SARCASM DETECTION IN TEXT USING DEEP NEURAL NETWORKS

GİZEM GÜMÜŞÇEKİÇCİ

IŞIK UNIVERSITY
JANUARY, 2024

SARCASM DETECTION IN TEXT USING DEEP NEURAL NETWORKS

GİZEM GÜMÜŞÇEKİÇCİ
Işık University, School of Graduate Studies, Computer Science Engineering Master Program, 2024

Submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering

IŞIK UNIVERSITY
JANUARY, 2024

IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COMPUTER SCIENCE ENGINEERING MASTER PROGRAM

SARCASM DETECTION IN TEXT USING DEEP NEURAL NETWORKS

GİZEM GÜMÜŞÇEKİÇCİ

APPROVED BY:

Asst. Prof. Rahim Dehkharghani (Thesis Supervisor)	Kadir Has University
Asst. Prof. Emine Ekin	Işık University
Asst. Prof. İlknur Karadeniz	Özyeğin University

APPROVAL DATE: 31./01./2024

SARCASM DETECTION IN TEXT USING DEEP NEURAL NETWORKS

ABSTRACT

Sarcasm is a form of irony which is generally used in expressing negative opinions. Sarcasm poses a linguistic challenge due to its figurative nature where intended meaning contradicts with literal interpretation. Sarcasm is widely used in our daily lives and also upon many social platforms. Detecting sarcasm in written text is a challenging process that has captured the interest of many researchers. Hence, sarcasm has become a crucial task in the Natural Language Processing (NLP) field. This thesis study explores the concept of sarcasm, and its importance on existing sarcasm research. The automatic process of sarcasm detection involves dataset selection, pre-processing steps, and selecting proper approaches, including rule-based methods, Machine Learning (ML), Deep Learning (DL) and Transformer architectures. This study surveys previous research on sarcasm detection, specifically examining the dataset, methodology and performance. This thesis study attempts to automatically detect sarcasm by utilizing various ML, DL and transformer and hybrid neural network architectures on news headlines datasets. To overcome the dataset and performance limitations on existing approaches, we propose various methodologies to detect sarcastic text mostly focusing on DL, hybrid neural networks and transformer architectures. We combine appropriate architectures with several hand-crafted features and utilizing different word embedding models. To further extend the performance of our proposed methods and also enhance the existing news headlines dataset, we proposed several modifications. We contribute to the existing dataset by applying augmentation to increase the dataset size to help enhance the performance of the proposed models with overcoming dataset limitations. Our methodologies correctly identify sarcasm with 97.68% F1 score.

Key words: Sarcasm, News Headlines, Sarcasm Classification, Transformers, Text Augmentation

DERİN SİNİR AĞLARI KULLANARAK METİN İÇİNDE ALAYCILIK TESPİTİ

ÖZET

Alaycılık, genellikle olumsuz görüşlerin ifade edilmesinde kullanılan bir ironi biçimidir. Alaycılık, amaçlanan anlamın gerçek yorumla çeliştiği mecazi doğası nedeniyle dilsel bir zorluk teşkil etmektedir. Alaycılık günlük yaşamımızda ve birçok sosyal platformda yaygın olarak kullanılmaktadır. Yazılı metinlerde alaycılığın tespit edilmesi birçok araştırmacının ilgisini çeken zorlu bir süreçtir. Dolayısıyla alaycılık, Doğal Dil İşleme (NLP) alanında çok önemli bir görev haline geldi. Bu tez çalışması alaycılık kavramını ve bu kavramın mevcut alaycılık araştırmaları üzerindeki önemini incelemektedir. Otomatik alaycılık algılama süreci, veri kümesi seçimini, ön işleme adımlarını ve kural tabanlı yöntemler, Makine Öğrenimi (ML), Derin Öğrenme (DL) ve Transformer mimarileri dahil olmak üzere uygun yaklaşımların seçilmesini içerir. Bu çalışma, özellikle veri kümesini, metodolojiyi ve performansı inceleyerek alaycılığın tespitine ilişkin önceki araştırmaları incelemektedir. Bu tez çalışması, haber başlıkları veri seti üzerinde çeşitli ML, DL ve transformatör ve hibrit sinir ağı mimarilerini kullanarak alaycılığı otomatik olarak tespit etmeye çalışmaktadır. Mevcut yaklaşımlardaki veri kümesi ve performans sınırlamalarının üstesinden gelmek için, çoğunlukla DL, hibrit sinir ağları ve transformatör mimarilerine odaklanan alaycı metinleri tespit etmek için çeşitli yöntemler öneriyoruz. Uygun mimarileri, farklı kelime temsil modellerini kullanarak çeşitli el yapımı özelliklerle birleştiriyoruz. Önerilen yöntemlerimizin performansını daha da genişletmek ve mevcut haber başlıkları veri setini geliştirmek için çeşitli değişiklikler önerdik. Önerilen modellerin performansının veri kümesi sınırlamalarının üstesinden gelmesine yardımcı olmak amacıyla veri kümesi boyutunu artırmak için büyütme uygulayarak mevcut veri kümesine katkıda bulunuyoruz. Metodolojilerimiz alaycılığı %97,68 F1 puanıyla doğru bir şekilde tespit edebiliyor.

Anahtar Kelimeler: Alaycılık, Haber Manşetleri, Alaycılık Sınıflandırması, Transformers, Metin Arttırma

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my thesis advisor Asst. Prof. Rahim Dehkharghani, for his invaluable guidance and support throughout the process of completing my thesis. His expertise, encouragement, and dedication have been greatly affected in shaping the outcome of this thesis. I am truly grateful for the knowledge and inspiration we have shared, which have been beneficial in my academic journey.

I also would like to thank all of my instructors especially I want to express my sincere appreciation to Asst. Prof. Emine Ekin for her encouragement and instructions during the completion of this thesis. I am deeply grateful for the knowledge and skills she has imparted, significantly contributing to my development to be a better researcher. Her mentorship has been invaluable, and I am truly thankful for her guidance throughout my academic process.

Finally, I express my gratitude to my family for their support and encouragements throughout my life. Their dedication and selflessness have facilitated my educational journey. I especially thank my father and grandmother for their excitement and heartfelt encouragement for my academic career.

Gizem GÜMÜŞÇEKİÇCİ

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NLP: Natural Language Processing

ML: Machine Learning

DL: Deep Learning

BoW: Bag of Words

LSTM: Long Short-Term Memory

LR: Logistic Regression

MTL: Multitask Learning

AUC: Area under the ROC Curve

ROC: Receiver Operating Characteristic Curve

GRU: Gated Recurrent Unit

CNN: Convolutional Neural Network

GloVe: Global Vectors for Word Representation

TF-IDF: Term Frequency - Inverse Document Frequency

BERT: Bidirectional Encoder Representations from Transformers

RoBERTa: A Robustly Optimized BERT

NB: Naïve Bayes

kNN: K-Nearest Neighbors

SVM: Support Vector Machine

RF: Random Forest

NLTK: Natural Language Toolkit

ReLu: Rectified Linear Units

MV: Majority Voting

AI: Artificial Intelligence

DNN: Deep Neural Network

CHAPTER 1

1. INTRODUCTION

1.1 Introduction to Sarcasm

Sarcasm is defined as a form of irony that is intended to express concepts such as mockery, contempt and ridicule and many more. Sarcasm has a figurative nature meaning the intended meaning of words is opposite to their literal interpretation (Joshi et al. 2017). For instance, a sentence can have a positive surface sentiment, but the intended meaning can be negative. To illustrate, in the text “Oh, fantastic! I absolutely love it when my computer decides to crash right in the middle of an important presentation. It adds such a thrilling element to my day.” The event of the computer breaking down during a presentation is a negative occurrence but in the sentence, it is stated as a positive event. There is an obvious sarcasm that can be detected here: the opposite meaning is actually intended. In this example, the sarcasm is obvious, but in most cases, identification can be more challenging. For instance, another example would be the sentence “You're a real genius, aren't you?” The use of "real" adds a layer of sarcasm to the sentence.

Sarcasm is related to many concepts such as irony, deception, metaphor, and humor. Sarcasm and all the concepts mentioned here are highly used among society. It is commonly used on a regular basis in many different areas. Sarcasm can be found in news, movies, literature, daily conversations, marketing, politics and so on. And with the increasing use of social media, sarcasm becomes more prevalent in our lives especially on major social media platforms such as Twitter, Reddit etc. which makes the identification of these concepts an issue in the Natural Language Processing (NLP) field. Sarcasm can have many forms and be interpreted differently by anyone. In other

words, everyone's perspective can vary, due to this reason the sarcastic features can be evaluated differently by anyone. All these reasons make the detection of sarcasm a challenging task in the NLP field.

1.2 Sarcasm in Social Platforms

Sarcasm, recognized as a form of figurative language, is used in many different areas in our lives. We highly use sarcasm to express ourselves or we are being exposed to it regularly for different reasons. In sarcasm the usage of humor, metaphors and such similar concepts are highly common to increase the effectiveness (magnitude) of the proposed message while it adds an additional challenge in the process of detecting sarcasm. Sarcasm is considered a linguistic issue because it involves the use of language and linguistic elements to convey a specific meaning. Sarcasm, as a linguistic phenomenon, engages various linguistic features to be identified. Nonetheless, due to its extensive usage, sarcasm has become a popular research topic in recent years. Consequently, the popularity of sarcasm detection has grown immensely in recent years and it grows annually.

Sarcasm is incorporated into our daily lives and is also highly utilized mostly in social media, news domains, marketing campaigns, politics etc. to alter the perception of society. With the recent advancements in online services in commerce, tourism, and business, the companies in these areas are eager to involve sentiment and sarcasm analysis into their marketing strategies to attract the attention of eligible consumers (Lahaji et al. 2023). According to the statistics published in 2014 (Weiguo and Michael, 2014), social networking can be considered as the most popular online activity and 91% of online adults use social media sites on a regular basis. In regard to another similar study, a data analysis research has been conducted on adolescents aged 12-15 years (Kira et al., 2019). The result of this study states that 97% of these online adolescent users regularly access and spend hours on these social media sites. Individuals with diverse age ranges actively engage with social media platforms to mainly communicate or post about opinions, facts, events, ideas and humor, and these posts are often accompanied by images, emotes, or videos. Social media platforms Facebook, YouTube, and Twitter are the most popular sites visited by millions of people each day (Weiguo and Michael, 2014). Likewise, other platforms such as Instagram, Reddit etc. have noticeable popularity among other social platforms as well.

Consequently, detecting sentiment, opinions, feelings, and sarcasm on online resources has become a subject of extensive research in NLP. Figure 1 illustrates the search results from Google Scholar for Sarcasm detection from 2009 to 2021. As shown in Figure 1, there has been a consistent increase in sarcasm research since 2012.

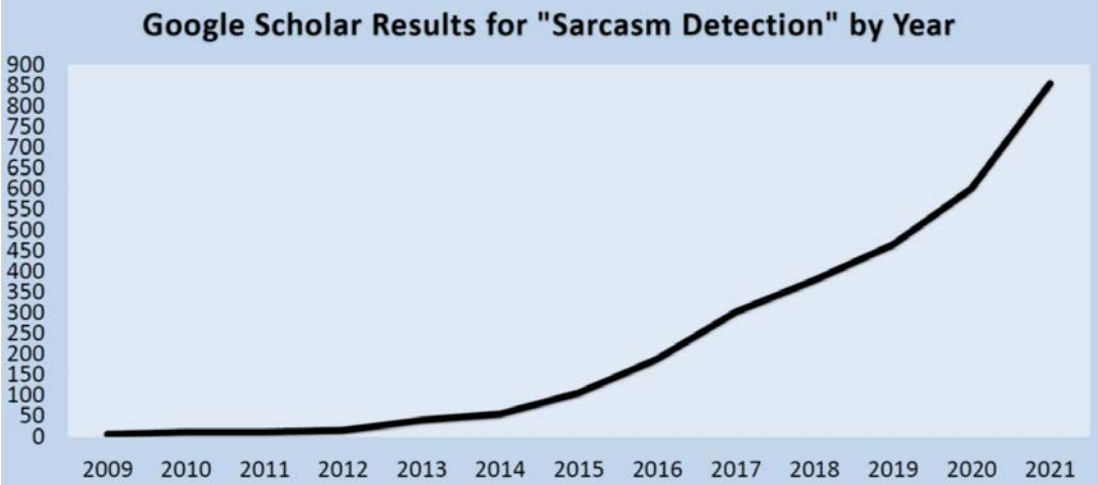

Figure 1.1 Google Scholar Search Results

(Source: Moores, B., & Mago, V.K. (2022). A Survey on Automated Sarcasm Detection on Twitter. *ArXiv, abs/2202.02516*.)

1.3 Aspects of Sarcasm

Irony and sarcasm are forms of metaphorical language that are often used to communicate the opposite of what is expressed. Sarcasm is considered as a specific form of irony, which is used when the objective of the interpretation is a person. Sarcasm is associated with the intentions of insult, mockery, ridicule, contempt, satire, insinuation and various other expressions (Filik et al., 2016). Frequently, sarcasm is utilized to criticize a person or a situation. To illustrate, consider the sentence “You’re so early!” when uttered to an individual who is late to an event it transforms into a sarcastic context. Although infrequent, sarcasm can also be used to praise. To demonstrate, consider the example “You're such a terrific tennis player!” When addressed to someone who claims to be bad at playing tennis, but contrary wins at an important competition the context becomes sarcastic.

