because its main objective is to publish sarcastic news. The news headlines we acquired are also self-contained in the sense that they do not rely on any context, making it simpler to spot the genuine sarcasm compared to tweets, which are frequently replies to other tweets and are context dependent.

The News Headlines dataset has three properties for each record, namely, the headline of the news article, a binary label indicating whether the headline is ironic/sarcastic or not, and a link to the original news article that may be used to gather supplemental information. In conclusion, this new dataset is an invaluable resource for sarcasm detection academics and practitioners, enabling a more precise and nuanced examination of sarcasm in language as well as the advancement of models and methods for sarcasm identification.

3.2 Creation and annotation of the CustomAPR Dataset

With 82.83 million unique reviews, the original Amazon Reviews dataset contains a sizable database of product evaluations from over 20 million people. The data in this is collected from May 1996 to July 2014. It spans over 9.35 million products/items [1]. With the use of this dataset, we intend to create a more applicable and practical sarcasm detection system, which will ultimately help companies to enhance their goods and services based on consumer feedback and reviews. In this work, we identify the potential impact of detecting sarcasm in product reviews. This will help businesses better understand their customers' attitudes towards their products and leverage the feedback given by them to improvise their products in future.

Unfortunately, there is no pre-existing product reviews dataset that fits our needs where information about sarcasm is labelled for experimental study. Hence, we decided to create a new product reviews dataset by filtering out instances from Amazon reviews dataset uniformly across various products. The Amazon reviews dataset consists of a vast collection of product reviews from around 20 million users, totaling 82.83 million reviews. This dataset needs to be annotated manually by assigning a label to reviews based on whether the review is sarcastic or non-sarcastic. If the review is sarcastic, it will be labeled as 1 and if the review is non-sarcastic, it will be labeled as 0. Before labeling, probable sarcastic reviews need to be filtered from this huge dataset and then manual annotation of each review is required. To filter sarcastic reviews from the Amazon reviews dataset, a two-step filtering process is employed. In the first step, each review is passed into the sentiment classification model which predicts whether the input review is having positive or negative sentiment. If there is a disparity between the sentiment of the review and its rating, the review will be labeled as sarcastic. If there is no disparity between the sentiment of the review and its rating, then the review is labeled as non-sarcastic. The bert-base-multilingual-uncased-sentiment model [15]is used for the sentiment classification of the review. This model takes a text as input and then predicts the sentiment of the text as a number between 1 and 5 where 1 indicates poor and 5 indicates best score.

In the second step, dictionaries of positive and negative words is created [10], and each word from the filtered reviews is compared against these dictionaries. The number of positive and negative words is counted in each filtered review. If the number of positive reviews greatly dominates the number of negative words, then the review is considered a positive sentiment review. If the number of negative words greatly dominates the number of positive words, then the review is considered a negative sentiment review. After the two-step filtering process, the following class of reviews is filtered out from the dataset:

(1) Reviews that have a high positive sentiment but a poor rating of 1.
(2) Reviews that have a high negative sentiment but a very high rating of 5.

These filtered reviews need to be manually annotated to be used for the task of sarcasm detection. If the review and summary had contrasting opinions about the product, the annotator or gold labor labels it as sarcastic. If there was no disparity between the review and its summary, then the review is labeled as non-sarcastic by the human annotator. This way a dataset consisting of 3000 manually annotated reviews was created. This dataset consisted of 128 sarcastic reviews and 2872 non-sarcastic reviews. Here we would like to point out that the sarcasm is a small component of the natural language used in any form of communication and so the dataset is unbalanced as we had expected it to be. Henceforth in this work, we refer to the dataset created by us as CustomAPR dataset.

