ISLAMIC UNIVERSITY OF TECHNOLOGY

Sarcasm Generation with Emoji: A Multi-Modular Framework Incorporating Valence Reversal & Semantic Incongruity

By

Tasmia Binte Sogir (180041201)

Faria Binte Kader (180041207)

Nafisa Hossain Nujat (180041213)

Supervisor	**Co-Supervisor**
Md. Kamrul Hasan, PhD	Md. Mohsinul Kabir
Professor	Assistant Professor
Dept. of CSE, IUT	Dept. of CSE, IUT

A thesis submitted in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering

Academic Year: 2021-2022

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
A Subsidiary Organ of the Organization of Islamic Cooperation.
Dhaka, Bangladesh.

May 2023

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Tasmia Binte Sogir, Faria Binte Kader and Nafisa Hossain Nujat under the supervision of Professor Dr. Kamrul Hasan and Mohsinul Kabir, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Tasmia Binte Sogir
Student ID - 180041201

Faria Binte Kader
Student ID - 180041207

Nafisa Hossain Nujat
Student ID - 180041213

May 2023

Co-supervisor:

Md. Mohsinul Kabir
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Supervisor:

Dr. Kamrul Hasan
Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Abstract

Sarcasm pertains to the subtle form of language that individuals use to express the opposite of what is implied. We present a novel architecture for sarcasm generation with emoji from a non-sarcastic input sentence. We divide the generation task into two sub tasks: one for generating textual sarcasm and another for collecting emojis associated with those sarcastic sentences. Two key elements of sarcasm are incorporated into the textual sarcasm generation task: valence reversal and semantic incongruity with context, where the context may involve shared commonsense or general knowledge between the speaker and their audience. The majority of existing sarcasm generation works have focused on this textual form. However, in the real world, when written texts fall short of effectively capturing the emotional cues of spoken and face-to-face communication, people often opt for emojis to accurately express their emotions. Due to the wide range of applications of emojis, incorporating appropriate emojis to generate textual sarcastic sentences helps advance sarcasm generation. We conclude our study by evaluating the generated sarcastic sentences using human judgement.

Keyword - Sarcasm Generation; Emoji; Commonsense Knowledge; Valence Reversal; Semantic Incongruity.

Acknowledgements

We would like to express our heartfelt thanks to Allah Subhanu Wata'ala for giving us the strength to finish this study and being with us when no one else was.

We would like to express our grateful appreciation for **Professor Dr. Kamrul Hasan** and **Mohsinul Kabir**, Department of Computer Science & Engineering, IUT for being our advisers and mentors. Their motivation, suggestions and insights for this research have been invaluable. Without their support and proper guidance this research would never have been possible. Their valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing algorithm, modification till the project implementation and finalization which helped us to do our thesis work in proper way. We are really grateful to them.

We are also grateful to **Dr. Hasan Mahmud**, Associate Professor, Department of Computer Science & Engineering for his valuable inspection and suggestions on our proposal of Sarcasm Generation framework.

This work is supported by
Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Contents

List of Figures

List of Tables

Dedicated to our parents and siblings for their unwavering and lifelong support in every step of our lives . . .

Chapter 1

Introduction

1.0.1 Overview

Sarcasm is defined as the use of remarks that clearly mean the opposite of what is said in order to hurt someone's feelings or to criticize something in a humorous way[1]. Sarcastic remarks are often challenging to interpret considering their literal meaning differs greatly from the speaker's actual intent. Compared to verbal or in-person conversations, textual sarcasm presents additional challenges due to the absence of visual cues, vocal tone etc.

non-sarcastic input	**sarcastic output with emoji**
I really hate walking in the rain.	I really love the outdoors walking in the rain. I sat feeling thoroughly miserable.😫
Mom is in a bad mood today.	Happy mothers day mom is in a well mood today. She sounded tense and angry.😠
That movie was bad.	That movie was awesome. Bad intelligence and political incompetence.😠

TABLE 1.1: Sample sarcastic outputs with emoji generated from non-sarcastic inputs

The presence of sarcasm makes it significantly harder for machines to understand the actual meaning of the textual data. This has motivated research in detecting sarcasm in textual data. In order to train machines to detect sarcasm, we need quality datasets that represent different aspects of sarcasm in text. Even though we have an abundance

[1] `https://dictionary.cambridge.org/`

of social media data and resources, it can be difficult to collect correctly labeled sarcastic texts. Instead, many research have tried to generate texts that can accurately express sarcastic notions (Joshi et al. [2], Mishra et al. [3], Chakrabarty et al. [1]). Many studies have also investigated strategies in incorporating sarcasm generation into chatbots (Joshi et al. [2, 4]).
Emojis, small ideograms that represent objects, people, and scenes (Cappallo et al. [5]), are one of the key elements of a novel form of communication due to the advent of social media. Using emojis within texts can give us additional cues on sarcasm, replicating facial expressions and body language, etc. Incorporating emojis with texts for training will let the machines catch these cues easily (Bharti et al. [6]). Subramanian et al. [7] observed that when emojis were included in the sentence, their emoji-based sarcasm detection model performed noticeably better.

This study introduces a novel framework where, upon receiving a non-sarcastic text as input, the text is transformed into a sarcastic expression using emojis. The incorporation of emojis is done to aid in clearly conveying the sarcastic intention of the text. Table 1.1 shows a few sample non-sarcastic input and sarcastic output pairs with emoji. In order to implement the architecture, we have focused on two major components: Sarcastic text generation and Emoji prediction for the text. For textual sarcasm generation, we are incorporating the works of Chakrabarty et al. [1] and Mishra et al. [3] and for Emoji prediction, a deep learning model fine tuned on OpenAI's CLIP (Contrastive Language-Image Pre-training)[2] Radford et al. [8] is used. The emoji prediction module along with the sarcasm generation module generates the final sarcastic text including emoji. This work provides two major contributions:

1. Propose a novel multi-modular framework for sarcasm generation incorporating the reversal of valence and semantic incongruity characteristics of sarcasm while also including appropriate emojis.
2. Create and publish a sarcastic corpora which can serve as valuable training data for sarcasm detection models.

As far as our understanding goes, there has been no previous framework proposed on textual sarcasm generation that also incorporates emojis. This framework can help chatbots gain a deeper understanding of sarcasm and produce more contextually relevant responses.

[2] `https://openai.com/research/clip`

1.0.2 Problem Statement

Sarcasm detection has been a trendy research domain for quite a while now. Sarcasm generation, however, remains fairly less explored. From the perspective of Natural Language Generation (NLG), sarcasm generation can be valuable in downstream applications such as conversation systems, recommenders, and online content generators Mishra et al. [3]. Sarcasm being an integral part of human interactions, systems for sarcasm generation can assist conversational agents like chatbots. In addition to the significance of sarcasm generation, emojis play a vital role in digital communication, especially on social media platforms. Generation of sarcastic utterances and the integration of emojis can not only make the task of sarcasm detection easier but also empower other intelligent systems. This research aims to develop a model which, given textual input, can generate a sarcastic sentence along with appropriate emoji with the help of an attention mechanism and common sense knowledge.

The research question we'll be trying to address is - can incorporating emoji based on sentiment and commonsense knowledge yield a better result in generating sarcastic sentences?

1.0.3 Motivation & Research Scope

While extensive research and development efforts spanning many years have been dedicated to addressing the challenge of sarcasm detection in textual data, only a few studies have aimed to develop systems for generating sarcasm. Sarcasm generation, according to Natural Language Generation (NLG), is an important research area that can improve conversational systems, recommenders, online content generators, and so on. For example, in a conversational setting, humans and machines could engage in more natural and interesting conversations if machines, like their human counterparts, can intermittently generate sarcastic responses. The few works that built frameworks for sarcasm generation, only focused on the textual generation of sarcasm. But, in real scenarios, written format of sarcasm is not only comprised of texts. In fact, emojis are widely used across social media platforms for delivering emotion signals. Hence, to mimic their human counterparts, only textual sarcasm generation may not be enough. As shown by Subramanian et al. [7], exploiting emojis for sarcasm detection in social media has been proven to improve the overall performance of sarcasm detection. Encompassing commonsense knowledge has also proven to improve sentiment analysis Agarwal et al. [9]. Based on this, we hypothesize that, incorporating emojis based on sentiment incongruity and commonsense should also prove to be useful in mitigating the performance of sarcasm generation.

1.0.4 Research Challenges

Compared to other language generation tasks, sarcasm generation is highly nuanced. This is because for a sentence to be considered sarcastic, it has to present an unusual situation where the unusualness (and hence, the sarcasm), arises from two implicitly opposing (incongruous), positive and negative, contexts Mishra et al. [3]. Since the majority of existing language generators are known to work on large-scale literal/non-sarcastic texts, they are typically unconcerned about potential collocations of context-specific incongruous phrases Joshi et al. [10]. Thus, figuring out contextually incongruous phrases are difficult for language generators Mishra et al. [3]. As for the task of emoji prediction, emoji representations differ across platforms, and new emojis are added to the Unicode standard on a regular basis, making it difficult to understand their meaning Fernández-Gavilanes et al. [11]. Moreover, the number of emojis used in most emoji prediction datasets are very low (the maximum being 64) which limits our option to choose. Emoji usage varies across cultures, social contexts, and author preference, so the original intent may be lost in interpretation Fernández-Gavilanes et al. [11].

1.0.5 Thesis Outline

This thesis is organized into 8 chapters. The first chapter provides a brief overview of our proposed framework and contributions, outlines the problem statements including possible research challenges and scope. The second chapter is the literature review subdivided into sections going through the works on types and characteristics of sarcasm, irony and sarcasm, sarcasm detection and generation, commonsense knowledge, emoji etc. The proposed methodology is explained in chapter 3 with the three modules discussed in separate subsections. The fourth chapter is the experimental setup describing the dataset in use, the model configuration and the evaluation technique and criteria. Chapter 5 is for the experimental analysis and results discussion. Chapters 6, 7 and 8 go over the limitations of our system and future work, finally ending with a brief conclusion.

Chapter 2

Literature Review

2.0.1 Types of Sarcasm

Several studies have been conducted in an attempt to categorize sarcasm into distinct types. Abulaish and Kamal [12] identified seven types of people: self-deprecating, brooding, deadpan, polite, obnoxious, manic, and raging. Sundararajan and Palanisamy [13] classified sarcasm into four types: Polite, Rude, Deadpan, and Raging. Kamal and Abulaish [14]) on the other hand, concentrated solely on detecting self-deprecating sarcasm, which includes self-referential humor. Oprea and Magdy [15] proposed 2 categories of sarcasm: intended and perceived, emphasizing the importance of treating them as distinct events. Their work included 2 forms of datasets: percieved (manually labeled) and intended (annotated remotely). The performance on the remotely annotated dataset was good. On the other hand, poor result was found in the dataset that was labeled manually, implying that the annotators might have failed in accurately understanding the authors' actual intentions.