Written forms of sarcasm are generally more difficult to identify since the usual markers such as the voice tone and facial expressions are absent in real (face-to-face) conversations (Filik et al., 2016). However, the use of emoticons, and punctuation

marks can be a contextual clue in the sarcasm identification process. These elements may provide additional context and help to interpret the intended tone of the written text to identify sarcastic features. To further explain the characteristics of sarcasm; Sarcasm is commonly expressed in three different variants (Moores and Mago, 2022):

- **A Positive Surface Sentiment With Negative Intended Sentiment:** This form of sarcasm is the most common, which is used to mock, ridicule or criticize (Joshi et al. 2017). For a better understanding, consider the given text "Monday's are the best!". The surface sentiment of this sentence is positive but the intended sentiment is negative.
- **A Negative Surface Sentiment With Positive Intended Sentiment:** In the research by (Joshi et al. 2017) these types of statements are not classified as sarcasm since it lacks the negative intended meaning. However, several other studies recognize it as a type of sarcasm and it can occasionally be used to praise. For instance, in the statement "I hate having a job, where I'm well paid to perform work I enjoy" surface sentiment is negative but the intended meaning is positive. In this instance, this type of sarcasm is used for self praise.
- **A Neutral Surface Sentiment With Negative Intended Sentiment:** This form of sarcasm has a neutral surface sentiment but the intended meaning is negative. It is generally used as a response during a dialogue. For instance consider the given text; "I am the king of the world". The sentence can be classified as sarcastic if it is given as a response, as it includes exaggeration. It has a neutral sentiment with a negative intention. This form of sarcasm is not very common and often overlooked by many researchers.

According to (Joshi et al., 2017) sarcasm can also be categorized into four types: (1) **Propositional**, where the sarcasm appears as a non-sentiment proposition but carries an implicit sentiment, (2) **Embedded**, characterized by embedded sentiment incongruity within words and phrases, (3) **Like-prefixed**, where a "like" phrase implies a denial of the presented argument, and (4) **Illocutionary**, involving non-textual cues indicating an attitude contrary to a genuine utterance. In instances of illocutionary sarcasm, prosodic variations play a role in expressing sarcasm.

Understanding written sarcasm requires a detailed analysis of various elements. Firstly, the overall context in which the message is presented must be considered. The surrounding information can offer valuable insights into whether the statement should be interpreted literally or sarcastically. Additionally, the sentiment and the tone of the

text can be an important sarcastic indicator; sarcasm often involves a distinctive and sometimes mocking or ironic tone. The choice of words, especially usage of verbs, are an important aspect, as sarcasm frequently relies on language that contradicts the literal meaning. Punctuation can also play a crucial role, with unconventional use of exclamation points, question marks, or ellipses serving as indicators. Emoticons or emojis can provide further context by signaling a sarcastic tone. All the factors mentioned above are factors that contribute to a more accurate interpretation of written sarcasm. By considering these key points, sarcastic features can be navigated through written text.

1.4 Sarcasm Detection

Sarcasm is one of the important downstream tasks in the NLP field. Sarcasm detection can engage with various topics in the NLP field such as Sentiment Analysis (SA), Opinion Mining (OM) etc. Sarcasm detection can contribute and influence these tasks to increase their performance and efficiency. With the increasing reliance on NLP technologies, sarcasm detection has gained considerable attention. The ability to detect sarcasm has practical applications in various fields. Some real-life applications could benefit from sarcasm detection include:

- **Social Media Monitoring:** Sarcasm detection can be used to analyze sentiments expressed on social media. For instance, detecting sarcasm in customer feedback can provide insights into areas that may need improvement or further attention and this may help the companies to gain insight about their product.
- **Customer Support with Automated Systems :** Sarcasm detection can be utilized to analyze the customer interactions with automatic systems and ultimately can enhance the effectiveness of these customer support systems. It allows for more accurate understanding of user queries and concerns.
- **Media Monitoring:** Sarcasm detection can be useful for analyzing news articles, blogs, and social media posts. Understanding the true sentiment of these contents can help to achieve correct information or to evaluate public opinions and biases.
- **Content Moderation:** Detecting sarcasm is essential for accurate content moderation on social media platforms. It can help in identifying potentially harmful or offensive content to prevent it from spreading in social platforms.
- **Emotion Recognition:** Sarcasm detection can be incorporated into applications to

monitor and detect emotion. There are serious mental diseases that can be identified early by analyzing people's feelings from their posts, writings etc. Using sarcasm can help in the analysis process to extract the literal meaning and this can contribute to more accurate emotion recognition.

Sarcasm detection in such applications can contribute to more effective communication, and user experiences in a variety of different domains. As technology continues to enhance and advance, using sarcasm detection is likely to be extended to a broader application area. To automatically detect sarcasm, finding a proper dataset is one of the initial steps in the process. Finding a proper dataset is crucial that determines the success of the system. There are many different resources that can be used to prepare datasets. However, creating a dataset is a very labor intensive process as the data must be extracted, organized and if necessary it must be labeled. Due to this time consuming process many researchers use the prepared datasets. Social media posts from platforms such as Twitter, Reddit etc. can be used as a dataset. Even news articles, political writings can be a possible dataset. There are various well known datasets used in sarcasm detection and researchers generally choose one or more to perform sarcasm detection tasks. More information on the dataset can be found in further chapters. After the datasets have been chosen, the data must be analyzed and preprocessed according to the specifications of the chosen task. Table 1 illustrates the primary pre-processing steps alongside their corresponding descriptions.

Table 1.1 Pre-processing stages

Pre-processing Step	**Description**
Text Cleaning	Remove irrelevant characters and convert text to lowercase.
Tokenization	Break down the text into individual words or tokens.
Stopword Removal	Eliminate common words that do not contribute significant meaning.
Stemming or Lemmatization	Reduce words to their base or root form to standardize variations.
Handling Contractions	Expand contractions to ensure linguistic consistency.
Handling Emoticons and Abbreviations	Replace emoticons or abbreviations with their full meanings.
Dealing with Spelling Errors	Correct spelling errors.
Removing URLs and User Mentions	Eliminate URLs and user mentions to focus on meaningful content.
Handling Imbalanced Data	Find and fix imbalances in data in order to prevent bias
Feature Extraction	Convert text data into numerical features for analysis.
Handling Missing Data	Address missing data to maintain dataset integrity.

Most of the pre-processing steps shown in Table 1 are applied to datasets to prepare the data for the sarcasm detection process. Following the pre-processing step, the proper approaches must be selected to perform sarcasm detection. In early years of sarcasm detection systems rule based approaches were utilized among many researchers. In rule based approaches sarcasm is attempted to be identified by specific evidences (Joshi et al., 2017). Studies published by (Veale and Hao, 2010), (Maynard and Greenwood, 2014), (Bharti et al., 2015), (Riloff et al., 2013) attempted to identify sarcasm with defining rule based approaches. These researchers specified different patterns and set baseline informations to identify sarcasm features to predict the sarcasm status from the textual data. With the advancements in technology, traditional supervised and unsupervised Machine Learning (ML) algorithms began to be used in sarcasm detection research. After the capabilities of ML models were discovered, many studies in the literature started to focus more on the ML models. Consequently, the concept of sarcasm began to gain more popularity. Researchers generally extract hand crafted features to be used in ML models. These features can be linguistic features such as word length, punctuation, pos tags etc. or sentiment features such as polarity label, polarity score etc. With the recent enhancements in ML algorithms,

Deep Learning (DL) models have been utilized. The most popular DL models are Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM). In recent years, transformer architecture models are utilized in sarcasm detection due to their learning abilities as well as their efficiency

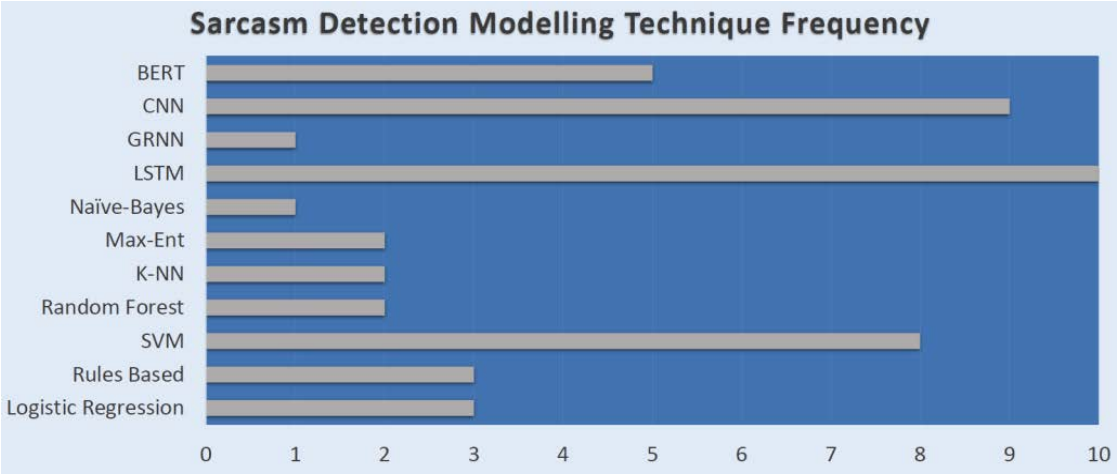

Figure 1.2 Frequency of Model Implementation in Sarcasm Detection Studies

(Source: Moores, B., & Mago, V.K. (2022). A Survey on Automated Sarcasm Detection on Twitter. *ArXiv, abs/2202.02516*.)

for finding patterns and extracting complex features. Transformer models such as Bidirectional Encoder Representations from Transformers (BERT) show promising results for detecting sarcasm. Further information on these approaches can be found in remaining chapters. In a survey study written by (Moores and Mago, 2022), over 100 sarcasm detection research have been analyzed considering their datasets, methodologies and performance. According to the chart presented in Figure 2 (Moores and Mago, 2022), ML models and DL models have been frequently used to solve the problem of sarcasm detection. Among other machine learning algorithms, the Support Vector Machine (SVM) algorithm stands out as the most commonly used in sarcasm detection. As for the DL and transformer models, LSTM, CNN and BERT are highly used among researchers. From Figure 1.1, we can observe that in recent years, ML, DL and transformer models are chosen over rule based approaches for implementing automatic sarcasm detection models.

In conclusion, sarcasm is a linguistic issue which is investigated by many researchers. Detecting sarcasm is a complex task since it is context-dependent and has a figurative nature. Various approaches, including rule based approaches, ML and DL and Transformer models have been explored to address this challenge. As technology advances new methodologies are being presented for sarcasm detection. The rest of the document as follows, Chapter 2 reports outstanding previous works on sarcasm detection. Chapter 3 provides the exploratory analysis of datasets used in the thesis, in Chapter 4 the proposed methodology is explained in detail. Chapter 5 represents the evolution of experiments. Lastly, Chapter 6 concludes the thesis.

CHAPTER 2

2. LITERATURE REVIEW

Sarcasm is generally used for expressing negative opinions. When it is used in written form the detection of sarcasm becomes nearly impossible. Written form of sarcasm is considered as a text classification problem that many researchers attempted to solve through various approaches and methodologies. In recent years sarcasm detection research gained popularity as being a challenging task in the NLP field. Many researchers were intrigued to solve the sarcasm detection problem due to its inherent complexity and the subtleties involved in linguistic expression. Sarcasm often relies on the interplay between the literal and intended meanings of words, making it a challenging task for automated systems to distinguish. Solving the sarcasm detection problem not only contributes to advancing language processing capabilities but also has several practical applications. Therefore, researchers tried to tackle the issue in detecting sarcasm on various domains using different forms of data collected from online sources. The main data source for researchers has become online resources. Online platforms such as Twitter, Reddit etc. present valuable data for sarcasm detection. Many researchers use these online platforms to gather data to be processed in their research. For this thesis study, we utilized a unique dataset named News Headlines created by (Misra, 2019). Further information on this dataset can be found in Chapter 3. Written forms of sarcasm can be detected using several different approaches. These are rule-based methods, ML techniques, and deep learning and transformer methods. Rule-based systems leverage from predefined patterns, lexical analysis, and contextual rules. Whilst, supervised ML techniques, use annotated datasets and feature engineering. Deep learning methodologies, and transformer models to identify patterns with their learning abilities. Each approach can provide

valuable solution techniques for sarcasm detection, contributing to a comprehensive understanding of this phenomenon. This section explains a variety of sarcasm detection research articles according to the datasets, methodology, and performance. This chapter mainly focuses on research studies that use the news headlines dataset to interpret meaningful comparison between this study and others in literature. Main objective of this chapter is to comprehensively analyze various research articles on sarcasm detection to analyze different methodologies used within the existing literature.

2.1 Sarcasm Detection Studies on the News Headlines Dataset

In a research article published by (Jariwala, 2020), a framework was proposed to detect sarcasm by utilizing a ML based methodology. This research was conducted on the News headlines dataset that was collected from online news sources. This dataset includes 27K news headlines where each headline is labeled as sarcastic or non-sarcastic. The ratio between sarcastic and non-sarcastic data is reasonably balanced as it contains 11.7K sarcastic headlines and 14.9K non-sarcastic. They applied pre-processing to clean and prepare the data for classification. These preprocessing steps are tokenization, stop word removal, and lemmatization. Main objective of this research is to identify optimal features that can be combined with ML models to be used in sarcasm detection. Feature extraction is a crucial step in the ML structure as it shapes the data, making it more suitable for modeling and improving the overall performance of ML algorithms. In conclusion, quality of ML models classification depends on the selected features (Jariwala, 2020). To test this theory, they extracted 17 features presented in Table 2.1.