3.3 Preliminary Experiments

The major focus of the work on sarcasm detection tasks is on the Twitter and Reddit datasets, or so to say the social media. We explore a more general use case of sarcasm detection such as sarcasm detection in product reviews. Sarcasm detection in product reviews is quite important because it can help businesses to understand what customers feel about their products. Sarcasm is often used by users to show disappointment and dissatisfaction about a product and hence can help businesses to improve their products based on users reviews or feedback which needs to be automatically processed and analysed.

Before proceeding towards the experiments, we carried out some preliminary experiments while doing our study of sarcasm detection. In this section, we detail out those experiments and the results of these experiments are shown in Table 1

3.3.1 ***LSTM model on News Headline Dataset.*** Here we develop an LSTM [9] (Long Short Term Memory) model to analyze the News Headline dataset. The main objective is to predict whether a given news headline is sarcastic or not. The LSTM model is selected because it has shown promising results in multiple Natural Language Processing tasks. We only use the news headline as input to the model leaving the news article aside. The goal was to see how effective LSTM models are on the News Headline dataset with only the news headline as the input.

The architecture of the model consists of an embedding layer, a bidirectional GRU layer, and a dense output layer with a sigmoid activation function. The model is then compiled using binary cross-entropy loss and Adam optimizer. We use the model having the best validation accuracy on an 80-20 split. The model is trained on the training data and validation is done on the testing data. The final loss and accuracy are reported along with other metrics such as F1 score, precision, and recall.

3.3.2 ***BERT model on News Headline Dataset.*** After working on the LSTM model, we explore more advanced models such as the BERT. Here also, we use the news headline as input to the BERT model. The goal is to see how well this model can classify the news headline as sarcastic or non-sarcastic with only the headline as input to the model. The bert-base-uncased model is used here which is fine-tuned and configured for the binary classification task of detecting sarcastic headlines. The model is compiled with the Adam optimizer and Sparse Categorical Cross-Entropy loss as the loss function. The model is trained for 3 epochs, and tested on the testing data. The performance metrics of F1 score, precision, recall, and accuracy are reported. In Table 2, we see that the scores on BERT model are very high. However, it is done using cross validation and the results are likely to lower if used on an unseen dataset.

Model	F1	Precision	Recall	Accuracy
LSTM on News headlines	0.72	0.67	0.79	0.65
BERT on News Headlines	0.98	0.96	0.98	0.97

Table 1: Preliminary Experiments Results

4 PROPOSED APPROACH

The LSTM and BERT model show impressive results on the News Headline dataset is what we observe from the preliminary experiments. However, this model incorporates only the news headline as input leaving the news article aside which could form an important context. These models misses out on useful information contained in the news article. Also, this model is trained and tested on the same dataset, which could not fit well in real-world scenarios where we can encounter unseen and unexpected data. To address these issues, a new approach to detect sarcasm is proposed that uses the textual entailment method.

4.1 Textual entailment

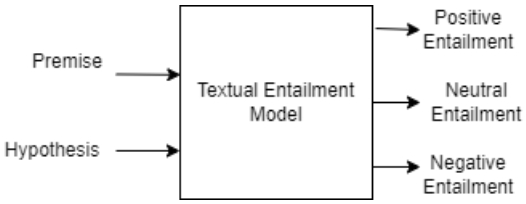

Figure 1: General Textual Entailment Model

Textual entailment is a natural language inference task that determines whether one text (called a hypothesis) can be inferred from another text (called a premise). [19] In general the premise is a paragraph or a longer text or context and hypothesis is what is to be classified as being entailed/inferred or not entailed / not -inferred. There are mainly 3 types of textual entailment. These are positive entailment, negative entailment, and neutral entailment. If the hypothesis can be inferred from the premise, then the entailment is positive. For example, if the premise is 'John owns a car' and the hypothesis is 'John owns a vehicle', then here we have positive entailment as the hypothesis can be inferred from the premise.

