National College of Ireland

Transformer based Detection of Sarcasm and it's Sentiment in Textual Data

MSc Research Project
Data Analytics

Shubham Ram Gosavi
Student ID: x20190824

School of Computing
National College of Ireland

Supervisor: Amandeep Singh

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shubham Ram Gosavi
Student ID:	x20190824
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Amandeep Singh
Submission Due Date:	19/09/2022
Project Title:	Transformer based Detection of Sarcasm and it's Sentiment in Textual Data
Word Count:	4942
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Transformer based Detection of Sarcasm and it's Sentiment in Textual Data

Shubham Ram Gosavi
x20190824

Abstract

Today data has become the most valuable aspect as it is driving all the major decision making process all around the globe. The biggest platform of data generation is social media where people openly convey their likes and dislikes, and using this data many organizations make business driving decisions. The field of *Sentiment Analysis* is one such application which helps classify a review of any service or product and help companies to improve their line of services. But automatically classifying sarcasm, which is widely used on the internet, is a challenge. This research proposes a novel approach, *Bidirectional Encoder Representations from Transformers* (*BERT*) which is a transformer-based neural network approach to detect sarcasm in the input text and sentiment behind the text with the help of *Valence Aware Dictionary for Sentiment Reasoning* (VADER). To achieve this, *News Headline* data originally sourced from website *TheOnion* will be incorporated with the transformer architecture which predicts if a particular text is sarcastic or not and also the sentiment behind the text using *VADER*. The analysis of this model shows that *BERT* comes out to be the best performing model with 77% accuracy in detecting sarcasm and *VADER* is able to classify each textual sentiment into positive and negative while evaluating that being sarcastic the sentiment seen is mostly negative. The proposed research can be incorporated to build classification systems by organizations who mainly deal in e-commerce to classify the reviews they get on the products and also understand the sentiment behind every review of their customer to improve product or business quality.

1 Introduction

With the evolution of language, sarcasm has become a very casual form of communication. To increase bitterness without really doing so, a combination of mocking and false politeness is used. In a face-to-face conversation, sarcasm may be detected by observing and analyzing the speaker's way of speaking, body language, and the scenario in which they are speaking. Additionally, since one cannot see or hear the speaker's emotion or tone, a context-aware technique is necessary to overcome these difficulties. However, social media is a single platform used by 58.4% Chaffey (2022) of the entire global population, and the data it generates is sizable enough to be employed in study to improve human lives. Research by Akula and Garibay (2021) informs that since the growth of the internet, such data is now available on social media and e-commerce sites, where it is used for sentiment analysis, opinion mining, and online trolling. Building computational models is a whole new job, despite the fact that fields from neuropsychology to language studies have demonstrated a lot of interest in the subject of sarcasm. Its contradicting nature—one doesn't mean what they say while they're being sarcastic—is the reason this is still in its infancy. For instance, the phrase "it's a wonderful feeling to carry an expensive mobile with short battery life" is meant to be ironic. The word "wonderful" makes the sentence sound incredibly positive, but it also conveys a negative feeling. Since the sentence is only a text, we are unable to determine how much negativity or positivity it expresses.

In earlier study Versaci (2020), the researcher created a number of rule-based humorous detection models with the inclusion of statistical elements such as punctuation, interjections, pragmatics, lexicon, etc. For the goal of sarcasm detection, deep learning-based models are the ones that have been investigated the most and have produced results that are close to state-of-the-art. The performance is improved when attention models are incorporated into deep learning. Customer service is the most important aspect of any business, and to provide it, according to Kreuz and Glucksberg (1989) They employ platforms like *SocialPilot*, *Buffer*, and *HootSuite*, which provide services like user sentiment research, content management, and aid companies in developing meaningful connections with their clients. The customer support team will need to put in extra work to fix the issue because these tools don't offer sarcasm detection in that corpus. The goal of this study is to create a dynamic model for textual context-based sarcasm detection and sentiment classification. The research question this study focus to answer is *How can one detect sarcasm in a textual data and understand the real sentiment behind the text? TheOnion*, which publishes news in a humorous tone, and the other traditional news platform, in charge of distributing news in a very sophisticated manner, are the two sources of the data that will be largely focused on.

A similar dataset was previously utilized by Misra and Arora (2019a), who developed an attention-based LSTM model. This study tries to develop a model of transformer architecture based on deep neural networks followed by a *NLTK VADER* model which tells the sentiment of the input sentences. *VADER* has a polarity range between (-1) to (1), with (-1) denoting that the text is highly negative and (1) denoting that it is highly positive results in the measurement and detection of sarcasm. These models will be used in the CRISP-DM technique, which offers an appropriate pathway for putting the study into practice.

1.1 Motivation

The detection of sarcasm is a particularly unusual problem since it can only be assessed or comprehended when it occurs in intimate conversation, i.e., when one can see the speaker's facial expression or hear their tone of voice. It is impossible to detect sarcasm when it is carried out on text data due to the lack of context in the data and it is impossible to read the writers' facial expressions or body language. Sarcasm in talks have become very common now a days. It is specifically employed when someone rejects or dislikes something, in this situation, sarcasm is used to represent the notion by combining good viewpoint with bad feeling. When making any kind of comment or providing a review, sarcasm has become particularly widespread. Although something is expressed in a highly favorable manner, its true meaning and feeling are quite the opposite and unpleasant. Politicians can utilize it to say or write anything that is politically correct but actually means the contrary. As mentioned in the research by Mohammad et al. (2016) British people are known for their dry, stinging sarcasm. It may be quite challenging for someone to form an opinion if they don't grasp the speaker's meaning. Since these jobs rely on social media and e-commerce data, sarcasm detection is crucial for tasks like sentiment analysis, opinion mining, improving customer service, improving a product or brand, and more. Therefore, to complete this assignment of sarcasm detection, data extraction and algorithmic modeling will be used.

2 Related Work

A summary report of related work can be found below in a tabular format 1.