Table 2.1 Features presented in (Jariwala, 2020)

Category	**Features**
Lexical	Noun and verb count, Bigram, Trigram, Skip gram, Unigram, Interjections, Punctuators, Exclamations, Question mark, Uppercase, Repeat words count
Sentiment	Sentiment score, Positive word frequency, Negative word frequency
Hyperbolic	Positive intensifier, Negative intensifier

They used the SVM algorithm with and without hand crafted features to observe the effect of extracted features. SVM model without features achieved 68.74% in terms of F1 score. While the same model combined with extracted features achieved 76.42% F1 score. Consequently, according to this experiment we can interpret that when the correct set of features combined with ML models, it enhances the performance of ML models. Several other research has utilized ML models to detect sarcasm. In research presented by (Nayak and Bolla, 2022), different vectorization and ML, DL models were evaluated to detect sarcastic headlines. Different word embedding models were combined with several ML and DL models to implement efficient algorithms to perform sarcasm detection. In their research, 4 different word embedding models were used to perform vectorization. These embedding techniques are TF-IDF, Word2Vec, Doc2Vec, BERT. According to their evaluation, different vectorization models highly affect the classification performance. As it is presented in their work, when DL models combined with pre-trained transformer based word embedding models, it achieves better results. In regard to various tests, using BERT word embedding with LSTM networks achieved 89.7 F1 score which is the highest performance presented in the study. In recent years, researchers began to focus more on the DL algorithms and transformer networks. Many researchers proposed different methodologies benefitted from these approaches. As shown in a study by Shrikhande et al. (2020), they proposed a CNN-based model with GloVe word embeddings and achieved an 85.6% F1 score in classifying sarcastic headlines. Another similar study (Sagarika et al., 2021), combines Bidirectional Long Short-Term Memory (Bi-LSTM) algorithm with TF-IDF word embeddings in sarcasm classification. Proposed approach achieves an accuracy of 87.79%. Another LSTM based model was proposed by (Ali et al., 2021). According to the statement presented in the study, the proposed model has three primary objectives: (1) understanding the original meaning of the text or headlines, (2) learning the nature of sarcasm, and (3) detecting sarcasm in the text or headlines. In this research, a hybrid dataset was prepared by merging the news headlines dataset with another corpus that includes 9K sarcastic sentences named Sarcasm Corpus V2. This new sarcasm corpus includes different types of sarcasm organized and separated within the corpus. According to the idea proposed in this study states, using multiple datasets can provide a more comprehensive understanding of sarcasm detection which can increase the performance of classification algorithms. This study utilizes various ML models such as SVM, Decision Tree, Random Forest (RF) etc. These ML models

failed to achieve the desired results. Therefore, a LSTM based model was designed, to correctly identify dependencies in the dataset, and is further optimized by including a GlobalMaxPool1D layer for improved feature extraction. This modification leads to an impressive 92.5% accuracy in sarcasm classification, outperforming similar studies. There can be many different reasons that might have caused this difference in performance, dataset variations can be the strongest factor in this case. This latest study indicates that adding diversity in data by including more content in the dataset can highly increase the performance. (Barhoom et al., 2022) focused on improving the dataset to directly affect the classification performance. In this study they examined 21 ML models with a custom DL based architecture to detect sarcastic news headlines. Unlike other research they did not use the raw news headline dataset instead, they increased the size of this dataset by creating two new headlines from each headline using a BERT based augmentation model. According to the result of this operation the dataset tripled in size. They tested their custom dataset with an LSTM model with Glove word embeddings. According to their evaluation, the deep learning model performed with 95.37% F1 score on the custom dataset. This experiment indicates that using a proper dataset is crucial for classification. When the dataset size is increased with proper techniques, we can observe enhancements in classification performance. Hybrid neural networks are known to be effectively used for sarcasm detection by combining different types of neural network architectures or integrating with other models. Some researchers choose to design hybrid neural network structures to correctly identify sarcasm. Hybrid neural networks can offer several advantages; by combining different neural networks, we can leverage the strengths of individual components, and this can result in enhanced overall performance and robustness. These networks are often more effective in feature extraction and capturing diverse patterns. Their use of attention mechanisms and transfer learning further adds to their adaptability and contextual understanding, which is a crucial part in identifying sarcasm. The following studies benefit from hybrid neural networks. Each study proposes unique hybrid neural networks models to identify sarcasm. In the study by (Mandal and Mahto, 2019), a hybrid CNN-LSTM network architecture which consists of an embedding layer, CNN layer and BI-LSTM layer. This architecture is trained and tested on the news headlines dataset and achieved an 86.16% accuracy score. Another study by (Mehndiratta and Soni, 2019) presented a similar architecture for sarcasm detection. Several traditional ML algorithms and word embedding models

were applied to news headlines dataset but these algorithms underperformed. Due to this reason another solution was proposed by implementing a hybrid CCN-LSTM model with Glove embeddings. This new model achieved an 86.6% accuracy score. (Misra and Arora, 2023), also build CNN-LSTM based hybrid neural network. Their architecture is a CNN based deep learning model with an additional LSTM layer and attention module. This model achieves 89.7% accuracy in classifying sarcasm. CNN-LSTM based hybrid neural network architecture is highly utilized for sarcasm detection because the sequential information encoded in the LSTM module can enhance the CNN module in the architecture. Another version of hybrid network architecture was built by (Sharma et al., 2023). The purpose of this study was to develop an efficient hybrid ensemble solution for identifying sarcastic text in social media. Hence, a hybrid ensemble model is proposed, combining training and classification from three context-based models: GloVe, Word2Vec, and BERT. BERT evaluates phrase context, distinguishing it from word-based models. The models are trained, and their classification probabilities are fed into a fuzzy layer for the final decision. Weighted information from each model is determined based on embedding vector scores, and fuzzy logic classifies sarcasm based on these weights. Using three diverse models aims to address intricacies, with fuzzy logic balancing their limitations. This study stands out from many in literature for having a higher performance. When the model is applied on news headlines dataset, it achieved 91.8% F1 score. Transformers are a type of deep learning architecture. The transformer architecture first introduced in 2017 by (Vaswani et al., 2017), has significantly influenced the research field. Transformer architectures are types of neural networks that benefit from attention mechanisms. The attention mechanism in transformers enables the model to focus on specific words or tokens in the input sequence, which becomes helpful in capturing contextual dependencies and relationships effectively to detect purposeful features. Transformer architecture has proven to be particularly beneficial for solving NLP tasks. In recent years, researchers have actively contributed to the literature by proposing innovative methodologies that leverage transformer architectures to address longstanding challenges in NLP tasks. Following research used these transformation architectures to classify sarcasm. (Scola and Segura-Bedmar, 2021) proposed a BERT base transformer model to classify sarcasm. They used the news headlines dataset to evaluate their model. According to their evaluation, the proposed model correctly classified news headlines with 90.88% F1 score. Another similar study (Jayaraman et

al., 2022) proposed a combination of several ML, DL, and transformer models to identify sarcasm in sarcastic headlines. Their ROBERTA-based model demonstrated outstanding performance, achieving the highest accuracy with a remarkable F1 score of 93.82%. This performance proves the effectiveness of their approach. Table 2.2 presents a summary of all the studies mentioned above.

Table 2.2 Summary of Sarcasm Detection Studies on the News Headlines Dataset

REFERENCE	YEAR	APPROACH	PERFORMANCE
Barhoom et al., 2022	2023	DL:CNN-LSTM	95,37% F1 score
Mandal and Mahto, 2019	2019	DL:CNN-LSTM	86.16% Accuracy
Nayak and Bolla, 2022	2022	DL:BI-LSTM	89.7% Accuracy
Misra and Arora, 2023	2019	DL:CNN-LSTM	89.7% Accuracy
Ali et al., 2023	2023	LSTM	98.39% F1 score
Shrikhande et al., 2020	2020	CNN	85.6% F1 score
Jariwala 2020	2020	ML:SVM	76.42% F1 score
Jayaraman et al., 2022	2022	Transformer: Roberta	93.82% F1 score
Scola and Segura-Bedmar 2021	2021	Transformer: BERT	90.88% F1 score
Sagarika et al., 2021	2021	DL:LSTM	87.79% Accuracy
Mehndiratta and Soni, 2019	2019	DL:CNN-LSTM	81.6% Accuracy
Sharma et al., 2023	2023	DL: Hybrid Fuzzy Logic	91.8% F1 Score

2.2 Sarcasm Detection Studies on Other Datasets

Online social platforms can provide valuable sources of data for sarcasm detection due to extensive user-generated content and the diverse linguistic expressions found in online platforms. Researchers extract data from online sources especially from platforms such as Twitter, Facebook, Reddit etc. Majority of previous studies use datasets extracted from Twitter and Reddit. In Figure 2.1. commonly used datasets are presented .

Statistic/Dataset	Headlines	Semeval	IAC	SARC
# Records	28,619	4792	3260	1,010,826
Domain	News	Twitter	Debate	Reddit
Labeling technique	Source-based	Hand labeled	Hand labeled	Tag-based
Label quality	Controlled	Controlled	Controlled	Not controlled
Language	Formal	Informal	Formal	Informal
% word2vec embeddings found	77	64	–	–

Figure 2.1 Existing Sarcasm Datasets Used in Studies (Misra and Arora, 2023)

Table 2.3 Summary of Sarcasm Detection Studies on Other Datasets

REFERENCE	YEAR	DATASET	APPROACH	PERFORMANCE
Muresan et al., 2016	2016	Reviews	SVM+Unigrams	75.7% F1 score
Kumar et al., 2020	2020	SARC dataset	MHA-BiLSTM	77.48% F1 score
Potamias et al., 2020	2020	SemEval	Recurrent CNN RoBERTA	82% Accuracy
Dadu and Pant 2020	2020	Twitter and Reddit	RoBERTa (Large)	Twitter Data: 77.1% Accuracy Reddit Data: 71.6% Accuracy
Abaskohi et al., 2022	2022	SemEval	RoBERTa	41.4% F1 score
Goyal et al., 2022	2022	iSarcasm+SARC	twitter-RoBERTa-sentiment-base	45.07% F1 score

Researchers generally use one or multiple of these datasets to perform automatic sarcasm detection. Table 2.2 outlines the sarcasm detection studies by their dataset choice, approach and performance. Many studies presented in Table 2.2 mainly utilize datasets displayed in Figure 2. Authors (Muresan et al, 2016) used an SVM algorithm with unigrams to classify sarcasm with an F1 score of 75.7%. (Kumar et al., 2020) applied a BiLSTM model on the SARC dataset and achieved an F1 score of 77.48%. The same year, (Potamias et al., 2020) implemented a RCNN-RoBERTa model and obtained an accuracy rate of 82%. In another similar study, (Abaskohi et al.,) applied a RoBERTa with mutation-based data augmentation, resulting in a F1-score of 41.4% in classifying sarcasm. (Dadu and Pant, 2020) built another RoBERTa model and achieved 77.1% accuracy in Twitter dataset and 71.6% accuracy in Reddit dataset. Similarly, (Goyal et al., 2022) used a pretrained twitter base RoBERTa model on SARC dataset and reported an F1-score of 45.07% in classifying sarcasm. In summary,

recognizing sarcasm automatically requires the use of techniques such as rule-based, ML, DL, and transformer architectures. The dataset's quality significantly impacts model performance. Therefore, finding or creating a proper dataset is crucial. Many researchers attempted to detect sarcasm on various datasets. Especially, while the use of the DL and transformer models became frequent this has significantly influenced the approaches of automatic sarcasm detection. These advanced models bring about notable improvements in the accuracy and efficiency of sarcasm detection systems. Therefore, performance of the existing sarcasm detection studies shows promising results and as technology evolves, ongoing research is essential for improving sarcasm detection models and enhancing natural language understanding.

CHAPTER 3

3. EXPLORATORY DATA ANALYSIS

In 2019, a special dataset was created by (Misra, 2019) featuring news headlines sourced from online platforms. These news headlines were collected from two online news sources “TheOnion” and “HuffPost”. “TheOnion” produces sarcastic versions of current news events. Whereas, “HuffPost” publishes original news events. Therefore, sarcastic portions of headlines are extracted from “TheOnion” platform while non-sarcastic portions are extracted from “HuffPost”. The dataset has two available versions called version1 and version2. The first version dataset includes 26,709 news headlines, featuring 11,724 sarcastic and 14,985 non-sarcastic headlines. The second version is updated from version1. In the second version, there are 28,619 headlines, consisting of 13,634 sarcastic and 14,985 non-sarcastic headlines. Specifically, version2 has been upgraded by incorporating an additional 1,910 sarcastic news headlines from the initial release to increase the diversity in sarcastic content. The dataset is given as a JSON file and it is available from a Kaggle repository. Each record in the dataset consists of three attributes:

1. **headline:** Represents the article headings
2. **is_sarcastic:** A binary flag, where 1 signifies a sarcastic instance and 0 signifies a non-sarcastic instance.
3. **article_link:** Includes a link to the original news article, potentially useful for obtaining additional information

Table 3.1 Sarcastic, non-sarcastic data example

article_link	headline	is_sarcastic
https://entertainment.theonion.com/nuclear-bomb-detonates-during-rehearsal-for-spider-man-1819572009	nuclear bomb detonates during rehearsal for 'spider-man' musical	1
https://www.huffingtonpost.com/entry/facebook-healthcare_n_5926140.html	facebook reportedly working on healthcare features and apps	0

Table 3.1 provides an illustration of example data for sarcastic and non sarcastic headlines. The dataset comprises three attributes: article_link, headline, and is_sarcastic. As indicated in the given sample, the headline “Nuclear Bomb Detonates During Rehearsal for 'Spider-Man' Musical” is labeled as 1, signifying its sarcastic context. Conversely, the headline “Facebook Reportedly Working on Healthcare Features and Apps” has a label of 0, indicating a non-sarcastic context. For additional information, a chunk of news headlines dataset is extracted and presented in Figure 3.1 and 3.2. First 5 rows in the version1 dataset are displayed in Figure 3.1. Moreover, Figure 3.2 presents samples for the version2 dataset .

	headline	is_sarcastic
0	former versace store clerk sues over secret 'black code' for minority shoppers	0
1	the 'roseanne' revival catches up to our thorny political mood, for better and worse	0
2	mom starting to fear son's web series closest thing she will have to grandchild	1
3	boehner just wants wife to listen, not come up with alternative debt-reduction ideas	1
4	j.k. rowling wishes snape happy birthday in the most magical way	0

Figure 3.1 Sample Dataset version1 (v1)

	is_sarcastic	headline
0	1	thirtysomething scientists unveil doomsday clock of hair loss
1	0	dem rep. totally nails why congress is falling short on gender, racial equality
2	0	eat your veggies: 9 deliciously different recipes
3	1	inclement weather prevents liar from getting to work

Figure 3.2 Sample Dataset version2 (v2)

Class distributions of sarcastic and non-sarcastic portions of the data from news headlines dataset is displayed in Table 3.2 and Table 3.3. In the news headline dataset, the ratio between sarcastic and non-sarcastic data is reasonably balanced as it contains 11.7K sarcastic headlines and 14.9K non-sarcastic in version1. So it includes 44% sarcastic content and 56% non-sarcastic content. Version1 can be further upgraded by adding more sarcastic content which will result in more sarcasm diversity and more balance between sarcastic and non-sarcastic data. For this reason, version2 is created by including more sarcastic headlines into version1. Consequently, version2 dataset is a more enhanced version of version 1 as it includes more sarcastic data. Which makes version 2 more balanced than version1 since this upgraded version since 48% of the dataset includes sarcastic content while 52% includes non-sarcastic comments. Still the density of non-sarcastic contents in both dataset is higher compared to sarcastic contents but this difference is lower in version2 dataset.