2.0.2 Characteristics of Sarcasm

Studies have identified a variety of potential sources for sarcasm. According to Gerrig and Goldvarg [16], sarcasm stems from a situational disparity between what the speaker desires, believes, or expects and what actually happens. Incongruity between text and a contextual information is mentioned as a factor by Wilson [17]. Context Incongruity (Campbell and Katz [18]) is addressed in the works of Riloff et al. [19] who suggests that sarcasm arises from a contrast between positive verbs and negative situation phrases. Burgers et al. [20] formulates that for an utterance to be sarcastic, it needs to have one or more of these five characteristics:

1. **The sentence has to be evaluative**. The evaluative proposition can be already present (explicitly evaluative) or need to be inferred (implicitly evaluative). In an explicitly evaluative utterance, the evaluative term can be replaced by its semantic opposite term. For example in "It was a great idea to invest in company X!", *nice* can be substituted for *bad*. Implicitly evaluative sarcasm, like "Investing in company X really earned me a lot of money!", do not have such evaluative term that can be reversed.

2. **It should be based on the reversal of valence of the literal and intended meanings**. In a sarcastic sentence if the intended meaning was negative, it would be presented in a positive manner ("Great investment idea, John" where the *idea* was bad) and vice versa ("Bad investment idea, John", where the *idea* was good).

3. **It should have a semantic incongruity with the context, which may consist of common sense or general information that the speaker and the addressee share**. In the example "I just filled for bankruptcy because of your suggestion to invest in company X. That was a great investment idea.", the former utterance is not sarcastic, while the latter is.

4. **It should be aimed at some target**. The target can be the speakers themselves ("I had a great investment idea!"), the addressee ("You had a great investment idea!"), a third party who is neither the speaker nor the addressee ("John had a great investment idea!"), or a social group that encompasses the speaker, addressee and/or a third party ("You and John had a great investment idea!").

5. **It should be in some manner relevant to the communication scenario**. The relevance in sarcastic utterances can be described as the degree to which it "introduces information about an accessible discourse topic" Giora [21]. If an utterance is directly relevant, one inference is needed to discourse the topic ("That was a great investment idea"). If its indirectly relevant, more than one inference is needed ("I am rich now!").

Many studies focused on one or more of these characteristics.

2.0.3 Irony vs Sarcasm

As per our observations, several studies have mentioned the challenging nature of distinguishing between sarcasm and irony, even for human experts Ilić et al. [22], Dimovska et al. [23], Potamias et al. [24], Naseem et al. [25].However, a few works have attempted to distinguish between sarcasm and irony across various social media platforms. There exists a subtle distinction between sarcasm and irony in literature. Irony occurs when something contradicts our expectations If the expectation is black, the ironic result is white, not gray or off-white[1]. Sarcasm, on the other hand, is typically a form of negative and witty mockery directed at a specific individual[2]. Ling and Klinger [26] investigated the underlying structural differences between tweets that are ironic and sarcastic. On the basis of the use of hashtags, tweet structure, ratios of parts of speech, frequency of words and phrases, Khokhlova et al. [27] distinguished between sarcasm and irony in Twitter data and compared them with the NRC Word-Emotion Association Lexicon (EmoLex)[3]. Their suggestion was that tweets that are sarcastic might exhibit a more positive tone compared to tweets that are ironic. The SemEval-2018 workshop[4], which focused on semantic evaluation, introduced a shared task on detecting irony. Tweets collected using hashtags related to irony such as #irony, #sarcasm, and #not were included in the dataset. Dimovska et al. [23] investigated impact of different features on detecting irony and sarcasm individually utilizing the SemEval-2018 dataset. The model with the strongest performance in irony detection employed a linear Support Vector Machine (SVM) with the hashing vectorizer on word n-grams.

2.0.4 Sarcasm Detection

Sarcasm detection typically involves classifying a given text as sarcastic or non-sarcastic.It is a classification task in its simplest form. While this field of research focused on Natural Language Processing is relatively new, it holds great promise. Sarcasm detection plays a critical role in sentiment analysis, making it a necessary component in comprehending text sentiment (Maynard and Greenwood [28]).

The majority of studies on sarcasm detection rely on widely available datasets, such as those used by Riloff et al. [19], Khodak et al. [29] and Cai et al. [30]. Notably, among various social platforms, Twitter is the most common source for obtaining sarcasm detection datasets. Datasets from other platforms, such as Reddit, Amazon, and various discussion forums, were also utilized in this research domain. We also saw a shift in Sarcasm detection methodologies from rule-based approaches (Riloff et al. [19], Bharti et al. [31]), machine learning and deep learning approaches (Bharti et al. [32], Poria et al. [33], Ghosh and Veale [34]) and recently to transformed based approaches (Dadu and Pant [35], Kumar et al. [36]).

[1] `https://www.vocabulary.com/articles/chooseyourwords/irony-satire-sarcasm/`

[2] `https://grammar.yourdictionary.com/vs/irony-vs-sarcasm-types-and-differences.html`

[3] `https://colab.research.google.com/github/littlecolumns/ds4j-notebooks/blob/master/upshot-trump-emolex/notebooks/NRC%20Emotional%20Lexicon.ipynb`

[4] `https://github.com/Cyvhee/SemEval2018-Task3/tree/master/datasets`

2.0.4.1 Datasets

The most commonly observed type of datasets in studies related to sarcasm detection are short texts. These datasets primarily consist of social media content due to the character limit imposed by most platforms. Twitter and Reddit are the main sources for these short texts. Twitter, a microblogging platform known for its 280-character limit, boasts a user base of 330 million active monthly users spanning various age groups, making it an excellent data source for sarcasm and irony analysis[5]. Researchers often use the Twitter API[6] to gather data. For instance, Riloff et al. [19] created a dataset from Twitter consisting of 3,200 tweets, with 742 labeled as sarcastic and 2,458 as non-sarcastic. This dataset has been widely used in significant sarcasm detection studies by Riloff et al. [19], Joshi et al. [37], Ghosh and Veale [34], Tay et al. [38]. Reddit is another platform that provides slightly larger-sized contents compared to Twitter, yet still falls under the short text category as it has a length limit Joshi et al. [37]. With over 430 million monthly active users[7], mainly comprising younger individuals, Reddit serves as a valuable data source for sarcasm detection. One prominent dataset in this domain is the Self-Annotated Reddit Corpus (SARC) proposed by Khodak et al. [29][8]. The SARC dataset contains 1.3 million sarcastic and 532 million non-sarcastic posts from Reddit. This dataset has been used in subsequent research, including the work by Khodak et al. [29], Hazarika et al. [39]. Additionally, there are numerous other datasets consisting primarily of short texts, either derived from subsets of these two datasets or created using new Twitter or Reddit data. The SemEval-2018 shared task on sarcasm detection, for instance, made use of the Twitter and Reddit datasets in experiments conducted by Ilić et al. [22], Wu et al. [40]. Furthermore, researchers have explored other forms of short text data collection, such as book snippets and online comments, as documented by Joshi et al. [41], Bharti et al. [6].

Long texts have emerged as a prominent category of datasets in the field of sarcasm detection research. Various datasets have been compiled using product reviews from Amazon Dharwal et al. [42], Agrawal and An [43], Parde and Nielsen [44], which is the largest e-commerce platform housing numerous products and their corresponding reviews. Filatova [45] assembled a dataset comprising 437 sarcastic reviews and 817 regular reviews sourced from Amazon. Similarly, Mishra et al. [3] developed a dataset that incorporated other data types alongside Amazon reviews. Discussion forums are another valuable source of extensive textual data, and Oraby et al. [46] constructed a

[5] `https://financesonline.com/number-of-twitter-users/`

[6] `https://developer.twitter.com/en/docs/twitter-api`

[7] `https://earthweb.com/how-many-people-use-reddit/`

[8] `https://nlp.cs.princeton.edu/SARC/`

dataset containing 2496 sarcastic and non-sarcastic remarks extracted from debate forums. Notably, these discussion forum datasets are often utilized in conjunction with data obtained from various social media platforms. For instance, Bharti et al. [6] integrated data from Twitter, product reviews, comments, books, and discussion forums into their dataset. Additionally, news portals, Facebook posts, and Yelp reviews serve as reliable sources of long text sarcasm data. Subramanian et al. [7] leveraged both Twitter and Facebook datasets. The utilization of datasets comprising long texts has gained momentum in recent years due to the increasing popularity of e-commerce, review sites, and web portals beyond Twitter and Reddit.

We primarily focused on working with textual datasets, although we also encountered some cases involving multimodal datasets where text data was one of the modalities. During our research, we came across image data that had accompanying texts in the form of captions. A majority of the multimodal datasets we examined consisted of tweets that contained both images and texts. For instance, in their study, Cai et al. [30] developed a multimodal dataset[9] comprising 14,075 sarcastic tweets and 10,560 non-sarcastic tweets, each of which included images. This dataset has subsequently been utilized in various other studies Wang et al. [47], Pan et al. [48], Xu et al. [49]. Another well-known multimodal dataset was constructed by Schifanella et al. [50], incorporating texts and images sourced from Instagram, Tumblr, and Twitter.

Table 2.1, a compilation of various datasets utilized in different studies on sarcasm detection is presented. The annotation section of the table specifies the method employed for annotating the corpus, whether it was through manual annotation, the use of hashtags, or an unlabeled dataset. It should be noted that some researchers may have combined multiple datasets with varying types of annotations Ptáček et al. [51], Poria et al. [33], Oraby et al. [46].