2.1 Attentive Neural Network

(Potamias et al.; 2019) in the year 2019 proved attentive recurrent neural network to be a better option than the usual algorithms like *Convolutional Neural Network (CNN)*, feature engineering for the purpose of sarcasm or irony detection. The research that they proposed used a large dataset which had 198041 rows of text data which are sarcastic and 197917 tweets which are not sarcastic. Three models were incorporated in the research which are *RNN*, attentive *Recurrent Neural Network (RNN)* and *CNN* in which LSTM units was passed as one of the important parameter to the *RNN* and attentive *RNN* models and *CNN* was passed with the 2,4 and 6 filter size. When evaluated the final results of all the 3 models *RNN* came out to be the worst model among all three. A higher precision value of 91.5% was shown by the CNN model but, attentive *RNN* came out to be the best model out of all and was also able to capture negative sentiment words like '*shit*' and '*suck*' in a tweet with positive sentiment.

2.2 Ensemble Machine learning and deep learning technique comparison for sarcasm detection

Deep Learning is the most tried and tested algorithm or model used for the purpose of sarcasm detection because most of the research paper are seen to be using deep learning techniques to detect sarcasm and also give good results. But Ghanem et al. (2020) can be seen to oppose to this thinking and proves in his research that machine learning gives better output. The research was conducted on a pre-labelled twitter dataset with

10000 data points. The modelling was carried out after performing all the required pre- processing of the text data and models like *neural network, naive bayes, k-nearest neighbour and support vector machine (SVM)* were implemented. As a hyper parameter ensemble learning algorithm was incorporated which helped in improved performance and depending on the model with best accuracy and precision was selected. The data contained emoticons so they were replaced with their descriptive meaning using a library in python named emoji. All the 4 ensemble learning models were integrated using an using a stacking method, where 1 model served as a meta classifier and 3 as base classifiers. Finally the evaluation of the entire research came out as the stacking classifier improved the performance of detection and machine learning model gave equivalent accuracy and precision as deep learning model.

2.3 Multilingual Sarcasm Detection

Huang et al. (2017) were the first researchers who carried out sarcasm detection on multiple language based on multiple cultures which was on French, Arabic and English language. The end objective of the research was to evaluate either monolingual or feature based architecture performed better. Twitter dataset from each language was used for the implementation. The corpus created was a political subject and tweets around it were used. After carrying out the complete modelling and implementation they concluded that multilingual sarcasm detection is possible with high model performance. They suggested to carry out sarcasm detection on Hindi language as the future work.

2.4 Modelling Multi-head model on a large imbalanced data

Kumar et al. (2020) has suggested conducting research on the use of vocabulary and pragmatic parameters that report on empirical inquiry to separate sarcasm from positive and negative feelings in posts on Twitter. The experiment was conducted using two unique algorithms, *Multi-Head Self Attention based Bidirectional Long Short-Term Memory* (*MHA-BiLSTM*) and statistical machine learning approach by SVM. A self annotated large dataset named *SARC* from Reddit was extracted and was kept imbalanced to represent real world scenario and used for this research. The results and evaluation came out as *MHA-BiLSTM* out-shined SVM and *BiLSTM* model with high margins. Also this research outperformed a similar research done by Joshi et al. (2015) with difference of 7.88% in accuracy. The final statement by the researchers were that including auxiliary feature increases model performance and results in better prediction.

2.5 Sarcasm detection on human interpreted data under BLEU metrics

The research proposed by Peled and Reichart (2017) starts by explaining the meaning of sarcasm and also state it to be ambiguous because it is even challenging for human beings to detect it when tried doing in textual form. In this research the sarcasm detection is evaluated on 3 main metrics like *NIST*, *METEOR* and most commonly used *BLEU* Papineni et al. (2002) on RNN, *SIGN* and *Moses* model based on the n-gram co-occurrence based score. *PINC* metric is also introduced in this research which is a reward based metrics. After the implementation it is evaluated that universal knowledge is very important to understand and detect sarcasm. The results state that 67.5% sentiment is

interpreted by *SIGN-context* which is higher than *Moses* and hence *SIGN* outperforms all other methods of sarcasm detection.

2.6 Importance of emoticons in sarcasm detection using pytorch

A state of art performance was achieved from the research proposed by González-Ibáñez et al. (2011) where they implemented *MHA* neural network for sarcasm detection. The important component of this model was Facebook library *PyTorch* which helped in taking into consideration the emoticons which were present in the sentences of the data used. The model implemented highlights the words which are responsible for the sentence to be sarcastic *Gated recurrent unit (GRU)*, was also incorporated to understand the relation between the highlighted words. The evaluation and results of the experiment is that emoticons plays an important role for interpreting sarcasm.

2.7 Sarcasm detection model based on SASI

Punctuation marks came out to be the strongest predictors in the research done by Tepperman et al. (2006) on sarcasm detection.Tsur (2010) rejects stating that punctuation marks are the weakest in terms of predictions because detection of sarcasm takes place differently between textual data and when spoken. Tsur (2010) argues that product review sarcasm is different from the one in private conversation. Therefore, to prove the hypothesis a research based on *Semi Supervised Algorithm for Sarcasm Detection (SASI)* on the product review data. The results evaluated state that *SASI* detected 81% of pattern of evaluation under 5-fold cross validation with 77% precision and 83.1% recall.

2.8 Hugging Face for sarcasm detection

Wolf et al. (2019) researched how the pre-trained models have made it easy for the future researches and that the transformer library made it possible to collect all this huge pre-trained models at one place and accessible for the machine learning community. Transformers is indeed an ongoing project that Hugging Face's team of technologists and researchers maintains with assistance from a thriving community of more than 400 outside contributors. At the end the researchers conclude saying that, since its debut, Transformers has had substantial organic growth and is prepared to continue offering the essential infrastructure while easing the availability of new models.

2.9 BERT Vs Machine Learning

González-Carvajal and Garrido-Merchán (2020) have put forward a research where comparison of BERT with traditional *Natural Language Processing (NLP)* machine learning model like TF-IDF has been carried out. The motive of the research is mentioned to give emperical support of evidence to make BERT as default for NLP task. The experiments are carried out on IMDB data, 'Real' or 'Not' tweet data, Portuguese news data and Chinese hotel review data alongside *BERT* traditional models like Logistic Regression, Linear SVC, Multinomial naive Bayes, auto ML, etc. are implemented. In each experiment BERT is seen to outperform all the other models over the accuracy metrics and hence proving that BERT can be used as a default model.