Table 3.2 Class distributions in v1 dataset

Class	Count	Percentage(%)
Sarcastic	11,724	44
Non-sarcastic	14,985	56

Table 3.3 Class distributions in v2 dataset

Class	Count	Percentage(%)
Sarcastic	13,634	48
Non-sarcastic	14,985	52

Applying proper pre-processing is crucial in order to maximize efficiency and effectiveness of the dataset. Preprocessing allows us to prepare the dataset by removing any unnecessary information. Preprocessing plays a crucial role in handling and reducing noise in data. Noise in data refers to irrelevant or information that can impact the efficiency of the dataset. In order to address the noise or any issues in the dataset, a comprehensive analysis is necessary to further enhance the understanding. For this reason, using visualization techniques such as bar charts, pie charts, and line plots, word clouds help us gain valuable information about the dataset. In relation to data visualization, to identify patterns in the data time series analysis, clustering methods can be applied. Consequently, to further enhance the understanding of the news headline datasets visualization techniques were used. To identify patterns, we extracted the distributions of character length and word length density information considering all the headlines in the dataset. These graphical informations are presented in Figure 3.3, 3.4, 3.5. In Figure 3.3 Character length distribution plot is presented. In Figure 3.3, x axis shows the character length in headlines while y axis shows headline count information. Figure 3.4 shows word length density in headlines where x axis has the word count information and y axis has density information of headlines. Figure 3.5 presents the frequency of the length of each headline in a wider range. These graphical contents presented in Figure 3.3, 3.4 and 3.5 can be useful in detecting the characteristics of outliers present in the dataset.

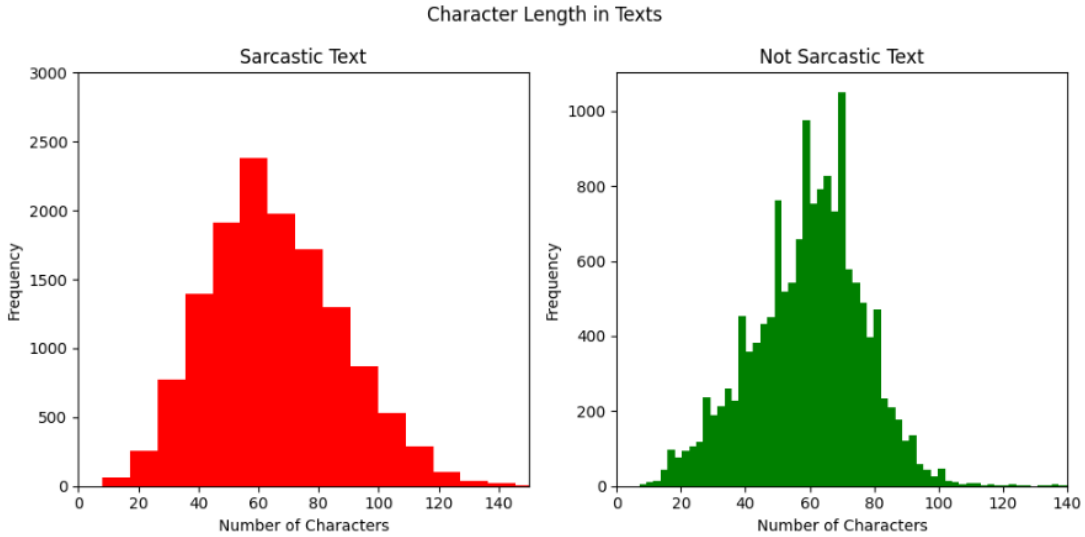

Figure 3.3 Character length frequency distributions in headlines

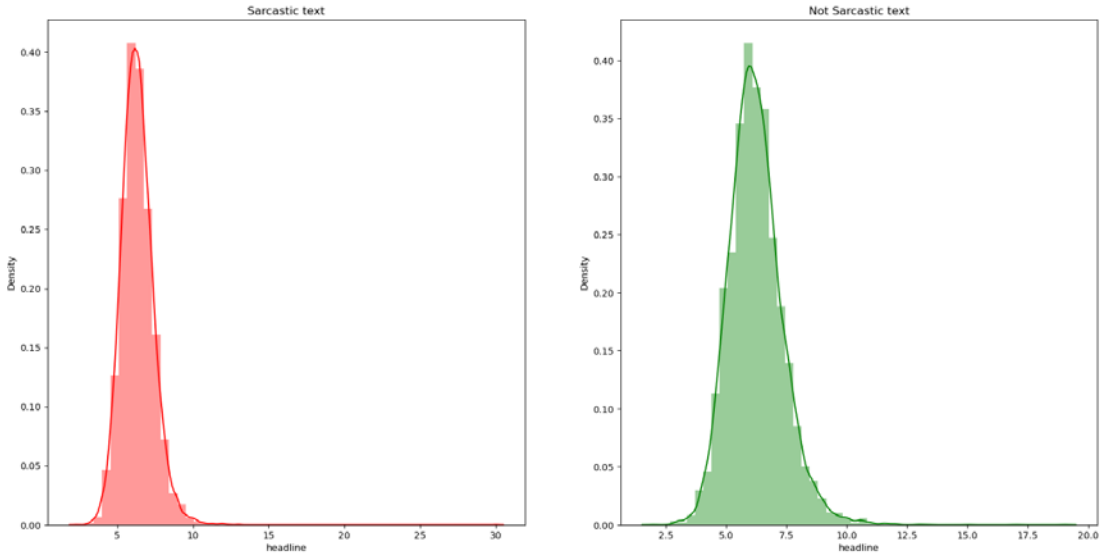

Figure 3.4 Word length density in headlines

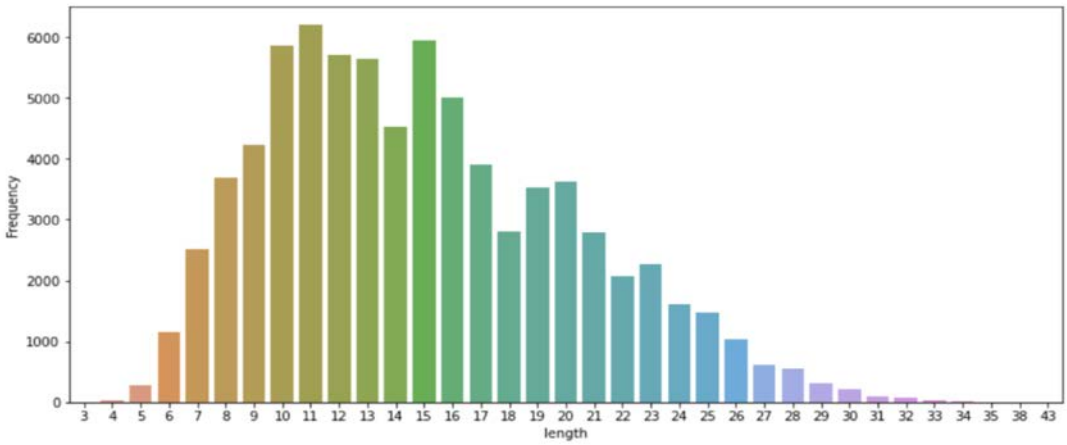

Figure 3.5 Frequency of the length of each headline in the dataset

Word cloud also is a visualization technique commonly used for representing the frequency or importance of words within the dataset. Word cloud is a graphical representation that visually highlights the most frequently occurring words in a given dataset. Word clouds efficiently summarize textual information by visually displaying key terms in different font sizes, highlighting their frequency within the given text. Word clouds provide insightful representations of specific dataset. We applied word cloud technique on news headlines dataset to further understand the dataset. Word cloud technique is applied separately on sarcastic and non-sarcastic contents in news headline dataset. From Figure 3.6, the first image presents the word cloud image for non-sarcastic contents in the dataset while the second image presents the word cloud

image for sarcastic contents. According to Figure 3.6, the words “man”, “new”, “nation”, “report”, “number”, “local”, “friend” stand out in the non-sarcastic part of the dataset. While, “trump”, “new”, “people”, “life”, “star”, “love”, “women”, “say” words are more prominent in the sarcastic part of the dataset. Especially, “trump” keyword is one of the most frequent words used in sarcastic headlines. This indicates that sarcasm generally applied to political content to mock a political situation or a political figure.

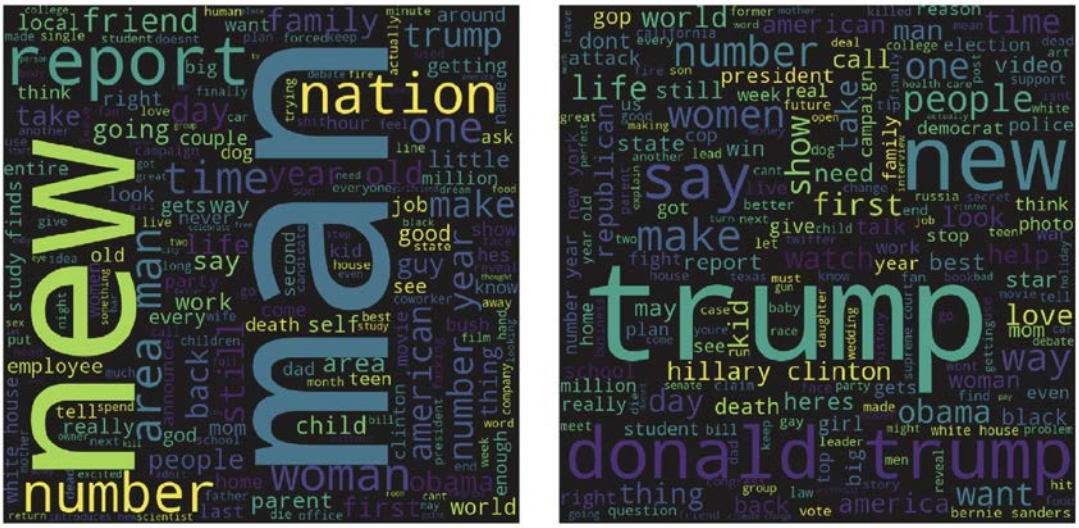

Figure 3.4 Word Cloud Representation of Non-sarcastic and Sarcastic Headlines

In summary, there are vast amounts of data that can be found on social platforms. The majority of sarcasm detection solutions rely on datasets retrieved from social media platforms; however, these datasets are mostly noisy in terms of both labels and language. In contrast, our news headline dataset, comprising news headlines sourced from the news platforms, offers distinct advantages compared to ones extracted from social media. Firstly, news headlines, written by journalists, with highly formal style, which ensures minimal spelling errors. This characteristic not only simplifies the task by reducing lexical inconsistencies. Moreover, given that TheOnion platform's primary objective is to publish sarcastic news, as a result we can acquire high-quality labels with different domains and significantly less noise in comparison to social media datasets. Unlike tweets from Twitter or posts from Reddit platform, which often constitute responses to other tweets, each extracted data is independent in the news

headlines dataset. News headline datasets overcome the limitations mentioned above and offer valuable data to be used in the automatic sarcasm detection task.

CHAPTER 4

4. PROPOSED METHODOLOGY

4.1 Dataset Preprocessing

In data preparation, preprocessing is one of the crucial steps in order to extract meaningful information. Key objectives of preprocessing include processes such as data cleaning to remove any irrelevant information or correcting any spelling and grammatical mistakes. Tokenization to break down each text into tokens. Normalization by converting data formats or numerical scales to ensure the consistency in data. Handling missing data by addressing and removing any null values in the dataset. Text vectorization for converting text into numerical values to be processed by the computer. All these key processes can be applied using preprocessing steps shown in Table 4.1. To apply these processes, generally used methodologies and some useful libraries are presented in Table 4.1.