Table 2.1: Summary of sarcasm detection datasets from different social media platforms

	Dataset					**Annotation**		
	Short Text	**Long Text**	**Image**	**Samples**	**Platform**	**Manual**	**Hashtag**	**None**
Filatova [45]		✓		1254	Amazon	✓		

[9] https://github.com/headacheboy/data-of-multimodal-sarcasm-detection

Riloff et al. [19]	✓			1600	Twitter	✓		
Ptáček et al. [51]	✓			920000	Twitter	✓	✓	
Barbieri et al. [52]	✓			60000	Twitter		✓	
Bamman and Smith [53]	✓			19534	Twitter		✓	
Amir et al. [54]	✓			11541	Twitter		✓	
Bharti et al. [6]	✓			1.5M	Twitter			✓
Joshi et al. [41]	✓			3629	Goodreads		✓	
Ghosh and Veale [34]	✓			41000	Twitter		✓	
Poria et al. [33]	✓			100000	Twitter	✓	✓	
Schifanella et al. [50]	✓		✓	600925	Instagram, Tumblr, Twitter		✓	
Zhang et al. [55]	✓			9104	Twitter		✓	
Felbo et al. [56]	✓			1.6B	Twitter			✓
Ghosh and Veale [57]	✓			41200	Twitter	✓		
Khodak et al. [29]	✓			533.3M	Reddit	✓		
Oraby et al. [46]		✓		10270	Debate forum	✓	✓	
Prasad et al. [58]	✓			2000	Twitter	✓		
Baziotis et al. [59]	✓			550M	Twitter			✓
Hazarika et al. [39]	✓			219368	Reddit	✓		

Ghosh et al. [60]	✓	✓		36391	Twitter, Reddit, Discussion Forum	✓	✓	
Ilić et al. [22]	✓	✓		419822	Twitter, Reddit, Debate Forum	✓	✓	
Tay et al. [38]	✓	✓		94238	Twitter, Reddit, Debate Forum	✓	✓	
Van Hee et al. [61]	✓			4792	Twitter	✓	✓	
Wu et al. [40]	✓			4618	Twitter	✓	✓	
Majumder et al. [62]	✓			994	Twitter		✓	
Cai et al. [30]			✓	24635	Twitter		✓	
Kumar et al. [63]	✓	✓		24635	Twitter, Reddit, Debate Forum		✓	
Subramanian et al. [7]	✓	✓		12900	Twitter, Facebook		✓	
Jena et al. [64]	✓			13000	Twitter, Reddit	✓	✓	
Potamias et al. [24]	✓			533.3M	Twitter, Reddit	✓	✓	

2.0.4.2 Features

In the study, it was observed that the most commonly utilized features in sarcasm detection on social media are lexical features. Lexical features encompass various types such as characters, n-grams, sentences, numbers, and hashtags. N-grams, specifically unigrams and bigrams, are extensively employed in natural language processing research Ling and Klinger [26], Joshi et al. [41], Schifanella et al. [50]. Researchers have also utilized character n-grams and complete sentences as features, as demonstrated by Dimovska et al. [23], who employed four distinct feature groups: character unigrams, character n-grams (with n ranging from 1 to 4), word unigrams, and word n-grams (with n ranging from

1 to 3). Additionally, certain experiments have incorporated numerical features Kumar et al. [65], Dubey et al. [66]. Hashtags have also been employed as features, as they can offer insights into the intention behind sarcasm Ghosh and Veale [34], Ilić et al. [22].

Pragmatic elements in text data typically refer to expressions and responses used to convey meaning. These can include emoticons and smileys, which are commonly employed to express emotions and distinguish between sarcastic and non-sarcastic statements Bharti et al. [6]. Researchers have incorporated emoticons and smileys as features in their studies, recognizing their significance. Additionally, ratings and reactions associated with social media content can also serve as indicators of sarcasm and irony. Consequently, pragmatic features have been frequently employed in sarcasm detection research involving social media. For instance, Das and Clark [67] utilized six types of user reaction counts on Facebook posts as one of their features, while Parde and Nielsen [44] included Amazon star ratings in their feature set. Felbo et al. [56] conducted experiments incorporating pragmatic features alongside lexical features. Furthermore, Onan [68] considered pragmatic features, along with lexical, implicit incongruity, and explicit incongruity-based features, in addition to word-embedding-based feature sets.

The detection of sarcasm relies on various hyperbole features such as interjections, intensifiers, and punctuation. These features play a crucial role in understanding the relationships between words and determining the significance of a sentence. In their study, Kumar et al. [63] examined five punctuation-based features to detect sarcasm in tweets, including the number of exclamation marks, question marks, periods, capital letters, and occurrences of the word "or". Another hyperbole feature worth considering is capitalization, which can indicate emphasis on specific n-grams and serve as a significant feature. Prasad et al. [58] also incorporated capitalization as a feature in their dataset.

Semantic features encompass various aspects such as the average word length, frequency of specific words, and sequence length, serving as supplementary details for a given statement. Chakrabarty et al. [1] utilized semantic incongruity as one of the features in their research endeavor.

In their study, Amir et al. [54] investigated various features, including syntactic features, in their experiment Amir et al. [54]. One widely employed technique for indicating the nature of words or tuples is the use of Parts-of-Speech (POS) tags.Ghosh and Veale [34] employed POS tags specifically for sarcasm detection. The method they utilized for extracting these tags was POS tagging, which involves converting a sentence into individual words or tuples and assigning tags to them[10].

[10] `https://www.geeksforgeeks.org/nlp-part-of-speech-tagged-word-corpus/`

Sentiment features play a significant role in identifying sarcasm, encompassing the polarity or emotional intensity of a statement. In their experiment, Khokhlova et al. [27] included sentiment polarity as one of the features. Sarcasm and irony are techniques employed to elicit specific sentiments in individuals, highlighting the importance of sentiment as a crucial feature in sarcasm detection. Various studies focusing on sarcasm detection, such as Joshi et al. [41], Ghosh and Veale [34], and Amir et al. [54], have acknowledged sentiment as a significant feature. In their experiment, Poria et al. [33] determined that sentiment and emotion features, along with baseline features, are the most valuable. Schifanella et al. [50] extracted subjectivity and sentiment scores as features from their multimodal dataset, incorporating them into their sarcasm detection model.

Lately, contextual features have gained significant traction and become increasingly popular. These features have greatly facilitated the identification of sarcasm, leading to their widespread adoption in numerous studies Amir et al. [54], Ghosh and Veale [57], Ghosh et al. [69], Sreelakshmi and Rafeeque [70], Poria et al. [33]. The Twitter and Reddit datasets provided in the FigLang2020 shared task[11] contain conversational contexts between users and their respective responses. The goal is to classify these responses as either sarcastic or non-sarcastic by utilizing the contextual information. These contextual features encompass various elements such as author or addressee information, audience, response, environment, and history. Hazarika et al. [39] incorporated both content and contextual information by employing user profiling to create user embeddings that capture behavioral traits indicative of sarcasm. Zhang et al. [55] conducted a sarcasm detection study using local and contextual features, demonstrating that the neural model achieved an accuracy of 78.55% using only local tweet features. However, when local and contextual features were combined, the accuracy of the neural model increased significantly to 90.74%. This finding underscores the importance of contextual features in sarcasm detection.

In this study, we primarily focused on textual datasets, although we also considered a few multimodal datasets where texts were accompanied by specific images as captions. Researchers who worked with multimodal datasets, such as the dataset presented by Cai et al. [30], primarily utilized two types of features: text features and image features. The inclusion of image features can provide valuable insights into the context and meaning of the associated texts. For instance, Schifanella et al. [50] employed a visual neural network that had been pretrained on ImageNet[12] to analyze multimodal sarcastic posts,

[11] `https://competitions.codalab.org/competitions/22247`

[12] `https://www.image-net.org/`

and they concluded that incorporating visual features enhanced the performance of the textual models.

In this particular study, the most widely used methods for extracting features are Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). The Bag-of-Words technique essentially converts a document into a collection of words, representing the most basic form of feature extraction in textual data. On the other hand, TF-IDF enhances the BoW method by assigning weights to words within the documents[13]. Ghosh et al. [69], Xiong et al. [71], Jamil et al. [72] employed the Bag-of-Words feature extraction technique in their studies. Similarly, Zhang et al. [55] utilized TF-IDF to extract certain features in their feature sets. TF-IDF was also utilized as a feature extraction technique by Dharwal et al. [42], Jain et al. [73] and Onan [68] in their respective research projects. However, it is important to note that both Bag-of-Words and TF-IDF have their limitations. By primarily focusing on word frequency, these methods fail to capture the contextual nuances present in a text, which can be crucial for sarcasm detection.

Various methods have been employed to convert words into vectors, such as different Word Embedding techniques. Among these techniques, Word2Vec is a widely popular approach that utilizes unsupervised learning to associate words with their most frequently occurring counterparts. To extract features from a multimodal dataset, Schifanella et al. [50] utilized the Word2Vec technique. Similarly, Joshi et al. [41] and Oraby et al. [46] employed Word2Vec for feature extraction. Two primary variations of Word2Vec are Continuous Bag of Words (CBoW) and Skip-Gram[14], each with its own methodology. CBoW predicts a target word based on its context, while Skip-Gram predicts a target word using its adjacent words[15]. Additional variations of Word2Vec, such as Doc2Vec and Emoji2Vec Eisner et al. [74], have been developed to generate vectors from different types of corpora. Khotijah et al. [75] utilized the Doc2Vec technique for feature extraction, while Subramanian et al. [7] focused on extracting and embedding emoji tokens using the Emoji2Vec technique. Another notable word embedding technique is GloVe (Global Vectors)[16], which effectively utilizes statistical information by training only on nonzero elements in a word-word co-occurrence matrix Pennington et al. [76]. This approach produces a vector space with meaningful substructure. Cai et al. [30] employed the GloVe technique for modality fusion in their multimodal dataset, rather than simply concatenating feature vectors from different modalities. FastText is another commonly used Word Embedding technique. While it bears similarities to

[13] `https://www.geeksforgeeks.org/feature-extraction-techniques-nlp/`

[14] `https://www.geeksforgeeks.org/word-embeddings-in-nlp/`

[15] `https://fasttext.cc/docs/en/unsupervised-tutorial.html`

[16] `https://nlp.stanford.edu/projects/glove/`

Word2Vec, FastText incorporates N-grams in conjunction with word collections, resulting in diverse word variations[17]. Mehndiratta and Soni [77] used Word2Vec, GloVe, and FastText techniques for feature extraction, and Onan [68] also employed these three techniques for the same purpose. Although Word2Vec and GloVe excel at mapping and labeling data, they tend to group together words that are actually antonyms, making sarcasm detection challenging. Additionally, these techniques encounter difficulties with out-of-vocabulary words, although FastText partially addresses this issue through the use of n-grams.

Several well-known machine learning models are currently being utilized to assist in the process of feature extraction. These models include Convolution Neural Network (CNN), Support Vector Machine (SVM), various iterations of Bidirectional Encoder Representations from Transformers (BERT), Long Short-Term Memory (LSTM), and Embeddings from Language Models (ELMo). In their work, Bharti et al. [6] employed a Hidden Markov model (HMM)-based algorithm for Part-of-Speech (POS) tagging. Numerous studies have made use of the NRC Word-Emotion Association Lexicon (EmoLex)[18], which consists of a collection of 14,182 English words (unigrams) categorized into positive or negative sentiments and labeled with eight primary emotions according to Plutchik's classification (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) Khokhlova et al. [27]. In their experiment, Agrawal and An [43] employed EmoLex to compute sentiment labels. Other notable methods for feature extraction include ResNet, SentiWordNet Baccianella et al. [78], SentiBank Borth et al. [79], TExtBlob[19], LIWC[20], and COMET Bosselut et al. [80].