2.10 Detection of Sarcasm with context.

Different techniques and methods are proposed by Kumar and Anand (2020) for sarcasm detection including content like BERT, RoBERTa, spanBERT. Two corpus, twitter corpus and Reddit Corpus by Khodak et al. (2017) are incorporated for the experiment. Two, sentence based data were created and used like Single sentence classification task and sentence pair classification task where BERT, SpanBERT and RoBERTa models were implimented on single sentence corpus and Siamese transformer and Dual Transformer on sentence pair classification corpus. The unique part about this research was that the pre-processing or hyper parameter tunning carried out was not dataset specific. The results this research gave is that the best F1 score was scored by the response_string i.e. the string without context and the best overall performance was given by LSTM over transformer model having layer *robert_large*.

2.11 Improvisation to the previous model on a self-built dataset to detect sarcasm

Twitter labelled dataset which consists of sarcastic tweets labelled as #sarcasm is the most widely used dataset for the purpose of sarcasm detection but according to Misra and Arora (2019b) research claim these datasets to be noisy when it comes to the languages used in these dataset and also the labels done over each tweet. To overcome this noise Misra and Arora (2019b) came up with new dataset which is created taken from two data source, one from the sarcastic new website and other from traditional news website. This created dataset was modelled on a hybrid neural network with inclusion of attention mechanism with LSTM as it output's the words which is responsible for sarcasm in the sentence. The produced results from this research was compared with the research proposed by Amir et al. (2016) which is based on n-gram word patter and under hypothesis that both the models complement each other. The comparison showed 5% increase in accuracy of the Misra and Arora (2019b) research over other.

Table 1: Related work Summary

RESEARCHER	PUBLICATION	METHOD	DATASET	WORK	ADVANTAGES
Lotem Peled et. al.	ICWSM	semi supervised algorithm for sarcasm detection, SASI	66000 in dimension it is a book and online product review that is self annotated by humans.	The semi-supervised SASI algorithm was used to model Amazon texts comments and review data, the accuracy produced by the model was 81%, and precision of 77%, and recall results 83.1%..	3 different evalution metrics were modelled like phrase based, syntax based and neural approach
Tsur et. al.	Association for Computational Linguistic	a pytorch architecture based on neural network named multi-head self-attention	different kinds of datasets	The researchers classified the material into sardonic and non-sarcastic categories using deep learning techniques and the PyTorch library, reaching state-of-the-art results. They claimed that people are less adept at recognizing sarcasm than basic classification models.	Unique, cutting-edge semi-supervised method A very big dataset dimension of 66000 reviews has been employed with SASI.
González-Ibáñez et. al.	Association for Computational Linguistics	MHA Neural Network	Multiple datasets	Implemented PyTorch library to detect sarcasm in the text from emoticons and also highlight the words responsible for sarcastic sentence.	First ever analysis of emoticons in model was conducted
Kumar et al.	IEEE Access	SVM as a statistical Model and MHA-BiLSTM and BiLSTM as deep learning models	SARC Reddit data that consists of 1246058 comments across different for forums	The researchers utilized two approaches , SVM and MHA-BiLSTM, maintaining the data unbalanced to reflect the actual circumstance. The MHA-BiLSTM model performed much better than all other models.	Auxiliary feature that is manually generated was included
Ghanem et.al	Springer	Deep learning and Ensemble Machine learning algorithm	Twitters tweet data with 100000 total tweet	executed some few ensemble machine learning techniques as well as the deep learning algorithm both alone and in a stacking fashion. Finally, it was shown that, contrary to most studies, machine learning algorithms can identify text sarcasm just as well as deep learners when employed in a stacking fashion to improve performance.	Demonstrated that deep learning methods with high computing power can perform just as well as machine learning models with lower computing capability.
Davidov et. al.	Association for Computational Linguistics	K-Nearest Neighbour (K-NN) was evaluated on SCUBA metrics	Twitter data and Ratings from amazon data.	A semi-supervised system was used to identify sarcasm on the text data utilizing psychological and behavioural characteristics and taking past tweets into consideration while retaining SCUBA as the evaluation criteria and achieving score above the benchmark.	For the initial time, users' prior tweets had taken into account during modeling, which boosted accuracy and efficiency..
Huang, YH et.al	Springer	NN Architecture and models based on features	Twitter dataset for each language mentioned in the research	Feature-based models and neural network topologies for irony identification were compared for multilingual (French, English, and Arabic) and multicultural (Indo-European vs. less culturally similar languages) languages. embedding and CNN models were used to create a corpus from three separate datasets. Following some testing, it was shown that multilingual techniques may be used for irony identification.	First ever sarcasm detection on multiple languages
Mishra et. al.	Association for Computational Linguistics	Attention mechanism incorporating LSTM in a Hybrid NN Architecture	Self-Prepared and annotated news headline dataset	They generated an own news dataset using DNN LSTM models and ran grid search hyper parameter tweaking after discovering that most of the time, twitter's noisy data is utilized to identify sarcasm. The result was 89.88% , which is more than 5%	Unique dataset they prepared by themselves from sarcastic news website and traditional news website
Potamias et.al	CoRR	CNN, RNN and Attentive RNN	contains 198041 sarcastic tweets and and 197917 normal tweets with no sarcasm involved	Demonstrated that the desired job may be predicted using deep learning approaches as opposed to feature engineering methods. The researchers demonstrate that Attentive RNN excels in sarcasm, irony, and other common linguistic patterns.	
Kumar and Anand	Association for computational linguistic	BERT RoBERTa spanBERT	Twitter Corpus and Reddit Corpus by Khodak et al. (2017)	Two sentence based data were created and used like Single sentence classification task and sentence pair classification task	The pre- processing or hyper parameter tunning carried out was not dataset specific but general to any data
Wolf et. al.	CoRR	NA	NA	Mentioned the advantages of pre-trained models in transformer library and accessing it through hugging face	Makes the new machine learning enthusiast familiar to transformer and how to use it
Gonzalez-Carvajal	CoRR	ERT, TF-IDF, Logistic Reg., Linear SVC, Naive Bayes, auto ML	IMDB data	BERT model is compared with TF-IDF, LR, Naive Bayes, Linear SVC, auto ML, etc. and BERT comes out to be the best model	An empirical support with proof has been provided for making BERT as the default model for NLP task

2.12 Conclusion

After critically reviewing the above said research papers it can be depicted that data specific pre-processing cannot be a mandate while applying BERT for sarcasm detection as we have seen in the research done by Kumar and Anand (2020) where not just pre-processing but hyper parameter tunning as well was generalized. Also the researches conducted by González-Carvajal and Garrido-Merchán (2020) and Ghanem et al. (2020) proved BERT and deep learning models to be the best performing models over any other machine learning models scoring accuracy above 90%.