If the hypothesis cannot be inferred from the premise, then the entailment is negative. For example, if the premise is 'John owns a car' and the hypothesis is 'John does not own a vehicle', then here

we have negative entailment because the hypothesis contradicts the premise. If there is no clear inference relationship between the hypothesis and the premise, then the entailment is neutral. For example, if the premise is 'John went to the store' and the hypothesis is 'John bought some stationary', then here we have neutral entailment as the hypothesis cannot be inferred from the premise directly and we need additional information for this. The basic model of textual entailment is shown in figure 1. In the case of sarcasm detection in the News Headlines dataset, the news headline is the hypothesis and the news article is the premise. If the news headline cannot be inferred from the news article, we can say that the news headline is sarcastic with respect to the article. If the news headline can be inferred from the news article or there is no clear relationship between the news headline and the news article, we can say that the news headline is non-sarcastic.

This textual entailment approach has several advantages over traditional methods. Traditional methods depend on pre-defined syntactic and lexical features which are unable to capture the complex and nuanced language used in most sarcastic sentences. The textual entailment approach leverages the full article content to predict sarcasm in the News Headline dataset. It can also handle new and unseen data because in this approach the model is trained on one dataset (natural language inference datasets such as SNLI, ANLI, DocNLI etc.), and tested on another dataset (such as News Headline dataset or Amazon reviews dataset). Overall, this approach is more flexible and general and can improve the performance and the robustness of the NLP models in detecting sarcasm. To determine the entailment relationship between the premise and the hypothesis, we feed them into the textual entailment model and it outputs either of the 3 possible types of entailment i.e. positive entailment, negative entailment, or neutral entailment.

4.2 Comparison of results on News Headlines dataset by training on individual available datasets and their combination

To use the textual entailment approach, we first need to train the model on various datasets such as SNLI, ANLI, DocNLI, etc. These datasets have been developed for the natural language inference task. These datasets contain large amounts of annotated records of textual entailment relationships. This allows us to train and test our model on a wide range of text and language. The training was done on different available datasets and testing was done on News Headlines dataset which was an unseen one. No cross validation was used. Looking at the results in Table 2, it is evident that the LSTM model trained on multiple datasets yields better results than training on a single dataset alone. The LSTM model trained on SNLI and ANLI datasets outperforms other models. Hence, this model is more suitable for sarcasm detection as compared to other models. Thus, the conclusion was that for further experimentation, training could be done using SNLI and ANLI both datsets and testing on the relevant test dataset namely News Headlines/ CustomAPR whichever applicable.

Model	F1	Precision	Recall	Accuracy
SNLI	0.43	0.45	0.41	0.60
ANLI	0.41	0.43	0.39	0.60
SNLI + ANLI	0.47	0.48	0.46	0.65
DocNLI	0.21	0.25	0.18	0.30

Table 2: Train using LSTM Model on various datasets and test on News Headlines dataset

4.3 Textual Entialment on News Headlines Datatset

The modeling of the sarcasm detection problem as a textual entailment problem on the News Headlines dataset is as shown in the figure 2

4.3.1 ***Textual Entailment on News Headlines Dataset using LSTM model.*** To use textual entailment for sarcasm detection in the News Headline dataset, we need to train this model. Once the model is fully trained, we can pass the news headline and news article as input to the model and the model would predict the 3 possible types of entailments - positive, negative, or neutral. If the model predicted negative entailment, we say that the news headline is sarcastic. If the model predicted positive or neutral entailment, we conclude the news headline to be non-sarcastic. By using the textual entailment in this way, we can take advantage of the full content to better predict the sarcastic sentences and understand the user's intention and motivations. We use a bi-directional recurrent neural network (RNN) with two different LSTM units to build the textual entailment model. The bi-directional RNN reviews the premise and the hypothesis both independently as well as in relation to each other. This helps in a better understanding of the relationships between the two texts. To stop the network from assigning undue importance to inconsequential words like 'a', 'an', and 'the', a dropout layer is used. This layer is used on everything except the internal gates of the LSTM layers. This is done because the loss of certain pieces of crucial memory could negatively affect the complicated relationships required for forming first-order logic. The output from the LSTM layers is then passed through the fully connected layers. It provides a single-valued score which indicates the strength of each type of entailment. This score is used to calculate the final result and confidence level. SNLI dataset consists of annotator labels which are used to calculate the scores of each of the entailments. For example, for the annotator labels with 3 positive entailments, 1 neutral entailment, and 1 negative entailment, the score will be [3/5, 1/5, 1/5] corresponding to positive, negative, and neutral entailment respectively. These scores are used to train the model.