In Table 2.2, a concise overview is presented, outlining various categories of features and the methods employed for extracting them in various studies focusing on sarcasm detection.

[17] `https://shorturl.at/EFM25`

[18] `https://colab.research.google.com/github/littlecolumns/ds4j-notebooks/blob/master/upshot-trump-emolex/notebooks/NRC%20Emotional%20Lexicon.ipynb`

[19] `https://textblob.readthedocs.io/en/dev/`

[20] `https://www.liwc.app/`

Table 2.2: Summary of types of features and feature extraction methods in sarcasm detection

	Type of Feature								**Extraction Method**					
	Lexical	**Pragmatic**	**Hyperbole**	**Semantic**	**Syntactic**	**Sentiment**	**Context**	**Image**	**BoW**	**TF-IDF**	**Word2vec**	**GloVe**	**FastText**	**Machine Learning Models**
Amir et al. [54]	✓		✓		✓	✓	✓		✓					✓
Bharti et al. [6]					✓	✓								✓
Ghosh and Veale [34]					✓	✓			✓					✓
Joshi et al. [41]	✓	✓	✓	✓		✓					✓	✓		
Poria et al. [33]				✓		✓								✓
Schifanella et al. [50]	✓			✓		✓		✓			✓			✓
Zhang et al. [55]	✓						✓			✓				
Felbo et al. [56]	✓	✓									✓			✓
Ghosh and Veale [57]							✓							
Ghosh et al. [69]	✓	✓				✓			✓					
Mukherjee and Bala [81]	✓				✓									
Prasad et al. [58]			✓		✓	✓								
Ghosh et al. [60]	✓	✓	✓		✓				✓					✓

Ilić et al. [22]	✓													✓
Hazarika et al. [39]							✓							✓
Tay et al. [38]	✓								✓					✓
Van Hee et al. [61]	✓									✓				
Wu et al. [40]					✓	✓			✓		✓			✓
Cai et al. [30]	✓							✓				✓		✓
Kumar et al. [63]		✓	✓									✓		
Majumder et al. [62]						✓								
Mehndiratta and Soni [77]	✓										✓	✓	✓	
Onan [68]	✓	✓									✓	✓	✓	
Chakrabarty et al. [1]	✓			✓			✓						✓	

2.0.4.3 Methodologies

Earlier works of Sarcasm Detection (González-Ibánez et al. [82], Davidov et al. [83]) approaches manly consisted some rule-based approaches or some pattern-based methods. Some notable works include Riloff et al. [19], Abulaish and Kamal [12], Bharti et al. [84] etc. Riloff et al. [19] used an bootstrapping algorithm which tries to search for a positive verb in a negative sentiment sentence. Bharti et al. [84] proposed 6 rule-based approaches to detect 6 types of sarcasm they noticed occurring in twitter. Abulaish and Kamal [12] employed a rule-based approach which leveraged regular expressions to establish filtering rules. Rather than relying solely on string matching, the initial layer of the model was designed to detect tweets containing self-deprecating sarcasm. As in rule-based approach, we do not need any training and sarcasm comes in various forms and situations, bounding them by some rules, cannot generalize sarcasm detection, so the detection shifted towards data-centric approach where we train some models to detect sarcasm.

Following rule-based approaches, a range of machine learning methods have been applied to sarcasm detection. SVM was utilized by Joshi et al. [85], Davidov et al. [83], Joshi et al. [41], Riloff et al. [19], González-Ibánez et al. [82], Oraby et al. [46] for sarcasm detection. Logistic Regression was employed by Abercrombie and Hovy [86], Bamman and Smith [53] in their sarcasm detection tasks. Some studies focused on feature engineering to enhance the performance of their machine learning models for sarcasm detection. Mukherjee and Bala [87] demonstrated that Naive Bayes outperformed Maximum Entropy when content and function words were utilized as features across multiple tweet datasets. Thakur et al. [88] employed Naive Bayes to show that POS tags were not particularly useful features for sarcasm detection. Naive Bayes was also used by Parde and Nielsen [44] for domain-general sarcasm detection on Twitter and Amazon product reviews. Bharti et al. [32] extracted lexical, hyperbolic, and behavioral features, along with universal facts using their proposed algorithm PBLGA (parsing-based lexical generation algorithm), and found that Decision Tree outperformed SVM, Naive Bayes, and Maximum Entropy for sarcasm detection. Sreelakshmi and Rafeeque [70] conducted an experiment where SVM with Radial Basis Function (RBF) Kernel outperformed Decision Tree. Prasad et al. [58] compared various machine learning approaches, incorporating their emoji and slang dictionary, and found that Gradient Boosting performed better than other classifiers. The combination of BERT and GloVe embeddings as features with Logistic Regression yielded strong results in the study conducted by Khatri et al. [89]. GloVe embeddings also performed better with Random Forest than SVM and Decision Tree (Eke et al. [90]). Random Forest and Ada-Boost based ensemble algorithms were employed for feature ensembling in sarcasm detection (Sundararajan et al. [91]). Banerjee et al. [92] explored minority oversampling techniques to address class imbalance issues and concluded that lazy learners like KNN were not suitable when minority oversampling was employed. To alleviate the laborious process of feature engineering, Di Gangi et al. [93] employed a Distributional Semantics approach using latent semantic analysis (LSA) (Landauer et al. [94]) and tested various machine learning classifiers (SVM, Logistic Regression, Random Forests, and Gradient boosting). Traditional machine learning models require handcrafted feature sets to expose data patterns to the learning algorithm and reduce complexity. Domain expertise and knowledge of data patterns are crucial for effective feature extraction in sarcasm detection tasks.

With the emergence of deep learning models, a noticeable shift has occurred in sarcasm detection towards the utilization of deep learning models or hybrid models combining deep and traditional machine learning approaches, as opposed to solely relying on traditional machine learning methods. The popularity of deep learning models stems from their ability to automatically extract features, eliminating the need for manual feature engineering. Poria et al. [33] were pioneers in this regard, employing a pre-trained CNN to automatically extract sentiment, emotion, and personality features, which were then fed into an SVM for classification. Various combinations of deep learning models have been explored to harness the semantic modeling capabilities. For instance, Ghosh and Veale [34] presented an experiment where a CNN model's output was fed into an LSTM layer before being passed to a DNN layer, resulting in improved performance compared to the works of Riloff et al. [19] and Davidov et al. [83], achieving an impressive F1-score of 0.92. Deep learning models consistently outperformed traditional machine learning models. Porwal et al. [95] and Salim et al. [96] both developed neural network (NN) models with RNN and LSTM components that automatically extract features. **(author?)** [Guo and Shah] demonstrated that LSTM surpassed baseline models such as Bag-of-Words, Naive Bayes, and even the traditional 'vanilla' neural network in their Reddit dataset. LSTM's ability to effectively learn sequential data without overfitting and leverage contextual information proved crucial for sarcasm detection. Moreover, Mehndiratta and Soni [77] revealed that, in the context of sarcasm detection, their single LSTM model outperformed their hybrid LSTM-CNN model, further emphasizing the effectiveness of deep learning models in this domain.

In the realm of sarcasm detection in social media conversations, Ghosh et al. [69] demonstrated that incorporating conversational context as an input to an LSTM model yields superior performance compared to using LSTM in isolation. Additionally, Ghosh et al. [60] employed the preceding sentence of a sarcastic utterance as context and showed that a multiple-LSTM architecture with sentence-level attention outperformed a single LSTM architecture. To capture the semantic relationship between candidate text and its context, Diao et al. [97] developed an end-to-end Multi-dimension Question Answering model. This model, based on Bi-LSTM and attention mechanism with multi-granularity representations, surpassed the state-of-the-art machine learning model by a significant margin in terms of F1 score. In exploring the impact of different contextual factors on sarcasm detection, Ren et al. [98] conducted experiments with two types of context: history-based context (views and opinions on events and individuals) and conversational context. They investigated two context-augmented CNNs called CANN-Key and CANN-ALL. CANN-KEY integrated key contextual information, while CANN-ALL incorporated all contextual information. Their findings revealed that for conversation-based contexts, CANN-ALL excelled in capturing subtle hints of sarcasm. However, history-based contexts yielded better results due to the relatively small number of conversation-based tweets. It is worth noting that these studies collectively highlight the significance of incorporating conversational context and various forms of context-based models, such as LSTM with attention and multi-granularity representations, for improved sarcasm detection in social media conversations.

To accommodate the variability in sarcastic expressions across individuals, Hazarika et al. [39] introduced CASCADE (ContextuAl SarCasm DEtector), a model that leverages contextual information from discussion forums and user embeddings. User embeddings encode stylometric and personality features using a CNN-based textual model. The inclusion of personality features alongside contextual information resulted in improved model performance. In a similar vein, Kolchinski and Potts [99] aimed to model author embeddings using two methods: a simple Bayesian approach that captures an author's raw propensity for sarcasm, and a dense embedding method that allows for intricate interactions between the author and the text. These author embeddings were incorporated into a baseline bidirectional RNN with GRU cells (BiGRU) to model user comments. While this method slightly underperformed compared to CASCADE on the full SARC dataset, it outperformed CASCADE on the posts from the r/politics subreddit within the same dataset. However, Misra and Arora [100] argued that models lacking common sense knowledge and information on current events fail to comprehend sarcasm effectively and instead rely solely on discriminative lexical cues. To address this, they removed user embeddings and focused on incorporating current events and common sense knowledge using an LSTM-CNN module. This modification improved the performance of baseline models by approximately 5%. It is worth emphasizing that dealing with longer sentences poses challenges for sequence-to-sequence models. These models compress all the information into a fixed-length vector, which can result in the loss of relevant information.

To address the challenge of capturing long-range dependencies, recent works have employed attention-based deep learning models. These models utilize an attention layer to learn the relevance of each word in a sarcastic statement, assigning different weights to individual words. Kumar et al. [101] introduced the Multi-Head self-Attention based Bidirectional LSTM (MHA-BiLSTM), which incorporates manually designed auxiliary features. This model outperformed a feature-rich SVM model that included semantic, sentiment, and punctuation features. The MHA-BiLSTM model exhibited a significant performance improvement, surpassing the SVM model by 4.45% and 7.88% on the balanced and imbalanced datasets, respectively. To enhance interpretability without compromising performance, Liu et al. [102] utilized GRU with Multi-Head self-Attention. This approach provided valuable cues for sarcasm detection while maintaining high accuracy. Kumar et al. [63] implemented another attention-based hybrid model that combined a soft-attention based BiLSTM with punctuation-based auxiliary pragmatic features. They integrated this hybrid model with a deep convolution network, achieving enhanced performance. Incorporating attention mechanisms has proven effective in addressing the challenge of long-range dependencies in sequence models. However, these models still face the limitation of sequential processing and the inability to leverage parallel processing. Transformer-based models, which will be discussed in later sections, offer promising solutions to these challenges.