The most unique research was done by Huang et al. (2017) in which not just English but French, Arabic languages were also incorporated to detect sarcasm and concluded to be a great success with high performing models.

Lastly, out of all, traditional machine learning algorithm showed a similar performance to deep learning models on the basis of accuracy when incorporated with ensemble technique.

3 Methodology

The "*Sarcasm Detection and sentiment analysis methodology*" was adopted for the implementation along with the most renowned and industry-proven *CRISP-DM* methodology that is known for an organized approach to designing a data mining project. A pictorial format can be seen in figure 1.

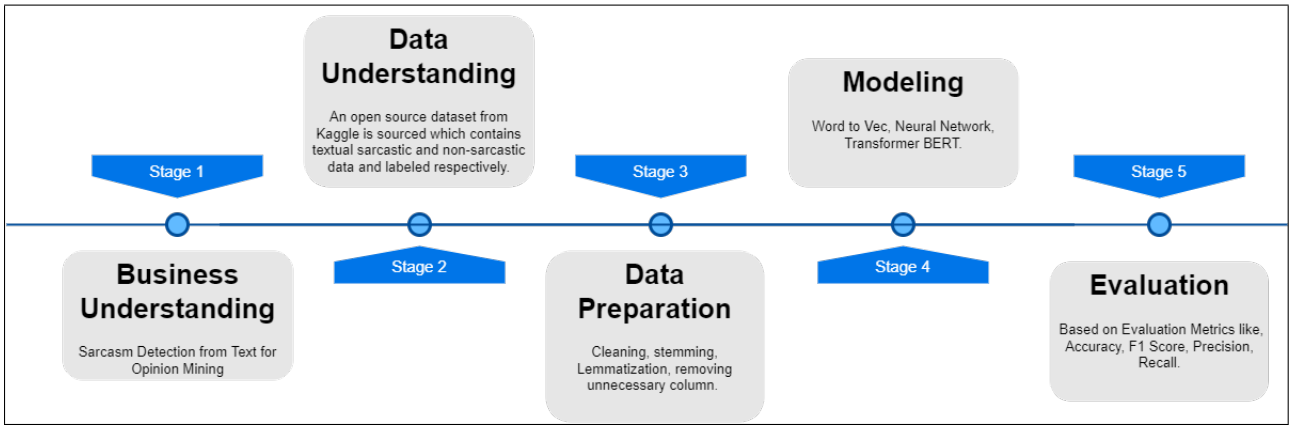

Figure 1: CRISP-DM Methodology

3.1 Data: Selection and Acquisition

In order to predict sarcasm it is necessary to have a realistic sarcastic data. The dataset utilized for this project is the *News Headline* dataset from the kaggle website Mishra (2022), which was compiled from *TheOnion*, a website that publishes satirical news. The dataset includes 26710 total observations as well as the three Features Headline, Sarcastic (binary feature), and Link (link of the news). The dataset is then imported in the Jupyter Notebook environment where it is taken up for further analysis. The reason to choose the sarcastic news dataset is because of its authenticity of the text containing sarcastic and non-sarcastic.

3.2 Data Pre-processing and Transformations

After the dataset was imported in the Jupyter Notebook all the data pre-processing pipelines were followed.The important libraries that were used are *pandas*, *numpy*, *NTLK*, The data was read using *pandas* library to read json file as the data is in a json format. The primary analysis of the data included finding the number of total words, value counts for sarcastic and non-sarcastic labels. After finishing a basic exploratory analysis the data was cleaned using *NLP* pipelines like removing stop words, punctuation's, etc. and the cleaned data was then split into sarcastic and non-sarcastic data frames for understanding.

Table 2: Snippet of the Data
(Misra and Grover; 2021)

Article_link	Headline	Is_sarcastic
https://www.huffingtonpost.com/entry/versace-black-code_us_5861fbefe4b0de3a08f600d5	former versace store clerk sues over secret 'black code' for minority shoppers	0
https://politics.theonion.com/top-snake-handler-leaves-sinking-huckabee-campaign-1819578231	top snake handler leaves sinking huckabee campaign	1

3.3 Modelling

For the purpose of sarcasm detection a *WordToVec*, neural network model has been implemented which served as the base models and a novel model which is a transformer based *BERT* Classification model is compared to the base models and depending on the classification metrics the best model is selected and further tuned. The data is pre-processed in such a way that each word is converted into a numerical vector for the implementation of word to vec model using *gensim* library. After the words are vectorised, tokanization is applied using the *NLTK* library. Once this is complete the data is split into train and test and the *wordtovec* model is trained.

For the purpose of sentiment analysis *NLTK* library *VADER* was implemented which gave the sentiment behind the sentences on which it was been applied and accordingly the way of expressing sarcasm was concluded.

3.4 Evaluation

For sarcasm detection the classification model developed were evaluated based on the classification metrics like accuracy, precision, recall and F1 score and the model that gives the best classification metrics was selected as the best model and tested on a test data. Similarly for *VADER* sentiment analysis, sarcasm with respect to sentiment chart was prepared which reveled the most frequently found sentiment while using sarcasm.

4 Design and Implementation

As shown in the figure 2 above this research followed the same process flow for the implementation which is the CRISP-DM methodology

4.0.1 Data Preparation Phase

The phase of data preparation consisted of selection of the right data set which matches the requirement of the final objective, data acquisition, data cleaning and data pre-processing.