Table 4.1 Most used methodologies and useful libraries for pre processing stages

Preprocessing Stage	Methods/ Useful Libraries
Tokenization	NLTK or spaCy libraries
Stemming or Lemmatization	Porter or Snowball stemmers, spaCy lemmatization
Removal of Stopwords	NLTK or spaCy libraries
Text Vectorization	TF-IDF, Word Embeddings (Word2Vec, GloVe), Bag of Words (BoW), Transformer models (e.g., BERT, GPT, DistilBERT), FastText, Doc2Vec
Normalization	Min-Max scaling, Z-score normalization

Table 4.2 Pre processing applied to news headlines datasets

Pre Processing Stage	Dataset	
	v1	v2
Text Cleaning	✓	✓
Tokenization	✓	✓
Stopword Removal	⊗	⊗
Stemming or Lemmatization	⊗	⊗
Handling Contractions	✓	✓
Handling Emoticons, Abbreviations	⊗	⊗
Dealing with Spelling Errors	⊗	⊗
Removing URLs and User Mentions	✓	✓
Handling Imbalanced Data	✓	✓
Feature Extraction	✓	✓
Handling Missing Data	⊗	⊗

Table 4.2 presents the preprocessing stages applied to news headlines datasets to prepare the data to be processed. First, we applied text cleaning by removing unnecessary characters and punctuations. Then all the words in each heading is converted to lowercase to ensure the consistency in data. Afterwards, the data is further standardized by handling contradictions. Contractions are shortened forms of words or phrases, often created by combining two words and using an apostrophe, such as "don't" (do not) or "can't" (cannot). To handle contradictions, the word is expanded to their full forms. We also removed the article_link column from the dataset since we only need the headline information and sarcastic labels. Later we detected headlines that are short or long compared to the overall size of headlines in the dataset. According to Figure 3.3 which shows character length distributions of headlines, we decided to delete headlines that have less than 45 characters and also deleted headlines that include more than 180 characters. This process ensures the optimization of the dataset by removing the least frequent components of the dataset. This also contributes to the performance of the learning models by providing reliable patterns and uniformity throughout the dataset. After this step, we tokenized and applied vectorization to transform the data into the numerical form so that it can be processed by the learning models.

News headline dataset is composed of headlines extracted from formal news sources. Therefore, it includes content that is written by professionals. Thence, it contains minimal grammatical or spelling errors. For this reason, we did not apply any process to detect and clean any spelling or grammatical errors. We also did not attempt to remove any emoticons as it does not contain any emoticons due to its formal context. The dataset does not contain any null values so consequently we did not handle any missing values in the dataset. Lastly, we did not remove stop words nor apply any stemming or lemmatization. The reason for this is, when the stemming or lemmatization process is applied to a dataset it can highly change the context. Since transforming the data to this degree can affect the sarcastic properties, we decided not to apply any form of stemming.

4.2 Feature Engineering

4.2.1 Data Augmentation

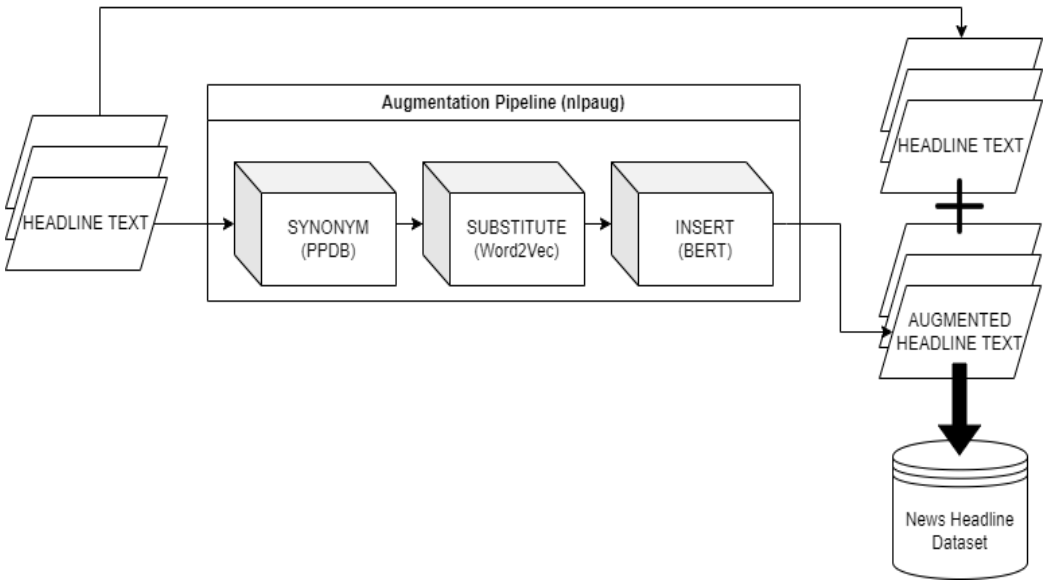

Figure 4.1 Proposed Sarcasm Detection Classifier Framework

The augmentation pipeline shown as in Figure 4.1, is a process used to enrich the dataset of news headlines that is used to train the sarcasm detection model. It does this by taking the original headlines and transforming them in a variety of ways to create new synthetic headlines. All applied steps in the augmentation pipeline are explained below:

4.2.1.1 PPDB: The Paraphrase Database for Assigning Synonyms

A database called PPDB, which is automatically extracted, has millions of paraphrases in sixteen different languages. Enhancing language processing through increased system resilience to linguistic variability and unknown terms is the aim of PPBD. Under the terms of the Creative Commons Attribution 3.0 United States License, the whole PPDB material is publicly accessible. ACL short paper (Pavlick et al., 2015) describes a supervised regression model that is used to rank the paraphrases in PPDB. The database is split into six sizes based on this score, ranging from S to XXXL. For maximum recall, XXXL contains all pairs, whereas S exclusively contains the highest-scoring pairs for maximum precision. With every size increase, the quantity of paraphrases doubles, and larger sizes absorb smaller sizes. Three categories of paraphrases can be found in PPDB: syntactic (paraphrase rules with non-terminal symbols), phrasal (multiword to single/multiword), and lexical (single word to single word). We have used the All-S module which has the smallest size and consists of all categories of paraphrases.

4.2.1.2 Word2Vec Method for Substitution

In the augmentation pipeline for our sarcasm detection system, the second module is word2vec, we implemented a word embedding-based technique to augment the dataset by substituting one word in each text with its Word2Vec vector equivalent. Word embeddings are a powerful tool in natural language processing, allowing for the creation of dense vector representations of words that capture semantic meaning and context. By utilizing Word2Vec, we were able to substitute words with vectors that retain the original word's meaning, thereby creating new variations of the text that preserve the underlying sentiment while adding diversity to the dataset. This approach has the potential to improve the robustness and generalization ability of our deep neural network model, leading to more accurate and reliable sarcasm detection.

4.2.1.3 BERT Method for Insertion

BERT, an acronym for Bidirectional Encoder Representations from Transformers, is a highly potent method of pre-training in natural language processing (NLP) that has demonstrated efficacy across a range of NLP tasks. One potential application of BERT lies in the augmentation pipeline, wherein a new word is inserted into sentences within a news headline dataset. This process entails identifying the semantic relationships among the words within the sentence and subsequently determining the optimal placement for the new word based on these relationships. For example, if the newly introduced word functions as a synonym for an existing word within the sentence, it may be advisable to insert the new word in the same position as the original. Conversely, if the new word pertains to the overarching theme of the sentence, it may be more appropriate to insert it at either the beginning or end of the sentence. By leveraging BERT to comprehend the semantic relationships between words, it becomes feasible to automatically insert new words into sentences in a manner that is both meaningful and contextually fitting.

Table 4.3 Textual Data Augmentation Example

	Sentence
Original	The quick brown fox jumps over the lazy dog
Synonym (PPDB)	The quick brown fox climbs over the lazy dog
Word Embeddings (word2vec)	The easy brown fox jumps over the lazy dog
Contextual Word Embeddings (BERT)	Little quick brown fox jumps over the lazy dog
PPDB + word2vec + BERT	Little easy brown fox climbs over the lazy dog

As shown in the Table 4.3, we have realized augmentation pipeline in the order of PPDB for changing the random one word with it's synonym, Word2Vec for swapping one random word with its substitution and BERT for inserting a new word which is semantically related with the help of contextual relationship understanding of transformer models. PPDB and Word2Vec processes choose a random word among non-stopwords members. Combination of these three augmentation techniques result in the best synthetic data for our dataset. In Table 4.3, the sentence given as example is "The quick brown fox jumps over the lazy dog". When we apply PPDB, the word "jumps" is changed into "climbs" which are synonyms of each other. Then, it is fed

into word2vec, the word “quick” is swapped with its substitution word “easy”. Lastly, the sentence is preceded by BERT for inserting a new context related word. In this example, BERT inserted the word “Little” at the beginning of the sentence. Our synthetic sentence would be augmented as “Little easy brown fox climbs over the lazy dog”. Once the augmentation pipeline has been applied, the augmented headlines are then added to the original dataset.

In a nutshell, the first step was to go over each headline and create a synthetic news headline from it by using the function aug_bert.augment. Thus, we have two times the number of headlines (i.e. we have 85,854 headline news). Then we saved it as a csv file for further usage and next steps. This creates a much larger dataset that the sarcasm detection model can be trained on. This can help the model to learn to better generalize to unseen data, and so improve its accuracy.

4.2.2 Hand Crafted Features

Handcrafted features, in other words engineered features that are manually created variables or characteristics extracted from the data. These features are crafted and can be highly used for capturing specific patterns, relationships, or information that may be relevant to a particular issue. In some cases, existing features in the dataset may not be sufficient for solving a specific issue. Therefore, using existing features with hand crafted properties can provide additional information and patterns that can be used in problem solutions. Automatic sarcasm detection is commonly solved by utilizing ML, DL and transformer models. Learning models can automatically extract features and learn from the data but including hand crafted features can further enhance the interpretability and optimize the learning process of these models.

4.2.2.1 Polarity Labels and Scores

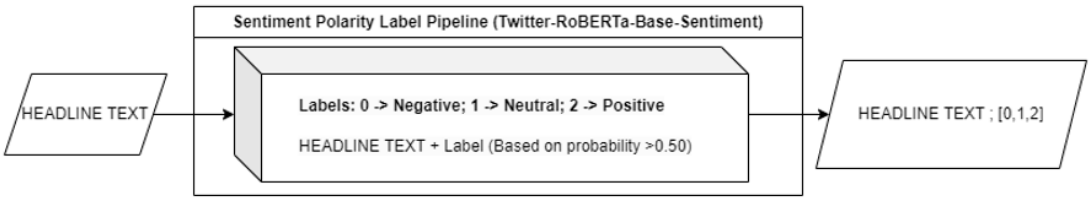

Figure 4.2 Sentiment Polarity Label Pipeline

Polarity label is one of the handcrafted features that we added for each news headline in the dataset. The model that we have used for this task is called Twitter-RoBERTa-Base-Sentiment. This RoBERTa-base model as shown in Figure 4.2 has been trained on approximately 124 million tweets spanning from January 2018 to December 2021. Furthermore, it has been meticulously fine-tuned for the purpose of sentiment analysis, specifically focusing on the TweetEval benchmark. The labels of the model are 0 for negative, 1 for neural and 2 for positive. Along with the label result, we can also get the percentage rate for that label.

4.2.2.2 Category Labels and Scores

Figure 4.3 News Categorization Label Pipeline

Our other handcrafted feature is category labels. The model that we have used for this task is called News-Category-Classification-DistilBERT. The DistilBERT model as shown in Figure 4.3 was trained on a corpus of 210,000 news headlines spanning from 2012 to 2022, which were sourced from HuffPost. There are a total of 42 news categories in the dataset and the top-15 categories are given also in Table 4.4.

The model assigns a label and the score of a label as separate columns to the news headline dataset based on the highest probability score among all categories. Example dataset is shown in Figure 4.4.

Table 4.4 Top 15 Category Labels

Category Labels	
POLITICS	QUEER VOICES
WELLNESS	FOOD & DRINK
ENTERTAINMENT	BUSINESS
TRAVEL	COMEDY
STYLE & BEAUTY	SPORTS
PARENTING	BLACK VOICES
HEALTHY LIVING	HOME & LIVING

headline	is_sarcastic	polarity	pol_score	category	cat_score
thirtysomething scientists unveil doomsday clock of hair loss	1	negative	0.49727	COMEDY	0.812403
dem rep. totally nails why congress is falling short on gender, racial equality	0	negative	0.82847	POLITICS	0.9999136
eat your veggies: 9 deliciously different recipes	0	positive	0.77599	WELLNESS	0.834558
inclement weather prevents liar from getting to work	1	negative	0.77285	WEIRD NEWS	0.5884792
mother comes pretty close to using word 'streaming' correctly	1	neutral	0.73079	PARENTS	0.9918844

Figure 4.4 Dataset v2 with handcrafted features

Category scores in addition to the labels is beneficial, as it allows the model to take into account the strength or intensity of the sentiment or category. Polarity scores reflect the overall sentiment of the text, whether it's positive, negative, or neutral. Category scores, on the other hand, reflect the degree to which the text belongs to a particular category, such as politics, sports, or entertainment.

Using both types of scores provide a more nuanced view of the text and help the model make more accurate predictions. For example, a text with a high polarity score for negativity and a high category score for politics may indicate that the text is sarcastic, as it expresses negative sentiment about a political topic.