GRU and LSTM models process words sequentially, which limits their ability to capture contrast, incongruity, and long-range dependencies across multiple sentences. To address this limitation, Tay et al. [38] proposed the Multi-dimensional Intra-Attention Recurrent Network (MIARN). This model utilizes intra-sentence relationships and incorporates the concept of compositional learning. MIARN surpassed other models such as NBOW, CNN, LSTM, ATT-LSTM (Attention-based LSTM), GRNN (Gated-RNN), and CNN-LSTM-DNN across all six datasets used in the study, demonstrating its superior performance. Building upon MIARN, Akula and Garibay [103] employed MIARN as encoders in their Dual-Channel Network (DC-Net). This approach allowed them to capture both the literal and deep meanings of sentiments, enabling the recognition of sentiment conflicts within input texts. In a different approach, Pan et al. [48] introduced snippet-level self-attention to model incongruity between sentence snippets. Their model consists of a convolution module (CNN), an importance weighting module, and a self-attention module. This approach proved particularly effective for Twitter datasets, outperforming long text datasets. These advancements in modeling techniques demonstrate the ongoing efforts to overcome the limitations of sequential word processing and to capture complex relationships, incongruity, and long-range dependencies in sarcasm detection tasks.

Multi-task learning is an innovative approach that involves training a single neural network to handle multiple classification tasks simultaneously. In the context of sarcasm detection, we came across several studies that employed multi-task learning. Majumder et al. [62] utilized multi-task learning for both sentiment classification and sarcasm detection. They utilized GRU with an attention mechanism for sentence representations and Glove word embeddings for word representations. The classification tasks were handled by two separate softmax layers, each dedicated to one of the tasks. By incorporating NTN (neural tensor network) into the multi-task classifier, they observed further improvements in performance, particularly in sarcasm detection. Similarly, Savini and Caragea [104] employed multi-task learning by utilizing sentiment classification as an auxiliary task to enhance the primary task of sarcasm detection. Both tasks shared the same BiLSTM model and made use of ELMo and FastText embeddings. However, they employed different Multi-layer Perceptron architectures and did not include user embeddings. This model outperformed systems that incorporated user embeddings, such as CNN-SVM and CUE-CNN. Although its F1-score was slightly lower than CASCADE, the difference was only 0.7%. These studies demonstrate the effectiveness of multi-task learning in sarcasm detection, where leveraging related tasks can lead to improved performance and enhanced understanding of sarcastic expressions.

Ensemble learning techniques have been employed in sarcasm detection to enhance performance and improve accuracy. In the work by Jain et al. [73], two ensemble learning methods, Random Forest and Weighted Ensemble, were utilized as sarcasm detection classifiers. The Weighted Ensemble model incorporates Naive Bayes, Linear Regression, and Random Forest as its component classifiers. The authors highlight that ensemble-based approaches tend to be more effective in terms of recall and precision, with their effectiveness closely tied to the individual classifiers' performance. Potamias et al. [105] proposed a Deep Ensemble Soft Classifier (DESC) consisting of three deep models: a BiLSTM, an AttentionLSTM, and a Dense NN. DESC achieved superior performance compared to all models published in the SemEval-2015 Sentiment Analysis task, demonstrating its effectiveness in sarcasm detection. Gupta et al. [106] employed a voting classifier that leverages majority voting to determine the best result among the outputs generated by multiple machine learning classifiers. This approach effectively combines the strengths of various classifiers to make accurate predictions. Lemmens et al. [107] proposed an ensemble method that incorporates additional features and predicted sarcasm probabilities from four component models. The component models include an LSTM with hashtag and emoji representations, a CNN-LSTM with casing, stop word, punctuation, and sentiment representations, an MLP based on InferSent embeddings, and an SVM trained on stylometric and emotion-based features. These ensemble learning techniques harness the advantages of multiple classifiers and additional features, leading to improved performance in sarcasm detection.

Transformer models have revolutionized natural language processing by effectively capturing both short and long-range dependencies using attention mechanisms, addressing limitations in previous architectures Fan et al. [108]. Recent architectures frequently rely on Transformer models, such as BERT and its variants like RoBERTa Liu et al. [109] and ALBERT Lan et al. [110]. Researchers have explored diverse applications of these models in sarcasm detection. Srivastava et al. [111] proposed a hierarchical BERT-based model for sarcasm detection, consisting of a context-summarization layer, a context-encoder layer, a CNN layer, and a fully-connected layer. model combined five pre-trained transformer models: BERT, RoBERTa, XLNet, RoBERTa-large, and ALBERT Gregory et al. [112] experimented with various transformers and found that BERT performed well as an individual model, but their best-performing ensemble. Kalaivani and Thenmozhi [113] compared BERT's performance with traditional machine learning and deep learning models, demonstrating BERT's superiority, especially in continuous conversation dialogues. To reduce data preprocessing overhead, Potamias et al. [24] introduced an end-to-end model that employed unsupervised pre-trained transformers in figurative language. They combined pre-trained RoBERTa with a RCNN to capture various forms of contextual information without handcrafted features or lexicon dictionaries. Javdan et al. [114] proposed a combination of aspect-based sentiment analysis and BERT for sarcasm detection in Twitter and Reddit. The individual BERT model performed best on Reddit, while LCF-BERT, an aspect-based sentiment classification method, achieved superior results on the Twitter dataset Zeng et al. [115]. Parameswaran et al. [116] fine-tuned publicly available BERT models, TD-BERT and BERT-AEN, for sarcasm target detection, showing that BERT models outperformed existing state-of-the-art models. Surprisingly, TD-BERT performed even better than BERT-AEN, suggesting that incorporating the target's position improved context understanding. Kumar et al. [36] introduced AAFAB, a model combining semantic encoding with BERT and high-quality manually extracted auxiliary features. Adversarial training, including perturbations to input word embeddings, enhanced parameter generalization. AAFAB outperformed several deep learning-based baseline models on balanced and imbalanced datasets Kumar et al. [36]. Lou et al. [117] proposed the Affective Dependency Graph Convolutional Network (ADGCN) framework, leveraging affective commonsense knowledge and dependency trees to construct affective and syntax-aware dependency graphs. BERT was used to learn vector representations, while multi-layer GCNs exploited affective dependencies for sarcasm detection. To collect less noisy sarcastic data using conversation cues, Shmueli et al. [118] introduced reactive supervision, a novel data collection technique. They created the SPIRS dataset, incorporating additional features and fine-tuned labels, enabling a new task of sarcasm perspective classification. Pre-trained BERT achieved superior performance compared to other deep learning methods in their evaluations. While transformer models offer parallel processing and faster computation, they face challenges in processing hierarchical inputs, as they lack the ability to leverage past representations of the input sequence to compute the current representation Fan et al. [108].

Table 2.3 shows a comparative analysis between the performances of various sarcasm detection systems. This should be noted, that the entries are not comparable to each other as the experiments were not done with the same datasets and conditions.

Table 2.3: Performance summary of various approaches used in sarcasm detection

	Data	**Architecture**	**Performance**			
			Accuracy	**F1-Score**	**Precision**	**Recall**
Davidov et al. [83]	Tweets	SASI (Semi-supervised Algorithm for Sarcasm Identification)	0.896	0.545	0.727	0.436
Gupta and Yang [119]	Tweets	CrystalNet		0.60	0.52	0.70
Bharti et al. [32]	Tweets	PBLGA with SVM		0.67	0.67	0.68
Mukherjee and Bala [87]	Tweets	Naive Bayes	0.73			
Jain et al. [73]	Tweets	Weighted Ensemble	0.853		0.831	0.298
Poria et al. [33]	Tweets	CNN-SVM		0.9771		
Ghosh and Veale [34]	Tweets	CNN-LSTM-DNN		0.901	0.894	0.912
Zhang et al. [55]	Tweets	GRNN	0.9074	0.9074		
Oraby et al. [46]	Tweets	SVM + W2V + LIWC		0.83	0.80	0.86
Hazarika et al. [39]	Reddit posts	CASCADE	0.79	0.86		
Ren et al. [98]	Tweets	CANN-KEY		0.6328		
		CANN-ALL		0.6205		
Tay et al. [38]	Tweets, Reddit posts	MIARN	Twitter: 0.8647	0.86	0.8613	0.8579

			Reddit: 0.6091	0.6922	0.6935	0.7005
Ghosh et al. [60]	Reddit posts	multiple-LSTM	0.7458	0.7607		0.7762
Diao et al. [97]	Internet arguments	MQA (Multi-dimension Question Answering model)		0.762	0.701	0.835
Kumar et al. [101]	Reddit posts	MHA-BiLSTM		0.7748	0.7263	0.8303
Kumar et al. [63]	Tweets	sAtt-BiLSTM convNet	0.9371			
Majumder et al. [62]	Text snippets	Multi task learning with fusion and shared attention		0.866	0.9101	0.9074
Potamias et al. [105]	reviews of laptops and restaurants	DESC (Deep Ensemble Soft Classifier)	0.74	0.73	0.73	0.73
Srivastava et al. [111]	Tweets, Reddit posts	BERT + BiLSTM + CNN	Twitter: 0.74			
			Reddit: 0.639			
Gregory et al. [112]	Tweets, Reddit posts	Transformer ensemble (BERT, RoBERTa, XLNet, RoBERTa-large, and ALBERT)		0.756	0.758	0.767

Potamias et al. [24]	Tweets, Reddit politics	RCNN-RoBERTa	Twitter: 0.91	0.90	0.90	0.90
			Reddit: 0.79	0.78	0.78	0.78
Javdan et al. [114]	Tweets	LCF-BERT		0.73		
	Reddit posts	BERT-base-cased		0.734		
Lee et al. [120]	Tweets, Reddit posts	BERT + BiLSTM + NeXtVLAD	Twitter	0.8977	0.8747	0.9219
			Reddit	0.7513	0.6938	0.8187
Baruah et al. [121]	Tweets, Reddit posts	BERT-large-uncased	Twitter	0.743	0.744	0.748
			Reddit	0.658	0.658	0.658
Avvaru et al. [122]	Tweets, Reddit posts	BERT	Twitter	0.752		
			Reddit	0.621		
Jaiswal [123]	Tweets, Reddit posts	Ensemble of several combinations of RoBERTa-large		0.790	0.790	0.792
Shmueli et al. [118]	Tweets	BERT	0.703	0.699	0.70 0.7741	
Dadu and Pant [35]	Tweets, Reddit posts	RoBERTa-large	Twitter	0.772	0.772	0.772
			Reddit	0.716	0.716	0.718
Kalaivani and Thenmozhi [113]	Tweets, Reddit posts	BERT	Twitter	0.722	0.722	0.722