The dataset selected and used in this research is the *News Headline* dataset Mishra (2022) downloaded from the kaggle website and extracted from the zip file. The extracted zip file was then imported in the python environment using Jupyter notebook IDLE using a python library pandas. The original format of the data is in a JSON format hence, the JSON format was read using pandas library and presented as a CSV which made it easy to understand. The dataset consisted of 3 columns *is_sarcastic, Headline and article_link* but this research demanded only 2 columns majorly which are *headline* and *is-sarcastic* where *headline* is the news which is present in a string format and *is-sarcastic* is a binomial attribute which has 1 and 0 here 1 denotes the text as sarcastic and 0 denotes the text as non-sarcastic. After finalizing the data the target variable is visualized for unbiased split of sarcastic and non sarcastic labels and it has been found that the labels are split equally.

As the data used is a textual data it contains many text or words which are supposed to be removed for the analysis purpose like punctuation's, prepositions, stop-words, etc. which are present repetitive in the data hence, a function *clean-text* is created to which the textual data is passed and the entire data cleaning has been performed.

After performing the necessary cleaning the data was further pre-processed according to the model. The first model used is the *WordToVec / GloVe* model for which the all the strings are split into single words using *.split* function present in the pandas library and then each word is converted to a numerical value called as vector using the gensim library and the total count of words generated and converted into vector is 37149. After word vectorization the data is further tokenized where each word is denoted with a unique symbol which helps in retaining data security. Next, *get-weigth-matrix* function is made to produce a weight matrix from the *Word2Vec Gensim* model and *Word2Vec* embedding vectors, which is then used as the weights of the non-trainable Keras hidden layer. Finally, all of the data preparation and cleaning is finished, and the data is sent for modeling.

4.0.2 Modelling Phase

In this part of the research the selected models which is *GloVe* including *LSTM* and Transformer *BERT*, were implemented on the cleaned and pre-processed data.

First, the *wordtovec* data is passed through a neural network build which consists of two bi-directional layers in which the first layer is a Long Short Term Memory layer *LSTM* and the next layer is a *GRU*. On top of this layers this neural network starts with an embedding layer which works as a one-hot encoding for the textual data along with dimensionality reduction Finally the network ends with a dense layer which helps in connecting all the input neurons to each other and with a *sigmoid* activation function as the output of this research is a binomial classification. The build model is then compiled incorporating the *adam* optimizer with a learning rate of 0.01, loss being binary-crossentropy and keeping attention on the metric as accuracy.

For the implementation of *BERT* model it starts with installing the required libraries like transformers and tensorflow following to which is utilized keras library pre-trained pre-processing layer and encoding layer. In the pre-processing layer the textual data is passed as input to the pre-processing layer. The pre-processed text is then passed

to the pre- trained encoding layer which will serve as a parameter for the keras model. Following this *BERT* layer which is built of 3 layers, a neural network layer with 2 layer was constructed which has a dropout layer and a dense layer. The dropout layer is supposed to output a pooled output which is than passed on to the 2nd layer which is a dense layer which contains *sigmoid* activation function which serves as the best activation function for any classification task. Finally, the entire model is compiled using an adam optimizer with loss being binary cross entropy and keeping track of the metrics which are accuracy, precision and recall. This entire model was than trained with 10 and 20 epochs with early stopping which keeps track of the accuracy metrics and stops the model implementation if the accuracy is seen to drop.

Coming to the modelling of VADER for text sentiment analysis, important libraries were imported like *SentimentIntensityAnalyzer* which was imported from *vaderSentiment* library and initialized under name *'analyser'*. As *VADER* gives output in the range of (-1) to (1) as discussed above 1, a simplification was done is such a way that if the sentiment score of a text is greater than or equal to 0.05 then that text will be considered as positive, if the sentiment score of a text is less then or equal to -0.05 than that particular text will be termed as negative and the text which does not fall in any of the conditions will be neutral text. This way a clear understanding of the sentiment was built. Finally, all the above created conditions and the *VADER* algorithm was modelled on the previously cleaned data and a dataframe was created as an output which had an added column named *'sentiment'* which gave sentiment score for each textual datapoint.

4.0.3 Application Phase

In this phase the two models were evaluated based on the metrics they perform like accuracy, precision, recall and F1 score. Also, a string of sarcastic and non-sarcastic sentences were passed to the neural network to evaluate the working of the model. Also, the sentiment analysis model was grouped with the sentiment score obtained and the relation of it with sarcastic sentences.

5 Evaluation

5.1 Experiement 1: GloVE

The Glove based neural network model with 4 layers having Bi-directional LSTM layer which includes 128 units, Bi-directional GRU layer which includes 32 units and a dense layer with sigmoid activation function all compiled using an adam optimizer having learning rate of 0.01 and metrics to measure given as accuracy. This neural network was trained on the 37149 words from the headline dataset.