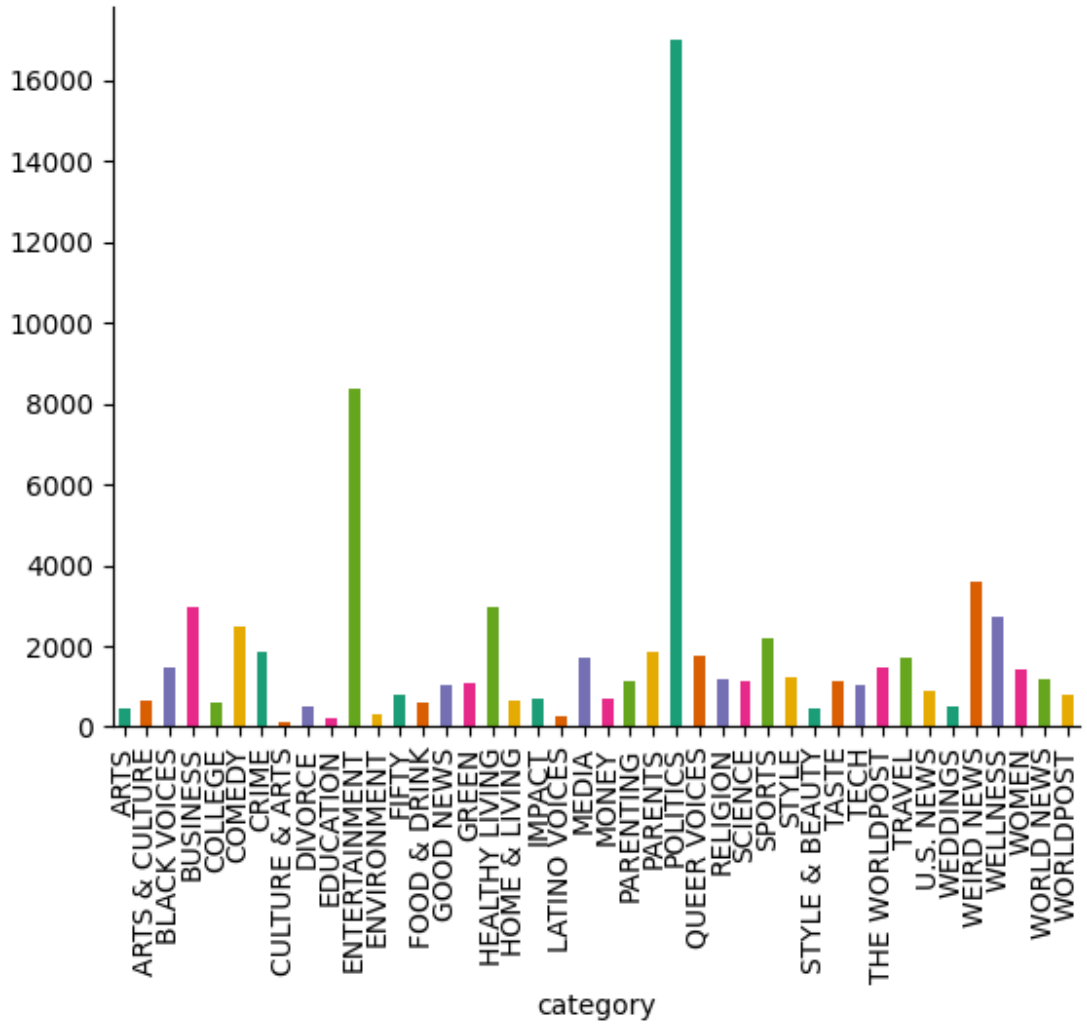

Figure 4.5 Category distribution of the news headlines

In Figure 4.5, it is revealed that the "Politics" and "Entertainment" categories hold the highest concentrations of sarcastic headlines. This finding aligns with the common observation that sarcasm is frequently employed in these domains, often to convey humor, critique, or social commentary.

Several factors might contribute to this prevalence of sarcasm in politics and entertainment news. In the realm of politics, sarcasm can serve as a potent tool for satire and ridicule, enabling individuals to mock opposing viewpoints or expose perceived hypocrisy. It can also function as a means to express dissent or criticism in a veiled manner. Within the entertainment sphere, sarcasm is often harnessed for comedic effect, infusing humor into news stories and headlines. By employing sardonic language, writers and editors can pique readers' interest and create amusement.

It's worth noting that the prominence of sarcasm in these categories doesn't necessarily imply that sarcasm is absent from other domains represented in the dataset. The visualization merely highlights the relative distribution of sarcastic headlines across various categories.

4.3 Framework of Proposed Methodology

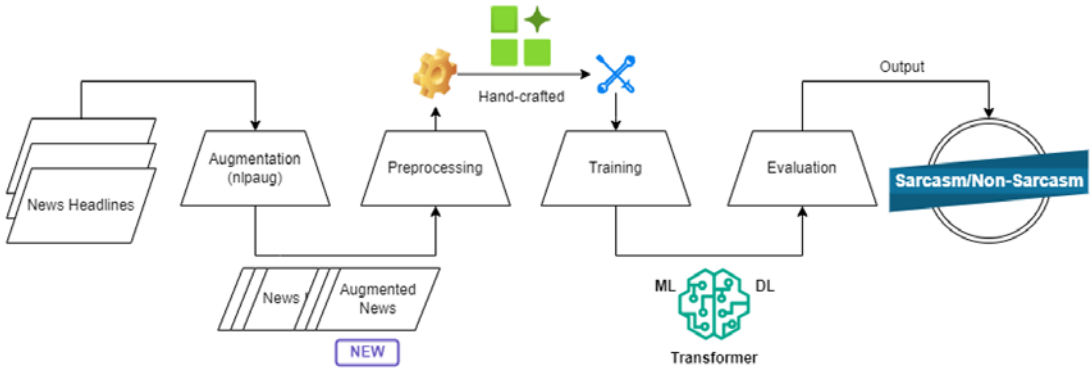

Figure 4.6 Framework of the Proposed Sarcasm Detector

The framework of the proposed sarcasm detector is given in Figure 4.6. This pipeline begins with news headlines as input. The news headlines dataset is then augmented using a library called “nlpaug”. This augmentation process involves 3 sub-processes which are swapping a word with it’s synonym using PPDB corpus, changing a word with its substitute by using word2vec and lastly inserting a new context related word by using BERT model to expand the dataset and potentially improve the

classifier's robustness. Next, both the original and augmented headlines undergo preprocessing to prepare them for training. This likely includes tasks like tokenization (breaking down text into smaller units), cleaning (removing noise or irrelevant information), and normalization (standardizing text formatting including contractions). The preprocessed headlines have been used for extracting hand-crafted features before they are fed into a model for training. Polarity labels and scores, category labels and scores of the news headlines are assigned by the help of transformer models as they are shown in Figure 4.2 and 4.3. We have used three distinguished techniques for the training and the comparisons of the sarcasm detector models. Traditional ML models such as SVM, Decision Tree and Random Forest have employed and tested on test data. On the other hand, deep learning models such as CNN, LSTM, BiLSTM and RNN are employed with a variety of diverse embedding models such as Fasttext and GloVe. Lastly, models based on transformers have been employed such as BERT, RoBERTa and DistilBERT. Once training is complete, the model's performance is evaluated using a set of test headlines and compared depending on the F1 score metric. This evaluation determines how accurately the model can distinguish between sarcastic and non-sarcastic headlines. Finally, the trained model is able to take new, unseen news headlines as input and classify them as either sarcastic or non-sarcastic.

4.4 Word Embedding

Word embedding is an NLP technique for representing words as numerical vectors in a high-dimensional space (100, 200, 300, ...). It allows us to compare words based on their semantic meaning, e.g. words with similar meanings will have similar vectors. The most popular word embeddings are used, Word2Vec, GloVe, and FastText for training the proposed models and representations of the words for the augmentation process.

4.4.1 Word2Vec

The Word2Vec (Mikolov et al., 2013) is a neural network with two layers that is intended to represent word linguistic contexts. It helps to comprehend how similar the terms are to one another. Utilizing extensive text corpuses such as our datasets, it produces a vector space with hundreds of dimensions (ranging from 100 to 300). For

the Word2Vec pre-trained model, 300 dimensions are the standard dimension. The skip-gram and CBOW models are two of the models used in the computation of the Word2Vec algorithms. In a text sequence, the skip-gram model creates a word based on the words surrounding it. While the center word produced by the CBOW model is determined by the context words that surround it. The cost function of Word2Vec can be calculated as in (4.1).

$$J(\theta) = \sum_{c} \sum_{o} log\ log\ P\ (w_c) \tag{4.1}$$

where:

$J(\theta)$ is the cost function, which measures how well the model predicts the context words given the center word,

θ are the parameters of the model, which are the weights of the neural network,

c is the index of the center word,

o is the index of the context word,

$P(w_c)$ is the probability of the context word w_o given the center word w_c. This is calculated using the softmax function.

4.4.2 GloVe: Global Vectors for Word Representation

Unlike Word2vec (Mikolov et al., 2013), which uses global statistics (word co-occurrence) coupled with local statistics (local context information of words) to produce word representations, GloVe (Pennington et al., 2014), designed by Standford, is one of the well-known word embedding methods. The identification of significant semantic links between words is made possible by GloVe. GloVe embeddings are used first in the experiments. Equation (4.2) shows the cost function for GloVe word embeddings.

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \widetilde{w_j} + b_i + \widetilde{b_j} - logX_{ij}\right)^2 \tag{4.2}$$

where:

X_{ij} represents how often word i appears in the context of word j,

w_i is the vector of main word,

$\widetilde{w_j}$ is the vector of the context word,

$b_i, \widetilde{b_j}$ are the main and context words' scalar biases, and

f is the weighting function that helps us to prevent learning only from prevalent word pairs.

4.4.3 FastText

An open-source library called FastText (Bojanowski et al., 2017) is useful for quickly learning word representations. It contains two-word representation computation models: continuous-bag-of-words (cbow) and skip-gram. Whereas the Cbow model predicts the target word based on its neighbors, the skip-gram model learns to estimate a target word's neighbors. It constructs vectors for unknown words using subword-level information. Cosine similarity between the vectors is employed. Equation (4.3) shows how cosine similarity is calculated as the dot product of the vectors normalized by their size.

$$cosine\_similarity(u, v) = \frac{u \cdot v}{||u|| \cdot ||v||} \tag{4.3}$$

4.5 Activation Functions

Activation functions play a crucial role in neural networks as they serve to introduce nonlinearity into the network. This nonlinearity is essential in allowing the network to capture and understand more intricate relationships between input and output variables.

A plethora of activation functions are available, each offering its own set of advantages and disadvantages. Consequently, the choice of which activation function to employ must be dictated by the specific task at hand, in order to maximize the potential for improved outcomes. In our models, we have implemented the most commonly used and highly beneficial activation functions, including relu, tanh, sigmoid, and softmax. Further elaboration on the characteristics and properties of each

of these functions can be found in the subsequent subsections.

4.5.1 ReLU

The Rectified Linear Units (ReLU) (Agarap, 2018) demonstrate that the output is the same as the input value if it is positive. Conversely, if the input value is negative, the result is 0, which is illustrated in Figure 4.7. This activation function is widely favored due to its computational efficiency and its ability to mitigate the vanishing gradient problem. The computation is described in equation (4.4), and it has been employed in conjunction with the soft plus function, a smoothed variant of ReLU. The calculation for the soft plus function is presented in equation (4.5).

$$f(x) = max(0, x) \tag{4.4}$$

$$f(x) = log(1 + e^{x}) \tag{4.5}$$

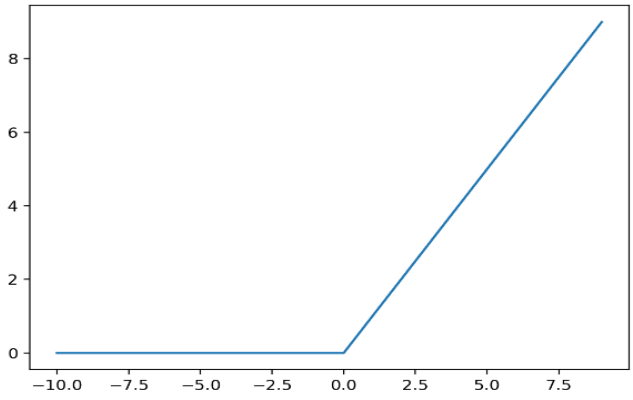

Figure 4.7 ReLU graph

4.5.2 Hyperbolic Tangent (Tanh)

The Tanh function (Namin et al. in 2009), bears resemblance to the sigmoid function; however, it possesses a range of values between -1 and 1. This particular characteristic renders it more appropriate for regression scenarios, as it has the capacity to represent both positive and negative values. The calculation of the Tanh function is illustrated in equation (4.6), while its graphical representation is depicted in Figure 4.8.

$$f(x) = (e^x - e^{-x})/(e^x + e^{-x}) \quad (4.6)$$

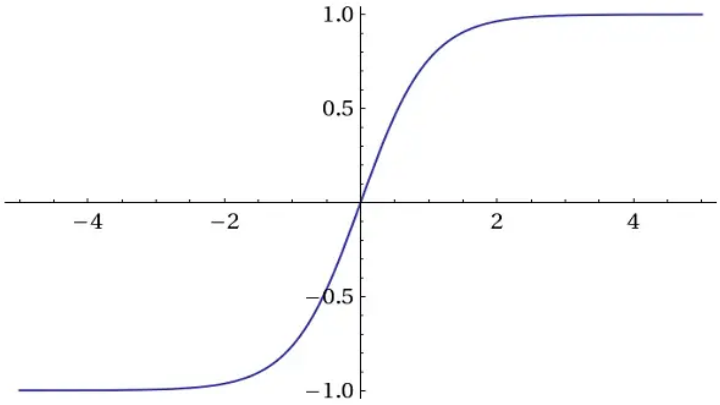

Figure 4.8 Tanh graph

4.5.3 Sigmoid

The sigmoid function (Narayan in 1997), is a function that exhibits an S-shaped curve and possesses a domain of [0, 1]. It is frequently employed in the realm of classification problems, as it can be interpreted as the probability of a specific class. In our investigation, we utilized this activation function in the context of binary classification for the identification of sarcasm in datasets. The mathematical expression and graphical representation of this function are illustrated in equation (4.7) and Figure 4.9, respectively.

$$f(x) = 1/(1 + e^{-x}) \quad (4.7)$$

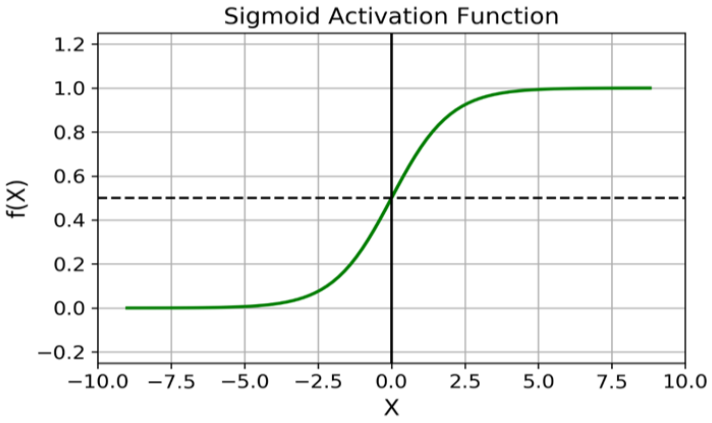

Figure 4.9 Sigmoid graph

4.5.4 Loss Functions

A loss function assesses the accuracy of a model in predicting true labels and plays a crucial role in shaping the model during training by minimizing this loss. Table 4.5 highlights the fundamental distinctions between various loss functions.