			Reddit	0.679	0.679	0.679
Naseem et al. [25]	Tweets	T-DICE + BiLSTM + ALBERT	0.93	0.93		
Dong et al. [124]	Tweets, Reddit posts	context-aware RoBERTa-large	Twitter	0.783	0.784	0.789
			Reddit	0.744	0.745	0.749
Kumar and Anand [125]	Tweets, Reddit posts	context-aware RoBERTa-large	Twitter	0.772	0.773	0.774
			Reddit	0.691	0.693	0.699
Kumar et al. [36]	Tweets	AAFAB (Adversarial and Auxiliary Features-Aware BERT)		0.7997	0.8101	0.7896
Lou et al. [117]	Tweets, Reddit posts	ADGCN-BERT (Affective Dependency Graph Convolutional Network)	Twitter: 0.9031	0.8954		
			Reddit: 0.8077	0.8077		

Shared Tasks are collaborative efforts aimed at addressing specific problems, where organizers provide problem sets and datasets, and research teams work towards finding solutions. In the case of the SemEval-2017 sentiment analysis task (Task 4), Gupta and Yang [119] incorporated a sarcasm detection mechanism into their sentiment analysis model, Crystalnet, using a linear SVM classifier. Their approach demonstrated the importance of accurate sarcasm detection in understanding the true sentiment of a text Gupta and Yang [119].The 2nd Workshop on Figurative Language Processing 2020 featured a shared task focused on sarcasm detection. Two datasets from Twitter and Reddit were provided to investigate the significance of conversational context in sarcasm detection (Ghosh et al. [126]). Baseline scores for both datasets were established by the organizers. Many of the proposed solution approaches centered around

Transformer-based architectures, particularly BERT and RoBERTa, indicating a growing trend of utilizing pre-trained language models for sarcasm classification. Several systems explored the impact of context length (3, 5, 7 sentences) on sarcastic sentence recognition. Baruah et al. [121] found that for the Reddit dataset, the BERT classifier achieved its highest F-score when using only the response as input, without any contextual utterances. However, for the Twitter dataset, the highest score was obtained by incorporating the previous utterance as context alongside the response. Dadu and Pant [35] and Dong et al. [124] both observed performance improvements in both datasets by including the previous utterance with the response as input and employing RoBERTa-large, ALBERT, and BERT for classification. RoBERTa-large yielded the highest improvement in their experiments Baruah et al. [121], Dadu and Pant [35], Dong et al. [124].Jaiswal [123] implemented the second-best performing model for the shared task, where choosing the three latest utterances as input achieved the best result using both BERT and RoBERTa. Their classification model employed a "majority voting" approach, where multiple models predict the outcome and the label is determined by the majority output Jaiswal [123]. Avvaru et al. [122] achieved promising results by using the seven latest context utterances, along with the response, as input Avvaru et al. [122]. Lee et al. [120] presented the top-performing solution, an architecture combining BERT with pooling layers consisting of BiLSTM and NeXtVLAD. The key improvement came from a data augmentation technique called CRA (Contextual Response Augmentation), which expanded the dataset using easily accessible conversational context from unlabeled dialogue threads on Reddit and Twitter. Each response in the labeled training set was encoded using BERT trained on natural inference tasks, as introduced in the work of Reimers and Gurevych [127].

In our investigation, it has become apparent that the field of study has extended beyond the realm of sarcasm detection. Dubey et al. [66] were the pioneers in proposing diverse rule-based, machine learning, and deep learning models for detecting sarcasm in the numerical components of input texts Dubey et al. [66]. Their deep learning model, which utilized CNN-FF with an attention mechanism, outperformed other models in the experiment. Patro et al. [128] adopted a deep learning approach to determine the target of sarcasm by passing word embeddings through either a bidirectional LSTM (Bi-LSTM) layer or a target-dependent LSTM (TD-LSTM) layer Patro et al. [128]. Similarly, there have been a few other endeavors to identify the intended target of sarcasm. Joshi et al. [37] were the first to tackle this task and employed SVMperf with two rule-based extractors as their classifier Joshi et al. [37].In recent years, there have been attempts to generate computational sarcasm, which poses a challenge as the generated utterances need to possess the characteristics of sarcastic texts. Mishra et al. [3] pioneered automatic sarcasm generation by relying on the theory of context incongruity and anticipating input with a negative sentiment. Their model utilized LSTM and was compared against SarcasmBot, UNMT, Monoses, ST, and FLIP Mishra et al. [3]. Chakrabarty et al. [1] harnessed the power of Transformer models to propose an unsupervised sarcasm generation technique, incorporating valence reversal and semantic incongruity—two key features of sarcasm—into sentences. They employed RoBERTa-large to incorporate semantic incongruity and found that their system generated sarcastic sentences 34% better than human judges Chakrabarty et al. [1]. Dubey et al. [129] also attempted to generate a non-sarcastic interpretation of sarcastic input text using rule-based, deep learning-based, and machine learning-based architectures. They observed that their statistical machine translation-based approach, utilizing Moses (an Open Source Toolkit for Statistical Machine Translation), outperformed other approaches on the first dataset Dubey et al. [129].

2.0.4.4 Trends

Figure 2.1 depicts the observed trends in sarcasm detection. It highlights six significant milestones - 1. Foundational Research, 2. Pattern & Rule-based Methods, 3. Distant Supervision with Hashtags, 4. Integrating Context, 5. Deep Learning and 6. Transformers.

The initial research on sarcasm detection dates back to a study conducted by Tepperman et al. [130]. Subsequently, several investigations explored supervised and semi-supervised methodologies, aiming to uncover patterns and utilize them as features for statistical or rule-based classifiers. With the rise of Twitter as a valuable data source, the practice of hashtag-based remote monitoring gained widespread adoption. As research progressed, there emerged a trend of incorporating contextual cues like author information, audience dynamics, conversational context, visual data, and other relevant factors. In recent times, there has been a notable inclination towards deep learning and transformer-based techniques among researchers in this field. We discuss the more recent trends below - Integrating Context, Deep Learning, and Transformers.

Integrating Context

In recent times, there has been a growing interest in considering the contextual information when predicting text classifications. Throughout this section, we will refer to the text that requires classification as the "target text," while "context" will encompass any additional information related to it. Contextual details can include conversational context, author context, visual context, target context, or cognitive features Ghosh et al.

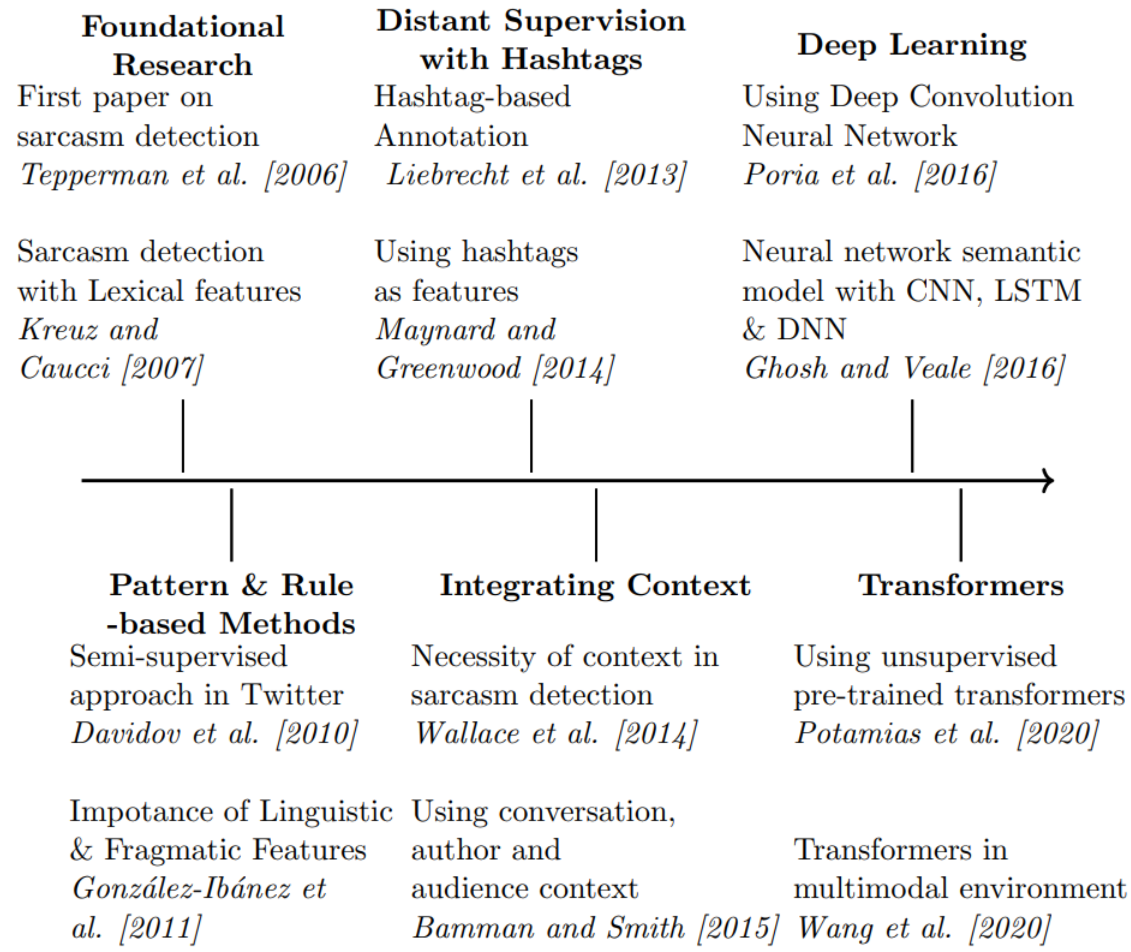
Foundational Research
First paper on sarcasm detection
Tepperman et al. [2006]
Sarcasm detection with Lexical features
Kreuz and Caucci [2007]
Distant Supervision with Hashtags
Hashtag-based Annotation
Liebrecht et al. [2013]
Using hashtags as features
Maynard and Greenwood [2014]
Deep Learning
Using Deep Convolution Neural Network
Poria et al. [2016]
Neural network semantic model with CNN, LSTM & DNN
Ghosh and Veale [2016]
Pattern & Rule -based Methods
Semi-supervised approach in Twitter
Davidov et al. [2010]
Impotance of Linguistic & Fragmatic Features
González-Ibánez et al. [2011]
Integrating Context
Necessity of context in sarcasm detection
Wallace et al. [2014]
Using conversation, author and audience context
Bamman and Smith [2015]
Transformers
Using unsupervised pre-trained transformers
Potamias et al. [2020]
Transformers in multimodal environment
Wang et al. [2020]

Figure 2.1: Trends in sarcasm detection throughout the years

[60].