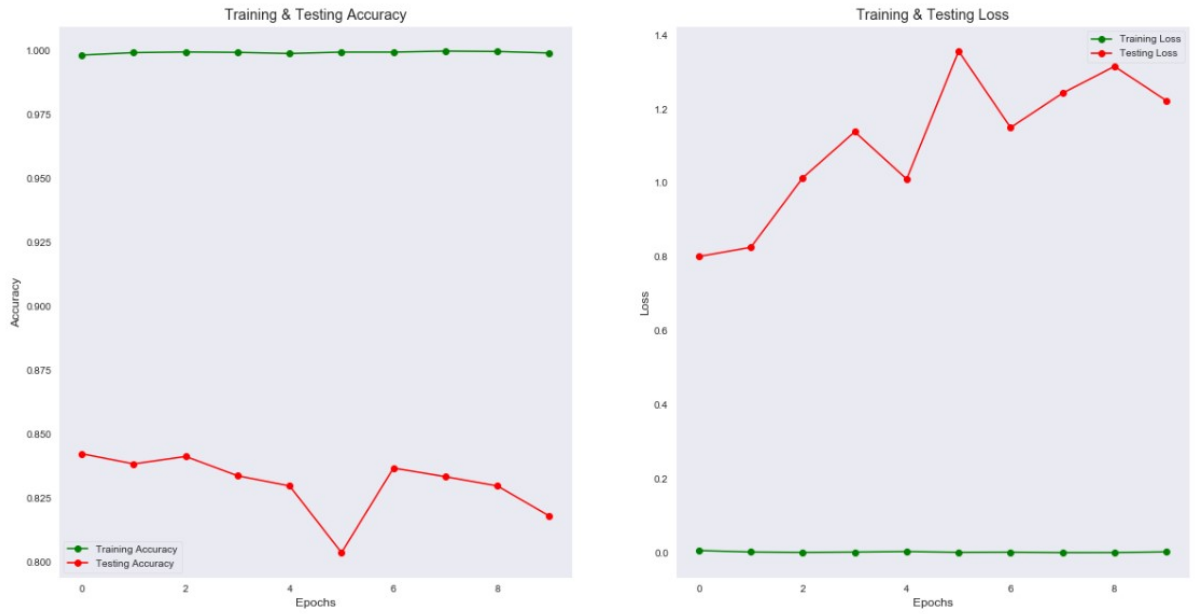

Figure 2: Glove

The complete model was trained for 10 epochs on the training data keeping x_test and y_test as the validation data. The final training accuracy and validation accuracy scored by this model was 99.92% and 81.81% respectively where as the final training loss and validation loss scored by this model was 0.0027 and 1.2227 respectively 2.

5.2 Experiment 2: BERT with 10 Epochs

The transformer based BERT model was built which utilized keras library pre-trained pre-processing layer and encoding layer. In the pre-processing layer the text in the data is passed as pre-processing layer input. The pre-processed text is then passed to the pre-trained encoding layer which will serve as a parameter for the keras model. Following this BERT layer which is built of 3 layers as discussed, a neural network layer with 2 layer was constructed which has a dropout layer and a dense layer. The dropout layer is supposed to output a pooled output which is than passed on to the 2nd layer which is a dense layer which contains sigmoid activation function which serves as the best activation function for any classification task. Finally, the entire model is compiled using an adam optimizer with loss being binary cross entropy and keeping track of the metrics which are accuracy, precision and recall. This entire model was than trained with 10 epochs having validation data as x_test and y_test with an earlystopping parameter which is suppose to monitor accuracy in such a way that that while training when the accuracy is reached to maximum and if the accuracy is seen to drop thrice as compared to the maximum accuracy the model will stop its training. The training and validation accuracy scored by this model after 10 epochs was 77.03% and 77.88% respectively and the final training and validation loss scored was 0.4844 and 0.4075 respectively 3

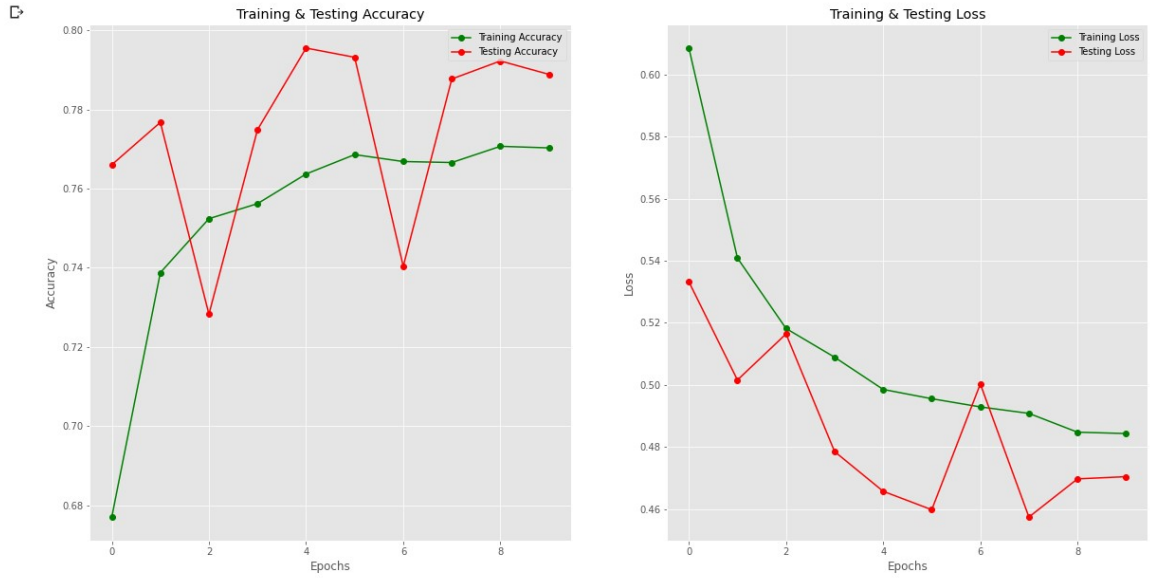

Figure 3: BERT 10

5.3 Experiment 3: BERT with 20 Epochs

While we discussed earlier that the BERT model with 10 epochs scored training accuracy of 77% having early stopping parameter set to monitor accuracy. It was noted that the accuracy of the model kept increasing with each epoch, but the limit of epochs been only 10 the model did not score beyond 77%. Hence the same BERT model was further accessed with 20 epochs keeping all the parameter same.