Table 4.5 Differences of Binary cross entropy and Sparse categorical cross entropy

Feature	Binary cross entropy
Number of classes	2
Representation of ground truth labels	Single value (0 or 1)
Use cases	Binary classification problems

Three distinct classification models have been instantiated, wherein two are dedicated to binary classification tasks, while the third is designed for multi-class classification. Throughout the training phase, the "binary-crossentropy" loss function, delineated in equation (4.9), was utilized for the binary classification models. The ascribed labels for the Sarcasm dataset are dichotomously assigned as either “sarcastic” or “non-sarcastic.”

$$BCE(y, \hat{y}) = -\sum_{i=1}^{N} \quad y_i \; log \; log \; (\hat{y_i}) \; + (1 - y_i) \; log \; log \; (1 - \hat{y_i}) \tag{4.9}$$

where:

BCE is Binary Crossentropy,

N is the output size,

y is the true label value and

$\hat{y}$ is the predicted label value.

4.6 Callback Functions

Callbacks are instrumental tools in machine learning designed to mitigate overfitting, and they are readily available in the Keras library. Within our models, we have incorporated three key callback methods: Early Stopping, Learning Rate Reduction, and Model Check pointing. A detailed elucidation of these callback functions is provided in the following sections.

4.6.1 Early Stopping

Early stopping ceases, the training process once, a specific number of attempts have been made, in the event that there is no further progress in validation accuracy. Consequently, this approach ensures that training time is utilized efficiently. The method itself relies on three key parameters: monitor, patience, and mode. The monitor parameter is responsible for specifying the metric that needs to be examined for signs of overfitting. Patience, on the other hand, sets the limit for the number of epochs during which the given metric is allowed to remain stagnant without improvement. Lastly, the mode parameter determines whether the metric should be monitored with regards to its increase or decrease. In our particular models, the metric that was monitored was validation accuracy, while the patience parameter ranged between 3 and 8, and the mode was set to max.

4.6.2 Reduce Learning Rate

The callback function known as "Reduce Learning Rate" is also applicable within the Keras library. By reducing the learning rate, the accuracy of a model can be enhanced as it enables the model to adapt the learning rate throughout the training process. This function consists of three parameters: monitor, patience, and factor. The usage of monitor and patience is similar to what we have previously explained for early stopping. The factor parameter determines the rate at which the learning rate is reduced and its default value is 0.1.

4.6.3 Model Checkpoint

The callback function of saving the model weights after each epoch is accomplished by the model checkpoint. This is done by closely monitoring the performance of the model to save the best model in case of any interruptions or malfunctions. Consequently, this enables us to have pre-trained models that can be utilized for future purposes. Moreover, it allows for the application of fine-tuning. The callback function can be configured with various parameters including the file path for specifying the location where the model is saved, the monitor for selecting the optimal weights, and “save_best_only” to determine whether to save all model weights or only the best ones.

4.7 Classification Models

4.7.1 SVM

Support Vector Machines (SVMs) are a popular and effective classification model in machine learning. SVMs are based on the concept of finding the optimal hyperplane that can best separate data points of different classes as it is shown in Figure 4.10. In other words, SVMs aim to maximize the margin or distance between the hyperplane and the nearest data points, also known as support vectors. This results in a more robust and generalized model, making it suitable for classification tasks with high dimensional data and small sample sizes. SVMs can also handle non-linearly

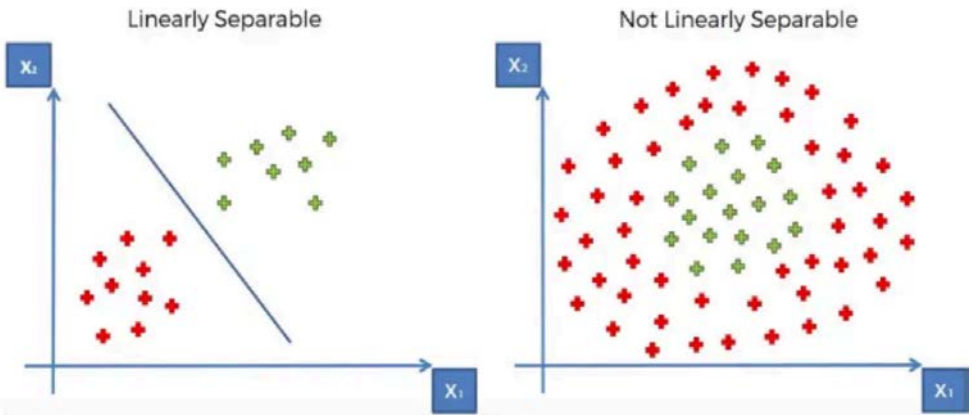

Figure 4.10 SVM Linear Separable and Not Linearly Separable Example

(Source: https://www.mdpi.com/2227-7390/11/14/3251)

separable data by mapping the data into a higher dimensional space using a kernel function, such as the radial basis function (RBF), polynomial or linear kernel. We have used linear kernel for this task. The choice of kernel function and corresponding parameters can significantly impact the performance of SVMs, and therefore, careful tuning is often required. Despite their simplicity, SVMs have been shown to perform well in various applications, including text classification and sentiment analysis.

4.7.2 Decision Tree

Decision Trees are a widely used and interpretable classification model in machine learning. A Decision Tree is a hierarchical structure that recursively partitions the data into subsets based on feature values, resulting in a tree-like model with decision nodes and leaf nodes. The decision nodes represent feature tests, while the leaf nodes represent the class labels. During training, the Decision Tree selects the feature and split value that maximizes a certain criterion, such as information gain or Gini impurity, at each decision node. Decision Trees are simple to understand and visualize, making them useful for explaining the decision-making process to non-technical audiences. However, Decision Trees can be prone to overfitting, especially when the tree is deep and has many branches. To address this issue, various pruning techniques, such as reduced error pruning or cost complexity pruning, can be applied to simplify the tree and improve the generalization performance. Decision Trees have been successfully applied to various applications, including fraud detection, medical diagnosis, and customer segmentation.

4.7.3 Random Forest

Random Forests are an ensemble learning method that combines multiple Decision Trees to improve the classification performance. The basic idea behind Random Forests is to build a set of Decision Trees, where each tree is trained on a random subset of the training data and features. During prediction, the Random Forest aggregates the predictions of all the trees, for example, by taking the majority vote or averaging the probabilities, to produce the final classification. Random Forests reduce the variance and overfitting of a single Decision Tree by introducing randomness and diversity among the trees. The key parameters of Random Forests include the number of trees, the maximum depth of each tree, and the number of features to consider at

each split. Careful tuning of these parameters is essential to achieve the optimal performance of Random Forests. Random Forests have been shown to be highly effective in various applications, including text classification, image recognition, and recommender systems.

In the field of NLP, Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM) networks, and Bidirectional Encoder Representations from Transformers (BERT) are popular deep learning models that have been widely used for various NLP tasks, including sarcasm detection. The following section provides an elucidation of each of these models and their respective importance within the framework of sarcasm detection.

4.7.4 Convolutional Neural Network (CNN)

A CNN is a type of deep learning model that is commonly used for image processing tasks. However, CNNs have also been successfully applied to NLP tasks such as sentiment analysis and sarcasm detection. In a CNN, the input data is convolved with a set of filters, also known as kernels, to extract local features. These filters are applied to different regions of the input data, and the resulting feature maps are then passed through non-linear activation functions to introduce non-linearity into the model. The output of the convolutional layer is then pooled to reduce the spatial dimensions of the feature maps. The pooled feature maps are then fed into one or more fully connected layers to perform the final classification task.

4.7.5 Bidirectional Long Short-Term Memory (BiLSTM)

A BiLSTM network is a type of recurrent neural network (RNN) that is capable of learning long-term dependencies in sequential data. An LSTM network is a type of RNN that is designed to address the vanishing gradient problem that affects traditional RNNs. An LSTM network consists of a set of memory cells that maintain a hidden state over time, and three gates that control the flow of information into and out of the memory cells. A BiLSTM network extends the LSTM network by processing the input sequence in both the forward and backward directions, allowing the network to capture contextual information from both the past and future. BiLSTM networks have been widely used in NLP tasks such as sentiment analysis, named entity recognition, and sarcasm detection.

4.7.6 BERT Transformer

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that has achieved state-of-the-art results on a wide range of NLP tasks. BERT is based on the Transformer architecture, which is a type of deep learning model that is well-suited for processing sequential data. The key innovation of BERT is its use of a bidirectional transformer encoder to learn contextualized representations of words in a sentence. BERT is pre-trained on a large corpus of text using two unsupervised tasks: masked language modeling and next sentence prediction. Once pre-trained, the BERT model can be fine-tuned on a specific NLP task such as sarcasm detection. The general transformer model architecture is given in Figure 3.5.

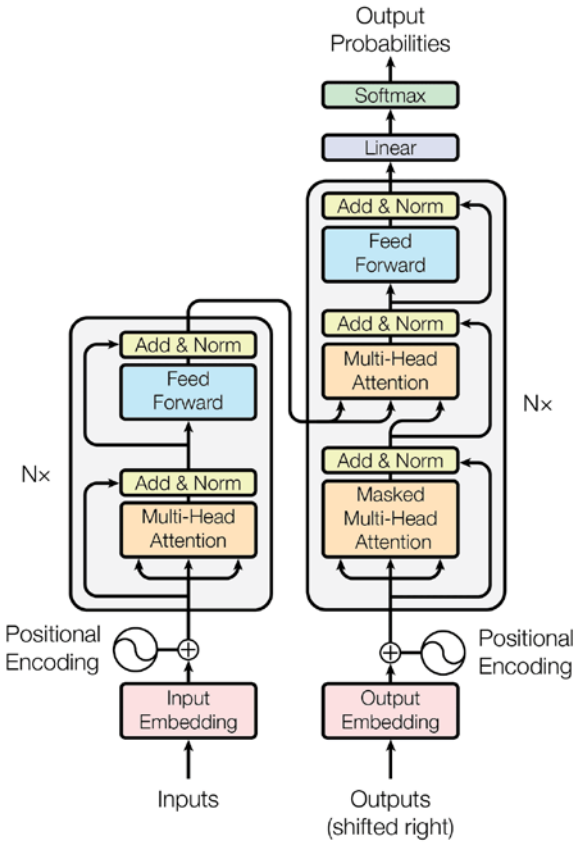

Figure 4.11 Transformer Model Architecture

(Source: Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *ArXiv*. /abs/1706.03762)

In the context of sarcasm detection, these models can be used in various ways. For instance, CNN and BiLSTM can be used to learn features from text data, and then a classifier can be trained on top of these features to detect sarcasm. On the other hand, BERT can be fine-tuned on a sarcasm detection dataset to learn task-specific representations of words and phrases. These representations can then be used to detect sarcasm with high accuracy.

CHAPTER 5

5. EXPERIMENTAL EVALUATION

5.1 Quantitative Results

In evaluating the performance of the diverse models employed for sarcasm detection, we scrutinize the results across three categories: Machine Learning (ML) Models, Deep Learning (DL) Models and Transformer Models.

5.1.1 Machine Learning Models

In this study, we evaluated the performance of several traditional machine learning models for sarcasm classification on two different datasets. The models included support vector machines (SVMs) with a linear kernel, decision trees, and random forests. We found that the SVM model with a linear kernel achieved the best results on both datasets, with an F1 score of 84.41% on dataset v2. The decision tree and random forest models also performed well, achieving F1 scores of over 70% on both datasets as it is shown in Table 5.1. These results suggest that traditional machine learning models can be effective for sarcasm classification, even in the presence of complex and nuanced language.

Table 5.1 Results for ML models

Model- ML	Dataset	Result
SVM-kernel:linear	v1	84.01% F1score
SVM-kernel:linear	v2	**84.41%** F1score
Decision Tree	v1	69.59% F1 score
Decision Tree	v2	70.92% F1 score
Random Forest	v1	81.49% F1score
Random Forest	v2	81.71% F1score

5.1.2 Deep Learning Models

We have also investigated the effectiveness of various deep learning models for sarcasm detection. In addition to the data augmentation, we also used our handcrafted features polarity score, label and category score, label when applying DL models. The models explored included FastText word embedding combined with a Bidirectional Long Short-Term Memory network (FastText+BiLSTM), FastText with a combination of Convolutional Neural Network and BiLSTM (FastText+CNN+BiLSTM), and GloVe embeddings used with various architectures like CNN, BiLSTM, and a combination of CNN and BiLSTM (GloVe+CNN, GloVe+BiLSTM, Glove+CNN+BiLSTM). The models were evaluated on four datasets: the original v1 and v2 datasets, and augmented versions of v1 and v2 (v1_aug and v2_aug).

The FastText+BiLSTM model achieved the highest F1 score of 91% on the v1 dataset, while the GloVe+CNN+BiLSTM model achieved the same score on the v1_aug,v2_aug and v2 datasets. These results suggest that the combination of word embeddings and BiLSTM architectures is effective for sarcasm detection, and that data augmentation can further improve performance along with the help of hand-crafted features where their weights are combined in the training phase. The FastText+CNN+BiLSTM model also achieved strong performance on the v2 and v2_aug datasets, with F1 scores of 89%, demonstrating the generalizability of the approach. Overall, the deep learning models outperformed traditional machine

learning models in sarcasm detection, highlighting their potential for this task.