The significance of context in detecting sarcastic texts was initially explored by Wallace et al. [131]. They observed that texts misclassified by machine learning algorithms using the Bag-of-Words (BoW) method in their Reddit irony corpus[21] were often the same texts for which human annotators requested additional context. Based on these findings, they argued that if human annotators require context, then providing contextual information to machine learning algorithms would be beneficial. This discovery inspired subsequent studies to incorporate context into their sarcasm detection algorithms Joshi et al. [132], Kolchinski and Potts [99], Plepi and Flek [133].

In our research, we encountered several architectural approaches that consider context. Specifically, we have come across three types of contexts: topical context, authorial context, and conversational context.

- Topical Context: Topical context, as the term suggests, pertains to the subject or theme of the target text. In a study by Wang et al. [134], sarcasm detection

[21] `https://github.com/bwallace/ACL-2014-irony`

was approached as a sequential classification task, with the consideration of such topic-based context. The researchers curated a Twitter dataset by collecting the entire tweet sequence, including multiple tweets with the same hashtag preceding the target tweet, to provide topic-based context. Another approach proposed by Joshi et al. [132] involved utilizing a topic model to identify topics that exhibited a higher prevalence of sarcasm. They employed a Twitter dataset and explored topic-level sentiment analysis to identify the sentiment associated with these sarcasm-prevalent topics. Their findings highlighted certain topics, such as 'work,' 'gun laws,' and 'weather,' which were more frequently associated with sarcasm.

- Authorial Context: Authorial context refers to the valuable insights provided by the author of the target text. In a study by Tay et al. [38], sarcasm was found to be associated with the disparity between the expressed emotion and the author's circumstances or context. Bamman and Smith [53] leveraged extra-linguistic data, including the author's historical salient terms, historical topics, profile information, historical sentiment, and profile unigrams, to capture authorial context. By combining these features with the context derived from the intended (or perceived) audience and the ongoing conversation, they achieved a significant improvement in accuracy compared to previous studies. This breakthrough established a new benchmark for sarcasm detection in social platforms. citemukherjee2017sarcasm aimed to capture the unique writing style of the author and discovered that certain authorial traits, such as function words and part-of-speech n-grams (particularly function words), played a critical role in identifying sarcasm. Ghosh and Veale [57] proposed the consideration of the author's mood. They employed a deep neural network architecture to model the author's emotional state at the time of utterance creation, utilizing mood indicators extracted from the author's most recent tweets. In addition, they modeled the context by incorporating attributes derived from the proximate cause of the response utterance. These investigations into authorial context have significantly advanced the field of sarcasm detection, setting new standards for accuracy in detecting sarcasm within social platforms.

- Conversational Context: By conversational context, we are referring to the interaction and discussion that takes place between the author and the audience of the target text. To capture this conversational context, Bamman and Smith [53] extracted binary indicators of pairwise Brown features between the original message and its corresponding response. Similarly, Wang et al. [134] collected complete tweet threads, including preceding tweets that contribute to the conversation with other users. Ghosh et al. [69] employed both of these methods to construct their conversational context dataset, which comprised 25,991 instances. Ren et al. [98] utilized the same dataset created by Wang et al. [134]. Since Wang et al. [134]

employed SVM$^{\text{multiclass}}$ Altun et al. [135] and SVM$^{\text{hmm}}$ Vanzo et al. [136] in their study, Ren et al. [98] opted for a neural network architecture, specifically CNN, to evaluate the impact of context. On the other hand, Ghosh et al. [60] selected LSTM and conditional LSTM networks to assess the effectiveness of conversational context in sarcasm detection. They considered the previous turn, the succeeding turn, or both as components of the conversational context. Additionally, conversational context has been widely employed in Transformer-based approaches, which will be further discussed in the Transformers section.

Deep Learning

The utilization of deep learning models in Natural Language Processing (NLP) research has a history of about a decade, with the first instance dating back to 2011 by Collobert et al. [137]. However, it was in 2016 when deep learning was first employed in the field of sarcasm detection, as demonstrated by Poria et al. [33]. Their approach involved training three models, namely sentiment, emotion, and personality, using Convolutional Neural Networks (CNNs). Each model was trained on its respective dataset, and the extracted features from these pre-trained models were then fed into a Support Vector Machine (SVM) for text classification. Notably, their architecture outperformed previous state-of-the-art methods. Additionally, their framework highlighted the significance of sentiment shifting, emotion, and personality traits in sarcasm detection.
Another notable contribution in this domain came from Amir et al. [54], who proposed a CNN-based model that incorporated prior utterances to learn and leverage user embeddings. Instead of treating sarcasm detection and sentiment analysis as separate tasks, Majumder et al. [62] developed an architecture that demonstrated and utilized the correlation between the two. By incorporating NTN (Neural Tensor Network) fusion, their multitask framework improved the performance of sarcasm detection. Furthermore, the inclusion of an attention network shared by both tasks enhanced the performance of sentiment classification.
With the advent of deep learning, researchers started exploring different types of data beyond traditional text for training models. Two such forms of data, namely visual and numerical data, have been extensively discussed in the subsequent subsections.

- Visual Data: Determining sarcasm solely from textual language can sometimes be challenging without understanding the underlying meaning of the text. Consider the following tweet as an example: ”Will be at the office in no time,” accompanied by an image depicting being stuck in a traffic jam. Without the contextual information provided by the image, this tweet may not be classified as sarcastic. To explore such scenarios, researchers began incorporating image data to enhance sarcasm detection.

The impact of visual content on sarcasm detection in social media was first empirically investigated by Schifanella et al. [50]. They employed deep learning techniques to combine a deep network-based representation of the image data by using unigrams as textual inputs. Another notable work in this domain is the Hierarchical Fusion Model proposed by Cai et al. [30]. Their architecture incorporated image features into sarcasm detection. They utilized ResNet to extract regional vectors from images and predict five attributes for each image. Text vectors were obtained using Bi-LSTM. These feature vectors were then reconstructed using raw vectors and guidance vectors. Subsequently, these refined vectors were fused together into a unified vector. The deep learning architecture presented by Cai et al. [30]. established a benchmark for multi-modal sarcasm detection, showcasing the potential of incorporating image data.

- Numerical Data: The need to detect sarcasm expressed through numbers has led to the emergence of a specific research trend. Instances like "Love driving 3 hours to work every day" or "Started the day with 22% charge on my phone. Today's gonna be great!" highlight the importance of understanding the role of numbers in identifying underlying sarcasm. Kumar et al. [65] conducted a comprehensive study involving various deep learning approaches, alongside rule-based and machine learning-based methods, to address such cases. Their CNN-FF (Convolutional Neural Network followed by a Fully Connected Layer) model, based on deep learning, achieved the most promising results in their experiments. Dubey et al. [66] also proposed deep learning architectures specifically designed to handle sarcasm conveyed through numbers. They introduced a CNN-FF model and an attention network, both of which exhibited improvements compared to previous works. Notably, the CNN-FF model outperformed the attention network, achieving an impressive F1-score of 0.93 as opposed to 0.91. These advancements highlight the significance of incorporating numerical information in sarcasm detection tasks.

Transformers

Another emerging trend in sarcasm detection involves the utilization of transformers. Transformers are deep learning models that leverage self-attention mechanisms to weigh input data based on learned relevance Vaswani et al. [138]. While transformers were initially introduced in 2017, their application in sarcasm detection is relatively recent. The pioneering work by Potamias et al. [24] introduced a methodology based on unsupervised pre-trained transformers. Their approach, RCNN RoBERTa, combines Recurrent CNN with a pre-trained transformer-based network architecture, surpassing state-of-the-art methods such as BERT, XLnet, ELMo, and USE. This breakthrough spurred a growing trend of employing pre-trained language models for sarcasm classification tasks. Transformers have found extensive usage in various applications within this field. In this discussion, we will delve into two notable cases of transformer-based architectures: 1) leveraging conversational context and 2) incorporating Multiple Modals.

- Utilizing Conversational Context: Transformer-based models have been widely adopted for sarcasm detection, incorporating contextual information Gregory et al. [112], Javdan et al. [114], Avvaru et al. [122], Dong et al. [124]. Among various types of context, conversational context has been prominently utilized in transformer-based architectures, as observed in our research. The significance of conversational context can be seen in the 2nd Workshop at the Figurative Language Processing 2020 shared task (FigLang2020[22]), where both Twitter and Reddit datasets were provided, incorporating conversational context. This contextual information encompassed immediate context (previous dialogue turn) as well as the entire dialogue thread, whenever available. Given the ability of transformers to capture relationships in sequential data effectively, many teams opted for architectures that incorporated transformer layers to leverage the provided context.
  In the FigLang2020 shared task, Dong et al. [124] proposed a model utilizing deep transformer layers that fully exploited the conversational context, achieving a baseline F1 score of 0.67 for the Twitter dataset and 0.6 for the Reddit dataset Ghosh et al. [60]. Another approach presented by Lee et al. [120] involved stacking a transformer encoder with BiLSTM Schuster and Paliwal [139] and NeXtVLAD Lin et al. [140] layers. They introduced a data augmentation technique called Contextual Response Augmentation (CRA) to generate new training samples, leveraging the conversational context from an unlabeled dataset. Additionally, they explored multiple context lengths using a context ensemble method. As a result, they observed a significant improvement in F1 scores, achieving 0.931 and 0.834 for the Twitter and Reddit datasets, respectively.

- Using Multiple Modals: In recent times, the integration of transformer-based models with image encoders has gained traction in multimodal sarcasm detection Cai et al. [30]. The Hierarchical Fusion Model proposed by Cai et al. [30] stands out as one of the pioneering works that incorporated multiple modalities in their sarcasm detection architecture. As mentioned previously, their architecture takes into account text, image, and image attribute features, which are then reconstructed and

[22] `https://competitions.codalab.org/competitions/22247`

fused. However, this reconstruction process might lead to the omission of certain details.

With the introduction of transformer-based models, the practice of pretraining models on image-text data has become popular Lu et al. [141, 142], Alberti et al. [143]. However, Wang et al. [47] argued that instead of solely pretraining BERT on image-text data and ResNet on image data, a larger text corpus should be used to pretrain BERT, and ResNet should be pretrained on a larger image dataset. In light of this perspective, they developed an architecture that directly employs pretrained BERT and pretrained ResNet without additional pretraining, establishing a bridge between the two. This approach offers architectural flexibility, as any transformer-based model can replace BERT, and ResNet can be substituted with other visual models as well.