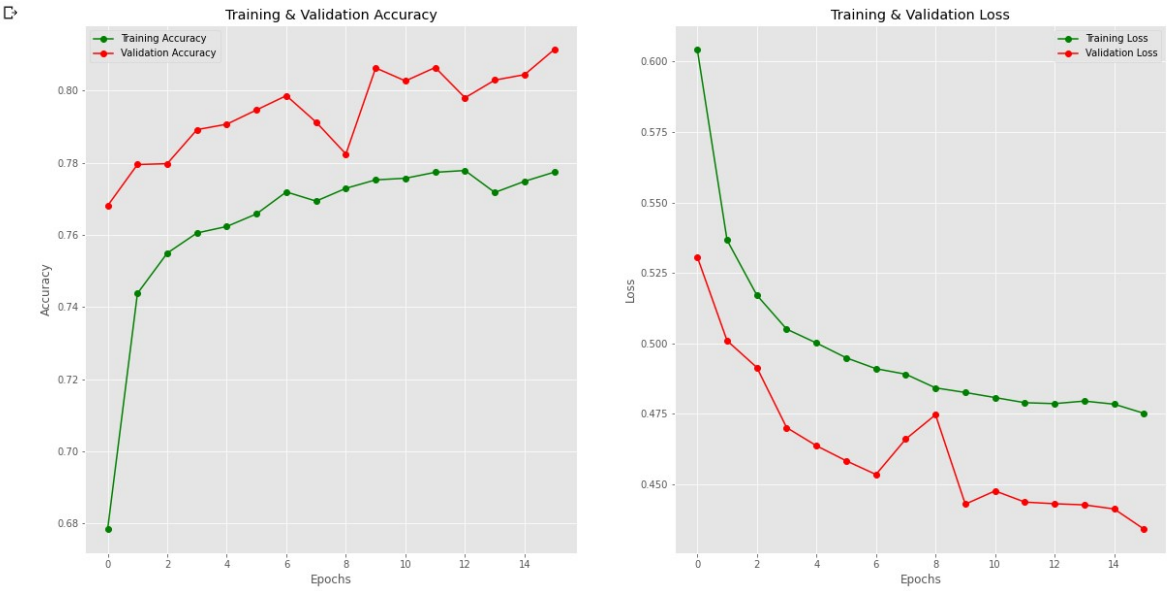

Figure 4: BERT 20

This time the model stopped at 16th epoch because the accuracy stopped going further up but improved its training and validation accuracy and training from 77.74% and 81.14% respectively to 77.79% and 81.33% respectively. Similarly the training loss and validation loss also dropped down from 0.4844 and 0.4075 respectively to 0.4752 and

0.4341 4.

5.4 Experiment 4: Testing BERT model on test data

Since from the above experiments it was evident that with 20 epochs the BERT model gives more accuracy when compared to the 10 epochs. The *test_data* was predicted on that model utilizing the *sklearn .predict* function on the test data and a confusion matrix was created.

```
[41] mat = confusion_matrix(Y_test_datat, y_predicted)
     labels = ['Sarcastic ', 'non-sarcastic']
```

```
sns.heatmap(mat, square=True, annot=True, fmt='d', cbar=False, cmap='Blues',
            xticklabels=labels, yticklabels=labels)

plt.xlabel('Predicted label')
plt.ylabel('Actual label')
```

```
Text(91.68, 0.5, 'Actual label')
```

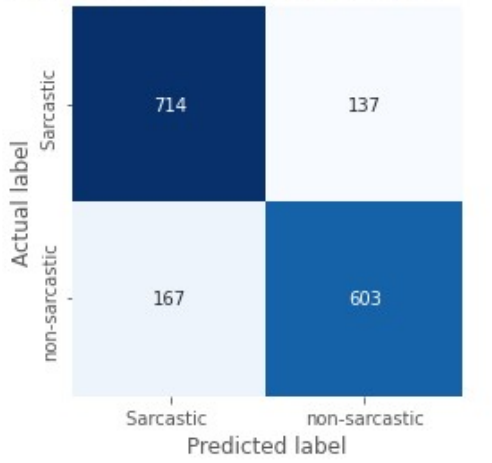

Figure 5: Confusion matrix of test data for BERT 20

The classification report generated using the predicted data and actual y_test data gave an accuracy of 81% with average precision of 81%.

```
[44] print(classification_report(Y_test_datat, y_predicted))
```

	precision	recall	f1-score	support
0	0.81	0.84	0.82	851
1	0.81	0.78	0.80	770
accuracy			0.81	1621
macro avg	0.81	0.81	0.81	1621
weighted avg	0.81	0.81	0.81	1621

Figure 6: Classification report for BERT 20

5.5 Experiment 4: VADER Sentiment Analysis

As mentioned in section 4.0.2 a model named *VADER* has been implemented for sentiment analysis of the textual data that contains sarcastic and non-sarcastic sentences. This model measures sentiment on the scale of (-1) to (1) where (-1) denotes extremely negative sentence and (1) denotes extremely positive sentence. Hence to simplify and not keep the score between -1 and 1, the score of a text greater than or equal to 0.05 is considered as positive sentiment text, the sentiment score of a text less than or equal to -0.05 is considered as negative sentiment text and the text which does not fall in any of the conditions said above will be neutral sentiment text. The results of *VADER* can be seen in the figure 7 below where a column named *sentiment* is created which gives the prediction of the sentiment behind every sentence.

```
In [56]: 1 data.head(10)
```

Out[56]:

	is_sarcastic	headline	sentiment
0	1	thirtysomething scientists unveil doomsday clo...	Negative
1	0	dem rep. totally nails why congress is falling...	Negative
2	0	eat your veggies: 9 deliciously different recipes	Positive
3	1	inclement weather prevents liar from getting t...	Negative
4	1	mother comes pretty close to using word 'strea...	Positive
5	0	my white inheritance	Neutral
6	0	5 ways to file your taxes with less stress	Negative
7	1	richard branson's global-warming donation near...	Negative
8	1	shadow government getting too large to meet in...	Neutral
9	0	lots of parents know this scenario	Neutral

Figure 7: Sentiment Score from VADER model.

From the output of this experiment it was found out that the count of negative sentiment with respect to non-sarcastic sentences is 4781 and that with the sarcastic sentences is 4653 as it can be seen in the figure. 8.

```
In [58]: 1 #sarcasm wise negative sentiment count
         2 negative_count = data.groupby('is_sarcastic')['sentiment'].apply(lambda x: (x=='Negative').sum()).reset_index(name='Negative_count')
         3
         4 #view results
         5 print(negative_count)

   is_sarcastic  Negative_count
0             0            4781
1             1            4653
```

Figure 8: Count of Negative Score

The count of positive sentiment with respect to non-sarcastic sentences is 4668 and that with the sarcastic sentences is 4271 as it can be seen in the figure. 9.

Figure 9: Count of Positive Score

Lastly, count of neutral sentiment with respect to non-sarcastic sentences is 5536 and that with the sarcastic sentences is 4710 as it can be seen in the figure. 10.

Figure 10: Count of Neutral Score

Therefore, it is clearly evident that most of the time sarcasm comes with a negative sentiment

5.6 Discussion

The study and evaluation shows that the first experiment *GloVe* model showed the best training accuracy of 99.92% and 81.81% of validation accuracy but, it also had high number of training and validation loss which visualized in figure. 2 which concludes that this model shows high accuracy with few small errors but also has high loss with few big errors. Hence, this model cannot be relied for the prediction.