Upon training the model on various datasets and analyzing the results, it is evident the model performs better when trained on multiple datasets as compared to training on a single dataset. The LSTM model trained on both SNLI and ANLI datasets performs better than other models. Therefore, this model is used for sarcasm detection in the News Headline dataset. To use the News Headline dataset for testing the LSTM textual entailment model, pre-processing needs to be done. The News Headline dataset consists of the field 'article

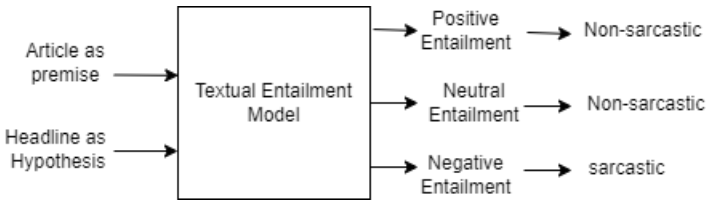

Figure 2: Textual Entailment Model on News Headlines

link' that contains links to the corresponding news article. The article content needs to be fetched using the article link to use this as a premise for our model. BeautifulSoup which is a Python library is used to extract the article content from the link. The problem with the fetched articles is that they are often lengthy and may contain unwanted texts such as image captions, ad texts, irrelevant information, etc that need to be removed before being used by the model. The lengthy articles can be summarized to 2-3 lines long using various summarization techniques to capture the key points of the article.

Summarization is a natural language processing task that involves condensing long texts into shorter, more digestible texts. Summarization of the fetched articles helps to focus on the relevant information and avoid getting bogged down by irrelevant details. It also helps in reducing bias in the testing process by presenting all the articles in a similar format. Google T5 summarizer [21] is used to summarize the data into around 512 words per record. This was done because the textual entailment requires a maximum of 512 tokens as input. Gensim summarizer [1] was also used for experimentation, but it has a limited number of input parameters resulting in less control over the output. Several flaws in the data preprocessing were discovered. Some of the article links had only one or two lines of text. 12.5k records out of 26k records had an unwanted text. After further analysis, we found that around 18.5k records had less than 512 words, and 8k records had more than 512 words. 2.5k records were dropped out of this 18.5 records due to insufficient text. After the News Headline dataset was preprocessed, we used the LSTM model trained on the SNLI and ANLI datasets to detect sarcasm. The news headline was used as the hypothesis and the news article was used as the premise and the model predicted the 3 possible types of entailments- positive, negative, or neutral entailment. The negative entailment meant that the news headline was sarcastic while other kinds of entailments meant the headlines were non-sarcastic. The predicted results were compared with the ground truth provided in the dataset to report performance metrics such as F1 score, precision, recall, and accuracy. The LSTM model on the DocNLI dataset is sub-optimal because of the lengthiness of the premise present in the dataset, which hindered the model's ability to understand and comprehend the meaning, relationships, and dependencies between the premise and hypothesis. This issue can be resolved by summarizing the premise into smaller, more manageable sentences and then using these summarized sentences to train and test the LSTM model.