Table 5.2 Results for DL models

Model- DL	**Dataset**	**Result**
Fasttext+BiLSTM	v1	**91%** F1 score
Fasttext+BiLSTM	v2	89% F1 score
FastText+CNN+BiLSTM	v2+v2_aug	89% F1 score
Glove+CNN+BiLSTM	v1_aug+v2_aug+v2	**91%** F1 score
Glove+BiLSTM	v1_aug+v2_aug+v2	89% F1 score
GloVe+CNN	v1_aug+v2_aug+v2	89% F1 score
GloVe+RNN	v1_aug+v2_aug+v2	88% F1 score

5.1.3 Transformer Models

We also have evaluated transformer architectures including BERT-base-uncased, RoBERTa-base-sentiment, RoBERTa-base-irony and DistilBERT. Uncased version of the models are just pre-trained version of the models. Besides, we have also used fine-tuned models trained for specific tasks such as sentiment classification and irony classification. By using our dataset we have fine-tuned all models to see the performance of transformer based models on the sarcasm classification task. RoBERTa-base-sentiment model is finetuned for sentiment analysis with the TweetEval benchmark. RoBERTa-based-irony is a fine-tuned version of roberta-base on the tweet_eval (irony) via tweetnlp python library. The models were tested on four datasets: the original v1 and v2 datasets, and augmented versions of v1 and v2 (v1_aug and v2_aug) created using nlpaug data augmentation techniques.

The BERT-base-uncased model achieved the highest F1 scores, reaching 97.68% on the v1_aug,v2 and v2_aug datasets and 96.15% on the v2 and v2_aug dataset. These results suggest that data augmentation can significantly improve the performance of transformer models for sarcasm detection. The RoBERTa-base-sentiment model also performed well, achieving 97.04% F1 scores on the augmented datasets plus v2 dataset. The results for applied transformer architectures are given in Table 5.3. Overall, the transformer models outperformed traditional machine learning models and deep learning models in sarcasm detection tasks.

Table 5.3 Results for Transformer Models

Model- Transformers	**Dataset**	**Result**
Bert-base-uncased	v1	93.82% F1 score
Bert-base-uncased	v2	93.11% F1 Score
Bert-base-uncased	v1_aug+v2+v2_aug	**97.68%** F1 score
Bert-base-uncased	v2+v2_aug	96.15% F1 Score
Bert-base-uncased	v1+aug	96.03% F1 Score
RoBERTa-base-sentiment	v1_aug+v2+v2_aug	97.04% F1 Score
RoBERTa-based-irony	v1_aug+v2+v2_aug	95.84% F1 Score
DistilBERT-uncased	v1_aug+v2+v2_aug	97.03% F1 Score

In conclusion, ML, DL, and Transformer models have all demonstrated the capability to detect sarcasm. Each architecture used in this study contributes uniquely to sarcasm detection. However, when comparing these approaches, it's important to note that Deep Learning and Transformer models outperforms traditional Machine Learning models in this particular task. ML models can be trained to identify patterns in sarcasm but their performance may be limited by the need for manual feature engineering and a predefined set of features. On the other hand DL models, especially those based on neural networks, have the ability to automatically extract and learn complex patterns and representations from data. This feature allows them to capture complex linguistic nuances associated with sarcasm, making them more effective than traditional ML approaches. Transformer architecture is an enhanced version of the deep learning architecture. In other words, transformer models are a specific type of deep learning architecture, and have demonstrated remarkable success in various NLP tasks, including sarcasm detection. Transformers, with their attention mechanisms and the ability to handle contextual information effectively, can capture long-range dependencies in language, improving their understanding of sarcastic expressions. We can explain the superiority of Transformer and Deep Learning when compared with Machine Learning using the following definitions.

- **Automatic Feature Learning:** Deep Learning models automatically learn hierarchical features from the data, eliminating the need for manual feature engineering.
- **Contextual Understanding:** Transformers, in particular, excel in understanding the contextual nuances of language, which is crucial for sarcasm detection where context plays a significant role.
- **Performance:** Deep Learning models, and especially Transformer-based models like BERT or Roberta, often achieve state-of-the-art performance in NLP tasks, including sarcasm detection, due to their capacity to handle large amounts of data and capture linguistic features.

In summary, while Machine Learning models can be employed for sarcasm detection, Deep Learning models and Transformer architectures perform better, due their ability to automatically learn complex patterns and understand contextual information. This generally results in superior performance in capturing the sarcastic expressions in natural language.

CHAPTER 6

6. CONCLUSION

6.1 Conclusion

The purpose of this study is to automatically detect sarcasm. To realize this goal, it is essential to understand the nature of sarcasm and the complexities associated with its detection. Sarcasm, considered as a form of irony commonly utilized to express negative opinions, poses a linguistic challenge due to its figurative nature. Detecting sarcasm becomes particularly challenging when conveyed in written form. Researchers recognized sarcasm detection as a complex text classification problem, introducing various approaches and methodologies to tackle this linguistic phenomenon.

This study conducts an extensive survey of prior research on sarcasm detection. We employ various ML, DL, and Transformer models, alongside hybrid neural network architectures, on news headlines datasets. Our evaluation of sarcasm detection models showcases that, within the ML category, traditional models such as SVM, Decision Trees, and RF achieved relatively high scores, with SVMs achieving the highest F1 score of 84.41% on dataset v2. Unfortunately, traditional ML models failed to achieve most of the state-of-art models. Nevertheless, despite the presence of complex features traditional ML models demonstrated effectiveness in sarcasm classification. On the other hand, Deep Learning models, including FastText+BiLSTM and GloVe+CNN+BiLSTM, outperformed traditional ML approaches, achieving respectively 91% in F1 scores. The utilization of word embeddings and BİLSTM networks showcased the efficacy of DL models in sarcasm detection. Additionally, data augmentation and the incorporation of hand-crafted features further enhanced

performance. Lastly, Transformer models, BERT-base-uncased, RoBERTa-base-sentiment, RoBERTa-base-irony, and DistilBERT performed marvelous results in detecting sarcasm. The BERT-base-uncased model achieved F1 scores of 97.68% and 96.15% respectively. The results underscore the transformative impact of Transformer architectures, leveraging attention mechanisms and contextual understanding to capture the sarcastic expressions effectively.

In summary, the performance metrics across our model presents a spectrum of results, ranging from 76.42% to 95.37% in F1 scores. Future research in automatic sarcasm detection could explore different ways to enhance model performance and address existing challenges. One of the possible ways can be fine-tuning transformer models, including pre-training on larger datasets and investigating different architectures, which remains crucial for achieving higher accuracy. Alongside, the existing datasets can be extended since finding a proper dataset is hard despite having vast amounts of data present in online platforms. In essence, this study contributes to the NLP field by providing valuable insights into sarcasm detection methodologies. The superior performance of DL and Transformer models, combined with the augmented datasets and handcrafted features, creates an optimized automatic sarcasm detection process. As we continue to advance in language processing, the usage of advanced architectures promise extensive results in sarcasm detection and other natural language tasks.

REFERENCES

Abaskohi, A., Rasouli, A., Zeraati, T., & Bahrak, B. (2022). UTNLP at SemEval-2022 Task 6: A Comparative Analysis of Sarcasm Detection using generative-based and mutation-based data augmentation. *arXiv preprint arXiv:2204.08198*.

Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU). *arXiv (Cornell University)*. Retrieved from https://arxiv.org/pdf/1803.08375.pdf.

Ali, R., Farhat, T., Abdullah, S., Akram, S., Alhajlah, M., Mahmood, A., & Iqbal, M. A. (2023). Deep Learning for Sarcasm Identification in News Headlines. *Applied Sciences*, *13*(9), 5586.

Barhoom, A., Abu-Nasser, B. S., & Abu-Naser, S. S. (2022). Sarcasm Detection in Headline News using Machine and Deep Learning Algorithms.

Bharti, S. K., Babu, K. S., & Jena, S. K. (2015, August). Parsing-based sarcasm sentiment recognition in twitter data. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*(pp. 1373-1380). Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051.

Dadu, T., & Pant, K. (2020, July). Sarcasm detection using context separators in online discourse. In *Proceedings of the Second Workshop on Figurative Language Processing* (pp. 51-55).

Filik, R., Țurcan, A., Thompson, D., Harvey, N., Davies, H., & Turner, A. (2016). Sarcasm and emoticons: Comprehension and emotional impact. *Quarterly Journal of Experimental Psychology*, *69*(11), 2130-2146.

Goyal, I., Bhandia, P., & Dulam, S. (2022). Finetuning for Sarcasm Detection with a Pruned Dataset. *arXiv preprint arXiv:2212.12213*.

Jariwala, V. P. (2020). Optimal feature extraction based machine learning approach for sarcasm type detection in news headlines. *International Journal of Computer Applications*, *975*, 8887.

Jayaraman, A. K., Trueman, T. E., Ananthakrishnan, G., Mitra, S., Liu, Q., & Cambria, E. (2022, December). Sarcasm Detection in News Headlines using Supervised Learning. In *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)* (pp. 288-294). IEEE. Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, *50*(5), 1-22. Kira E. Riehm, Kenneth A. Feder, Kayla N. Tormohlen, Rosa M. Crum, Andrea S. Young, Kerry M. Green, Lauren R. Pacek, Lareina N. La Flair, and Ramin Mojtabai. Associations Between Time Spent Using SocialMedia and Internalizing and Externalizing Problems Among US Youth. JAMA Psychiatry, 76(12):1266–1273,12 2019.

Kumar, A., Narapareddy, V. T., Srikanth, V. A., Malapati, A., & Neti, L. B. M. (2020). Sarcasm detection using multi-head attention based bidirectional LSTM. *Ieee Access*, *8*, 6388-6397.

Lahaji, M. N., Razak, T. R., & bin Ismail, M. H. (2023). Unveiling Sarcastic Intent: Web-Based Detection of Sarcasm In News Headlines. *Journal of Computing Research and Innovation*, *8*(2), 215-225.

Mandal, P. K., & Mahto, R. (2019). Deep CNN-LSTM with word embeddings for news headline sarcasm detection. In *16th International Conference on Information Technology-New Generations (ITNG 2019)* (pp. 495-498). Springer International.

Maynard, D. G., & Greenwood, M. A. (2014, March). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Lrec 2014 proceedings*. ELRA.

Mehndiratta,P. & Soni,D.(2019).Identification of Sarcasm in Textual Data: A Comparative Study. Journal of Data and Information Science,4(4) 56-83.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Misra, R. (2019). News headlines dataset for sarcasm detection. 2019.

Misra, R., & Arora, P. (2023). Sarcasm detection using news headlines dataset. *AI Open*, *4*, 13-18.

Moores, B., & Mago, V. (2022). A survey on automated sarcasm detection on Twitter. *arXiv preprint arXiv:2202.02516*.

Muresan, S., Gonzalez -Ibanez, R., G
of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology*, *67*(11), 2725-2737.

Namin, A. H., Leboeuf, K., Muscedere, R., Wu, H., & Ahmadi, M. (2009). Efficient hardware implementation of the hyperbolic tangent sigmoid function. *2009 IEEE International Symposium on Circuits and Systems*. https://doi.org/10.1109/iscas.2009.5118213.

Narayan, S. (1997). The generalized sigmoid activation function: Competitive

supervised learning. *Information Sciences*, *99*(1–2), 69–82. https://doi.org/10.1016/s0020-0255(96)00200-9. Nayak, D. K., & Bolla, B. K. (2022). Efficient deep learning methods for sarcasm detection of news headlines. In *Machine Learning and Autonomous Systems: Proceedings of ICMLAS 2021* (pp. 371-382). Singapore: Springer Nature Singapore.

Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., & Callison-Burch, C. (2015, July). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. 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)* (pp. 425-430).

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Potamias, R. A., Siolas, G., & Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, *32*, 17309-17320.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013, October). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 704-714). Seattle: Association for Computational Linguistics

Sagarika, N., Reddy, B. S., Varshitha, V., Geetanjali, K., Raju, N. G., & Kunaparaju, L. (2021). Sarcasm Discernment on Social Media Platform. In *E3S Web of Conferences* (Vol. 309, p. 01037). EDP Sciences.

Scola, E., & Segura-Bedmar, I. (2021). Sarcasm detection with BERT.

Sharma, D. K., Singh, B., Agarwal, S., Pachauri, N., Alhussan, A. A., & Abdallah, H. A. (2023). Sarcasm Detection over Social Media Platforms Using Hybrid Ensemble Model with Fuzzy Logic. *Electronics*, *12*(4), 937.

Shrikhande, P., Setty, V., & Sahani, A. (2020, November). Sarcasm detection in newspaper headlines. In *2020 IEEE 15th international conference on industrial and information systems (ICIIS)* (pp. 483-487). IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Veale, T., & Hao, Y. (2010). Detecting ironic intent in creative comparisons. In *ECAI 2010* (pp. 765-770). IOS Press.

Weiguo Fan and Michael D. Gordon. The power of social media analytics. Communications of the ACM, 57(6):74–81, 2014.

CURRICULUM VITAE

Publications

GİRGİN, A. B. A., & Gümüşçekiçci, G. (2022). From past to present: Spam detection and identifying opinion leaders in social networks. *Sigma Journal of Engineering and Natural Sciences*, 40(2), 441-463.

Gümüşçekiçci, G., Ezerceli, Ö., & Tek, F. B. (2020, September). Web service translating content into Turkish Sign Language. In *2020 5th International Conference on Computer Science and Engineering* (UBMK) (pp. 355-259). IEEE.