The second and the third experiment which involved *BERT* model with 10 and 20 epochs respectively. These models scored training and validation accuracy of 77% and 77.88% with 10 epochs and 77.74% and 81.14% with 20 epochs. Also the loss measured was 0.4844 with 10 epochs and 0.4752 with 20 epochs. Therefore, *BERT* model showed a consistent increase in accuracy and consistent drop in the loss with respect to epochs and hence can be said to be a reliable model for the future sarcasm prediction.

As *BERT* was chosen to be the best model it was further tested on the test data name *Y_test_datat* and the prediction accuracy came out to be 81%. with 81% precision for *sarcastic* as well as *non-sarcastic labels* independently.

6 Conclusion and Future Work

This research proposed and implemented sarcasm detection incorporating *BERT* machine learning algorithm. This research have implemented *BERT* on the *Sarcasm news* dataset along with sentiment analysis of the same. All the modelling pipelines were thoroughly followed during the implementation of this research which involved data preparation, feature engineering, modelling and evaluation.

The data preparation part involved pre processing and cleaning of the data like cleaning of the text, removing all the stopwords, punctuation's, tokenization, lemmatization. The implementation involved modelling of 3 models which were *Glove*, *BERT with 10 epochs* and *Bert with 20 epochs* following to which a sentiment analysis model *VADER* was implemented. This research gave accuracy very near to the research done before González-Carvajal and Garrido-Merchán (2020) and was novel in terms of applying an algorithm to a the dataset as *BERT* was never been implemented on the *News Headline* data. Also, this research found out the sentiment behind the text when someone is being sarcastic and it was evaluated that most of the time when a sarcastic statement is made it followed a '*Negative*' sentiment. This conveys that whenever a sarcastic statement is made online in terms of reviews or news it can mostly be complains or disapproval of the related topic.

Some weakness or limitations noted in the models which can be addressed in the future is that the *BERT* models shows a very smooth and continuous graph of training accuracy and training loss but testing accuracy and testing loss graphs shows sudden drastic fluctuations as it drops its accuracy with increase in epochs and again improves as the epochs move forward.

References

Akula, R. and Garibay, I. (2021). Interpretable multi-head self-attention model for sarcasm detection in social media, *CoRR* **abs/2101.05875**.

Amir, S., Wallace, B. C., Lyu, H., Carvalho, P. and Silva, M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media, *CoRR* **abs/1607.00976**.

Chaffey, D. (2022). Global social media statistics research summary 2022, *Website* . **URL:** *https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/: :text=More%20than%20half%20of%20the,social%20media*

Ghanem, B., Karoui, J., Benamara, F., Rosso, P. and Moriceau, V. (2020). Irony detection in a multilingual context, pp. 141–149.

González-Carvajal, S. and Garrido-Merchán, E. C. (2020). Comparing BERT against traditional machine learning text classification, *CoRR* **abs/2005.13012**. **URL:** *https://arxiv.org/abs/2005.13012*

González-Ibáñez, R., Muresan, S. and Wacholder, N. (2011). Identifying sarcasm in Twitter: A closer look, pp. 581–586. **URL:** *https://aclanthology.org/P11-2102*

Huang, Y.-H., Huang, H.-H. and Chen, H.-H. (2017). Irony detection with attentive recurrent neural networks, pp. 534–540.

Joshi, A., Sharma, V. and Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection.

Khodak, M., Saunshi, N. and Vodrahalli, K. (2017). A large self-annotated corpus for sarcasm, *CoRR* **abs/1704.05579**. **URL:** *http://arxiv.org/abs/1704.05579*

Kreuz, R. J. and Glucksberg, S. (1989). How to be sarcastic: The echoic reminder theory of verbal irony, *Journal of Experimental Psychology: General* **118**(4): 374–386.

Kumar, A. and Anand, V. (2020). Transformers on sarcasm detection with context, *Proceedings of the Second Workshop on Figurative Language Processing*, Association for Computational Linguistics, Online, pp. 88–92. **URL:** *https://aclanthology.org/2020.figlang-1.13*

Kumar, A., Narapareddy, V. T., Aditya Srikanth, V., Malapati, A. and Neti, L. B. M. (2020). Sarcasm detection using multi-head attention based bidirectional lstm, *IEEE Access* **8**: 6388–6397.

Mishra, R. (2022). News headlines dataset for sarcasm detection. [online], *Website* .

Misra, R. and Arora, P. (2019a). Sarcasm detection using hybrid neural network, *arXiv preprint arXiv:1908.07414* .

Misra, R. and Arora, P. (2019b). Sarcasm detection using hybrid neural network, *CoRR* **abs/1908.07414**.

Misra, R. and Grover, J. (2021). *Sculpting Data for ML: The first act of Machine Learning.*

Mohammad, S., Salameh, M. and Kiritchenko, S. (2016). How translation alters sentiment, *J. Artif. Intell. Res. (JAIR)* **55**: 95–130.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation, pp. 311–318.
URL: *https://aclanthology.org/P02-1040*

Peled, L. and Reichart, R. (2017). Sarcasm SIGN: Interpreting sarcasm with sentiment based monolingual machine translation, pp. 1690–1700.
URL: *https://aclanthology.org/P17-1155*

Potamias, R. A., Siolas, G. and Stafylopatis, A. (2019). A transformer-based approach to irony and sarcasm detection, *CoRR* **abs/1911.10401**.

Tepperman, J., Traum, D. and Narayanan, S. (2006). Yeah right: Sarcasm recognition for spoken dialogue systems.

Tsur, Oren Davidov, D. . R. A. (2010). Semi-supervised recognition of sarcastic sentences in online product reviews.

Versaci, Mario, S. K. P. A. (2020). Multi-rule based ensemble feature selection model for sarcasm type detection in twitter, *Hindawi* **abs/1704.05579**.
URL: *https://doi.org/10.1155/2020/2860479*

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing, *CoRR* **abs/1910.03771**.
URL: *http://arxiv.org/abs/1910.03771*