4.3.2 ***Textual Entailment on News Headlines dataset using BERT.*** After working on the LSTM model, we decided to work on state-of-the-art models for the textual entailment task. We selected the NLI-DeBERTa-Base (Natural Language Inference - Decoding-enhanced BERT with disentangled attention - Base) [2] model which is an extension of the BERT model and is based on the transformer architecture. This model has been extensively pre-trained on multiple datasets like SNLI and MNLI [23] and hence it is very effective for the textual entailment task. The DeBERTa architecture includes various modifications and improvements to the BERT model and therefore performs better on a wide range of NLP tasks. This model has also been used for other tasks such as question answering, text classification, and sentiment analysis. Overall, the NLI-DeBERTa-Base model is very powerful and versatile with the ability to extract robust and meaningful representations of texts using its advanced architecture and training approach.

The NLI-DeBERTa-Base model was used for sarcasm detection in the News Headline dataset. The news headline was passed as the hypothesis and the article content was passed as the premise to the model and the model predicted 3 possible types of entailments - positive, negative, and neutral. The negative entailment was mapped to sarcastic headlines, while other entailments were mapped to non-sarcastic headlines. The predicted values were compared with the ground label present in the dataset and various performance metrics such as F1 score, precision, recall, and accuracy are reported as shown in Table 3.

Model	F1	Precision	Recall	Accuracy
LSTM	0.25	0.45	0.18	0.58
NLI-DeBERTa-Base	0.98	0,.96	0.98	0.97

Table 3: Textual Entailment Approach using LSTM and NLI-DeBERTa-Base Models on the News Headlines dataset

5 EXPERIMENTS AND RESULTS FOR EVALUATION OF THE PROPOSED APPROACH

In this section, we present the experiments performed and the results obtained on two datasets namely News Headlines dataset and the CustomAPR datset using the proposed textual entailment approach.

[1] https://tedboy.github.io/nlps/generated/gensim.summarization.html

[2] https://huggingface.co/cross-encoder/nli-deberta-base

5.1 Results of proposed approach on News Headlines Dataset and CustomAPR dataset

The results in Table 3 show that the NLI-DeBERTa-Base model is much more effective in detecting sarcasm in the News Headline dataset as compared to the LSTM model. The NLI-DeBERTa-Base model achieved an F1 score of 0.50 which is far better than the F1 score of 0.25 achieved by the LSTM model. The NLI-DeBERTa-Base model also performs better in other performance metrics such as precision, recall, and accuracy. With this observation, we move ahead with the experiments on the customAPR dataset.

After we worked on the News Headline dataset, we started working on the annotated CustomAPR dataset. This dataset contains 4 fields - review, summary, rating, and sarcasm. The sarcasm column contains a binary label indicating whether a particular review is sarcastic or not. The dataset consists of 3000 manually annotated product reviews, out of which 128 are sarcastic reviews while 2872 are non-sarcastic reviews. From this dataset, 3 sets of the dataset are created by mixing different proportions of sarcastic and non-sarcastic reviews.

- Set 1: The first set consists of 128 sarcastic reviews and 1272 non-sarcastic reviews.
- Set 2: The second set consists of 128 sarcastic reviews and 2000 non-sarcastic reviews.
- Set 3: The third set consists of 128 sarcastic reviews and 2872 non-sarcastic reviews.

The NLI-DeBERTA-Base model is used for sarcasm detection in each of these sets because this model has outperformed the LSTM model in sarcasm detection in the News Headline dataset. The review was used as the premise and the summary was used as the hypothesis to pass input to the model and the model predicted the three possible types of entailments - positive, negative, or neutral entailment. If the model predicted a negative entailment, we considered the review to be sarcastic. If the model predicted positive or neutral entailment, we considered the review to be non-sarcastic. The modeling of the sarcasm detection problem as a textual entailment problem on the CustomAPR dataset is as shown in the figure 3. Since the datasets are highly imbalanced, macro-average F1 score, macro-average precision, and macro-average recall are used in place of F1 score, precision, and recall. This is done to give equal weightage to each class, regardless of its size. These performance metrics provide a more accurate measure of model performance. These metrics are commonly used in multi-class classification problems where class distribution is imbalanced.

The textual entailment approach was used in the CustomAPR dataset using the NLI-DeBERTa-Base model and the macro-average F1 score, precision, recall are reported. These evaluation metrics ensure that the performance of the model is evaluated fairly across all classes, regardless of their frequency or size in the dataset.

5.2 Baseline Model for comparison with the proposed approach results on CustomAPR dataset

To check how effective the textual entailment approach is as compared to other methods and models, a comparative analysis is done by testing the CustomAPR using a binary classifier with BERT as the baseline as explained below. This enables us to compare the performance of the textual entailment model with other widely used models and methods that exist already. By comparing the results obtained from both models, we can get a lot of valuable insights about the strengths and the weaknesses of each method and could identify areas where there is scope for improvement. This comparative analysis ensures that the textual entailment method is robust and effective and can be confidently used in real-world applications. BERT model for binary classification on CustomAPR dataset is created. The 'bert-base-uncased' model is used as the baseline model which is then fine-tuned for the sarcasm detection task in the CustomAPR dataset. The input data which includes reviews, summaries, and sarcasm labels, is loaded and then encoded into input IDs using the tokenizer. 10-fold cross-validation which involves splitting the data into ten subsets is used to evaluate the performance of the model. The cross-validation ensures that the model is robust. During the training phase, the BERT model is fine-tuned on the training set using AdamW optimizer with 2e-5 as the learning rate. The model is trained with three epochs.

5.3 Final Results on CustomAPR Dataset

Though we performed thorough experimentation on the News Headlines dataset, its applicability seems to be limited which is why we experiment on a dataset of product reviews where identifying sarcasm can prove to be beneficial. In the final experiments on CustomAPR dataset, we decide to use the NLI-DeBERTa-Base model which is a textual entailment model. We use the binary BERT classifier as the baseline. The product review is passed as the premise whereas the summary is passed as the hypothesis to the textual entailment model. The model predicts the three possible types of entailments. If the model predicts negative entailment, we conclude the review is sarcastic. If the model predicts other types of entailments, we conclude that the review is non-sarcastic. The predicted results are compared with the manually annotated ground truth to calculate various performance metrics. The final results are shown in the Table 4. Here the metrics Precision, Recall and F1 are all macro-averaged as the dataset is unbalanced.

The results obtained from utilizing both the models on different sets of CustomAPR dataset shows that the NLI-DeBERTa-Base model far outperforms the binary BERT classifier model in every performance metric. The F1 score achieved by the NLI-DeBERTa-Base model is significantly higher than the F1 score achieved by the baseline model. This suggests that the textual entailment approach is a highly effective method for sarcasm detection as it considers both the summary and review for prediction.

6 DISCUSSION AND ERROR ANALYSIS

Textual entailment models are better at capturing the context and the intricate semantic relationships, which are crucial for detecting sarcasm. These models are specifically designed to capture the relationships between different pieces of text, i.e., whether one text entails, contradicts, or is neutral with respect to another text. In this case, using NLI-DeBERTa for sarcasm detection leverages its ability to understand the nuanced relationships between the review and summary text, which helps in identifying sarcastic instances where the summary contradicts the review's sentiment. NLI-DeBERTa,

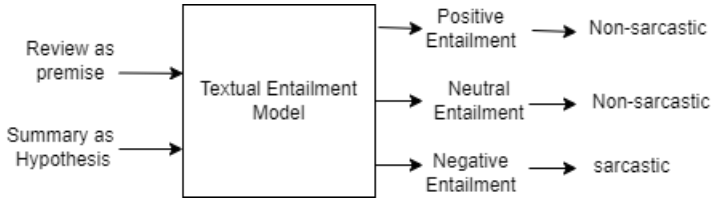

Figure 3: Textual Entailment Model on CustomAPR dataset

Dataset	Methods	Macro-average F1	Macro-average - Precision	Macro-average Recall
Set 1	Proposed Approach	0.76	0.71	0.87
	Baseline Model	0.48	0.47	0.50
Set 2	Proposed Approach	0.72	0.66	0.89
	Baseline Model	0.60	0.59	0.62
Set 3	Proposed Approach	0.66	0.62	0.88
	Baseline Model	0.62	0.73	0.58

Table 4: Proposed Approach Versus Baseline Model On CustomAPR dataset

being a more advanced model, has a better grasp of semantic information, such as implied meanings, subtle contradictions, and contextual cues. The reviews contain mostly short texts and the textual entailment models are more sensitive to contradictions within a short piece of text, making it effective at spotting sarcasm. Consider the review: " *After having a freezer meltdown in my absence, I was never able to rid the ice of odor that must have been trapped inside some inner piece of plastic. About one month after using this, all the smell is completely gone!*" and the summary: "*Great*". The gold label for the above <Review, Summary> tuple is 'sarcastic'. For the given tuple, the textual entailment model correctly predicts it as a sarcastic review whereas the bert-base-uncased model predicts it as non-sarcastic. The textual entailment model tends to pick up the negative entailment between the initial positive statements and the final negative sentiment. The key difficulty here is understanding the overall sentiment. While the summary contains a positive word ("Great"), the review provides a negative sentiment, criticising the fact that it took almost a month for the odour to go away.

There are situations where the textual entailment model too fails. Consider the examples in Table 5. The gold label for the instances in Table 5 is sarcastic, but both the textual entailment model as well as the bert-base-uncased model predict the given review as non-sarcastic. The review and summary do not contain any explicitly exaggerated words, tonal cues, or traditional markers of sarcasm. These models struggle with this example due to the lack of overt contradictory language. The limited length of the text makes it more challenging for models to capture the implied sarcasm. Short texts can lack the context needed for proper understanding. The sarcasm in some cases is implied through the contrast between an initial positive statement and the subsequent negative statement. This subtlety might be challenging for both models to detect. For example, consider the review: "*started out great... worked for a day*". and the summary: "*Worked for me*". Here, initially, the sentence begins on a positive note in the review, however, it is followed by a negative sentiment expressed in the next part of the sentence. This makes it difficult for the proposed model to pick the right sentiment and gets confused leading to a false negative.

7 CONCLUSION AND FUTURE WORK

In this work, we propose a model for sarcasm detection which relies on a Textual Entailment approach and tested it on our CustomAPR datset to obtain promising results of 76% macro-averaged F1. Though sarcasm is a small part of communicated natural language, it has its own importance. The results obtained from applying the textual entailment approach on the News Headline and the Custom APR datasets suggest the potential value of advanced NLP techniques for improving the accuracy of sentiment analysis tasks such as sarcasm detection. Such approaches could prove to be invaluable tools in a wide range of applications with further research and development in these areas. This approach could be impactful in a wide range of applications ranging from social media monitoring to feedback analysis and beyond. This research work focused on the classification of reviews into two classes - sarcastic and non-sarcastic reviews. In future, this work can be extended to detect sarcasm in memes. Memes are a distinct form of communication that uses sarcasm and irony to convey meaning. They are becoming more popular these days because of the increasing use of social media sites. Sarcasm detection in memes will require developing new approaches to understanding the visual and linguistic elements of memes and coming up with models that can effectively identify sarcasm in this context.

REFERENCES

[1] [n. d.]. The Amazon Product Reviews Dataset. https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews. [Online; accessed 01-Feb-2023].

[2] Kalaivani A. and Thenmozhi D. 2020. Sarcasm Identification and Detection in Conversion Context using BERT. In *Proceedings of the Second Workshop on Figurative Language Processing*. Association for Computational Linguistics, Online, 72–76. https://doi.org/10.18653/v1/2020.figlang-1.10

[3] Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language*

Sr. No.	Review	Summary	Gold label	Predicted label
1.	Unless you want it to actually do what it's supposed to do, the app works amazing	Breeze to use	sarcastic	non-sarcastic
2.	The book is a masterpiece, if your idea of a masterpiece is pages and pages of clichés	Five stars	sarcastic	non-sarcastic
3.	A real gem, when it's not lost in the sea of better options	Gem	sarcastic	non-sarcastic

Table 5: Examples where the proposed approach fails

Learning. Association for Computational Linguistics, Berlin, Germany, 167–177. https://doi.org/10.18653/v1/K16-1017

[4] David Bamman and Noah Smith. 2021. Contextualized Sarcasm Detection on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media* 9, 1 (Aug. 2021), 574–577. https://doi.org/10.1609/icwsm.v9i1.14655

[5] Arup Baruah, Kaushik Das, Ferdous Barbhuiya, and Kuntal Dey. 2020. Context-Aware Sarcasm Detection Using BERT. In *Proceedings of the Second Workshop on Figurative Language Processing*. Association for Computational Linguistics, Online, 83–87. https://doi.org/10.18653/v1/2020.figlang-1.12

[6] Mondher Bouazizi and Tomoaki Otsuki. 2016. A Pattern-Based Approach for Sarcasm Detection on Twitter. *IEEE Access* 4 (2016), 5477–5488. https://doi.org/10.1109/ACCESS.2016.2594194 Publisher Copyright: © 2016 IEEE..

[7] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 632–642. https://doi.org/10.18653/v1/D15-1075

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[10] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Seattle, WA, USA) *(KDD '04)*. Association for Computing Machinery, New York, NY, USA, 168–177. https://doi.org/10.1145/1014052.1014073

[11] Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic Sarcasm Detection: A Survey. *ACM Comput. Surv.* 50, 5, Article 73 (sep 2017), 22 pages. https://doi.org/10.1145/3124420

[12] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. 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)*. Association for Computational Linguistics, Beijing, China, 757–762. https://doi.org/10.3115/v1/P15-2124

[13] Amit Kumar Jena, Aman Sinha, and Rohit Agarwal. 2020. C-Net: Contextual Network for Sarcasm Detection. In *Proceedings of the Second Workshop on Figurative Language Processing*. Association for Computational Linguistics, Online, 61–66. https://doi.org/10.18653/v1/2020.figlang-1.8

[14] Rishabh Misra and Prahal Arora. 2023. Sarcasm detection using news headlines dataset. *AI Open* 4 (2023), 13–18. https://doi.org/10.1016/j.aiopen.2023.01.001

[15] Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained Sentiment Classification using BERT. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, Vol. 1. 1–5. https://doi.org/10.1109/AITB48515.2019.8947435

[16] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 188–197. https://doi.org/10.18653/v1/D19-1018

[17] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4885–4901. https://doi.org/10.18653/v1/2020.acl-main.441

[18] Natalie Parde and Rodney Nielsen. 2018. Detecting Sarcasm is Extremely Easy ;-). In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles*. Association for Computational Linguistics, New Orleans, Louisiana, 21–26. https://doi.org/10.18653/v1/W18-1303

[19] Adam Poliak. 2020. A survey on Recognizing Textual Entailment as an NLP Evaluation. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Association for Computational Linguistics, Online, 92–109. https://doi.org/10.18653/v1/2020.eval4nlp-1.10

[20] Rolandos Alexandros Potamias, Georgios Siolas, and Andreas Georgios Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications* 32, 23 (jun 2020), 17309–17320. https://doi.org/10.1007/s00521-020-05102-3

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21, 1, Article 140 (jan 2020), 67 pages.

[22] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*. 97–106. https://doi.org/10.1145/2684822.2685316

[23] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. https://doi.org/10.18653/v1/N18-1101

[24] Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A Large-scale Dataset for Document-level Natural Language Inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 4913–4922. https://doi.org/10.18653/v1/2021.findings-acl.435

Received 2023; revised 2023; accepted 2023