Another attempt to enhance the Cai et al. [30] architecture was made by Pan et al. [48]. Their focus was on the incongruity among images, text, and hashtags. They established a relationship between text and hashtags using a co-attention matrix and performed text-image matching by utilizing BERT for text encoding and ResNet-152 for image encoding. Subsequently, the results of these two relationships were compared. This integration of transformer-based encoders with image encoders yielded significant improvements in the performance of multimodal sarcasm detection Pan et al. [48].

2.0.5 Sarcasm Generation

Compared to sarcasm detection, research on sarcasm generation is still in its early stages. Joshi et al. [2] introduced SarcasmBot[23], a chatbot that caters to user input with sarcastic responses. SarcasmBot is a sarcasm generation module with eight rule-based sarcasm generators where each of the generators produces a different type of sarcastic expression. During the execution phase, one of these generators is selected based on user input properties. Essentially, it yields sarcastic responses rather than converting a literal input text into a sarcastic one, the latter one being a common practice in future research. This method was later utilized in the author's subsequent work (Joshi et al. [4]) where they built SarcasmSuite, a web-based interface for sarcasm detection and generation.

The first work on automatic sarcasm generation conditioned from literal input was performed by Mishra et al. [3]. The authors relied on the sarcasm characteristics of Context Incongruity mentioned by Riloff et al. [19] and employed information retrieval-based

[23] `https://github.com/adityajo/sarcasmbot/`

techniques and reinforced neural seq2seq learning to generate sarcasm. They used unlabeled non-sarcastic and sarcastic opinions to train their models, where sarcasm was formed as a result of a discrepancy between a situation's positive sentiment context and negative situational context. A thorough evaluation of the proposed system's performance against popular unsupervised statistical, neural, and style transfer techniques showed that it significantly outperformed the baselines taken into account. However, their models were trained using unlabeled non-sarcastic and sarcastic opinions and they only utilized negative sentiment sentences as input to convert into sarcastic utterances. But sarcasm can also arise from using a negative utterance to deliver a positive sentiment.

Chakrabarty et al. [1] introduced a new framework by incorporating context in the forms of shared commonsense or world knowledge to model semantic incongruity. They based their research on the factors addressed by Burgers et al. [20]. Their architecture is structured into three modules: Reversal of Valence, Retrieval of Commonsense Context, and Ranking of Semantic Incongruity. With this framework they were able to simulate two fundamental features of sarcasm: reversal of valence and semantic incongruity with the context. However, they opted for a rule-based system to reverse the sentiments. The authors also noticed that in a few cases, the simple reversal of valence strategy was enough to generate sarcasm which meant the addition of context was redundant.

Recent similar works in the field include that of Oprea et al. [144] where they developed a sarcastic response generator, Chandler, that also provides explanations as to why they are sarcastic. Das et al. [145] manually extracted the features of a benchmark pop culture sarcasm corpus and built padding sequences from the vector representations' matrices. They proposed a hybrid of four Parallel LSTM Networks, each with its own activation classifier which achieved 98.31% accuracy among the test cases on open-source English literature. A new problem of cross-modal sarcasm generation (CMSG) that creates sarcastic descriptions of a given image was introduced by Ruan et al. [146]. However, these studies have only focused on generating textual sarcastic sentences, but as described by Subramanian et al. [7], incorporating emojis improved the overall performance of sarcasm detection and thus can be a potential research scope.

2.0.6 Commonsense Knowledge Generation

Commonsense knowledge is information that encapsulates practical knowledge about how the world works that is universally accepted by humans but is usually stated implicitly. In recent years, the development of pre-trained language models (PLMs) has sparked a great deal of interest in the NLP community in determining what kind of commonsense knowledge PLMs possess and how far such knowledge can be used to address commonsense knowledge generation tasks (Petroni et al. [147], Davison et al. [148], Zhao et al. [149]). The task of commonsense knowledge generation can be divided into two broad categories - Knowledge Base Generation and Constrain Commonsense Text Generation. A Knowledge Base is a collection of relational facts, each of which is represented as a triple $< s, r, o >$, where s is the SUBJECT, r is the RELATION, and o is the OBJECT. According to Bhargava and Ng [150], one of the most successful knowledge generation approach with pre-trained language models is arguably Commonsense Transformers (COMET) (Bosselut et al. [80]). Given s and r, COMET can be used to generate o after being pre-trained on a knowledge base such as ATOMIC (Sap et al. [151]), which is a large-scale Knowledge Graph consisting of textual descriptions of inferential knowledge (if-then relations), or ConceptNet (Speer et al. [152], Singh et al. [153]), which represents (mostly taxonomic) commonsense knowledge as a graph of concepts (words or phrases) connected by relations (edge types). Later on, based on COMET, the authors built a general-purpose commonsense knowledge graph (CSKG), COMET-ATOMIC$^{20}_{20}$ (Hwang et al. [154]), with 1.33M everyday inferential knowledge tuples about entities and events. It contains knowledge that is not readily available in pre-trained language models. COMET-ATOMIC$^{20}_{20}$ offers 23 commonsense relations types. Using knowledge distillation (Hinton et al. [155]) technique, a machine trained 1.5B parameters commonsense model, COMET$^{\mathrm{DIS}}_{\mathrm{TIL}}$ (West et al. [156]), was built upon COMET. Knowledge distillation was applied on the general language model GPT-3. Three variants of COMET$^{\mathrm{DIS}}_{\mathrm{TIL}}$ are available - COMET$^{\mathrm{DIS}}_{\mathrm{TIL}}$, COMET$^{\mathrm{DIS}}_{\mathrm{TIL}}$ + critic$_{\mathrm{low}}$ and COMET$^{\mathrm{DIS}}_{\mathrm{TIL}}$ + critic$_{\mathrm{high}}$. Some common application of commonsense knowledge generation include essay generation (Yang et al. [157]), story and story ending generation (Guan et al. [158, 159]), response generation (Zhou et al. [160, 161]), conversation or dialogue generation (Zhou et al. [162], Wu et al. [163], Zhang et al. [164], Wu et al. [165], Young et al. [166]), question answering system (Talmor et al. [167]), mathematical problem generation (Liu et al. [168]) etc. Chakrabarty et al. [1] utilized commonsense knowledge generation for their sarcasm generation task to incorporate additional commonsense context to their sarcastic sentences.

2.0.7 Emoji Based Sarcasm Detection

Besides sarcasm detection on only textual data, many works have incorporated emojis with the textual data to see if emojis help in better assessing the sarcastic notion of a sentence or not. Some of the notable emoji based sarcasm detection work include Chaudhary et al. [169], Subramanian et al. [7], Chauhan et al. [170]. Among them, Subramanian et al. [7] proposed a framework named ESD (Emoji based Sarcasm Detection) which is trained on Facebook and Twitter data incorporated with various emojis and showed that the model outperformed the then state of the arts FSNN, CASCADE in Accuracy, prcision, recall and F1-socres. Their performance comparison with the other textual models are shown in Figure 2.2. This proves that incorporating emoji information actually helps to better detect sarcasm.

Datasets	Metric	FSNN	CASCADE	RCCSD	ESD
Twitter	Accuracy	0.891	0.753	0.763	0.991
	Precision	0.910	0.798	0.768	0.998
	Recall	0.904	0.802	0.791	0.976
	F1	0.899	0.867	0.820	0.987
Facebook	Accuracy	0.878	0.745	0.768	0.971
	Precision	0.901	0.771	0.733	0.975
	Recall	0.889	0.789	0.745	0.979
	F1	0.893	0.842	0.772	0.969

FIGURE 2.2: Performance comparison between textual sarcasm detection models and emoji based textual sarcasm detection model

2.0.8 Emoji Prediction

Emoji Prediction can be considered as a multi-class classification problem where given an input text the model will predict an emoji as the class the text belongs to. Emoji prediction of tweets is an emerging problem Barbieri et al. [171] which combines the nuances of sentiment analysis with the noisy data characteristic of social media. Barbieri et al. [171] investigated the relationship between words and emojis, studying the task of predicting which emojis are evoked by text-based tweet messages. The authors trained several models based on Long Memory Short-Term networks (LSTMs). This marked the beginning of the task of emoji prediction.

As a part of SemEval 2018 Task 2[24], a Multilingual Emoji Prediction Task was organized, with two subtasks proposed, one for English and one for Spanish, in which teams were given a text as input and had to create models that predicted an emoji based solely on the textual content of that message. Çöltekin and Rama [172] ranked first by means of an SVM classifier. In their feature extraction procedure, they took into account both the character level and word level in order to extract the bag-of-n-grams features. Baziotis et al. [59] obtained 2nd place in the competition for the English dataset with their

[24] `https://competitions.codalab.org/competitions/17344`

context-aware word-level BiLSTM architecture with an attention mechanism that outperformed the then state-of-the-art (Barbieri et al. [171]). Other notable works include - Coster et al. [173] with their best performing linear SVM using the SGDClassifier model, Groot et al. [174] showed that their SVM model performed better than their LSTM and ensemble model. Chen et al. [175] proposed a vector similarity-based approach for this task where the similarity between the tweet vector and each emoji's embedding is evaluated. The most similar emoji is chosen as the predicted label. This similarity-based approach performed better than the classification approach on the evaluation set but performed worse in the test set due to having many unseen words and different class distributions.

Some works have considered emoji prediction task as a multi-label prediction problem, where multiple emojis are predicted for an input text. Guibon et al. [176] utilized CBOW embeddings to obtain 18 specific clusters of emojis that represent similar emotions, they used these clusters to further recommend multiple emojis to users while texting. Peng and Zhao [177] regarded multi-emoji prediction as a sequence generation task to better learn the correlation between emojis with their hierarchical structured BiLSTM-CNN-RNN model named Seq2Emoji. Barbieri et al. [178] compared emoji embeddings trained on a corpus of different seasons and show that some emojis are used differently depending on the time of the year and that using the time information, the accuracy of some emojis can be significantly improved. The recent advances in emoji prediction are extended even further by Barbieri et al. [179] with their multimodal approach to predict emojis from Instagram posts which include an image and a textual caption related to that image. They used ResNet and FastText for their model and showed that using a compound model that includes the two synergistic modalities increases accuracy in an emoji prediction task. Basile and Lino [180] experimented using three different approaches: using bag-of-words with Naive Bayes and SVM, using word embeddings and a deep neural classifier, and modeling as a translation problem using a state-of-the-art neural translation system to predict the labels as translated sentences. They concluded that neural models can give good results but hyper-parameter tuning is a hard task and if it is not successful, then a good linear classifier with a bag-of-words representation can easily outperform the neural model. Alexa et al. [181] implemented two main modules: a Recurrent Neural Network and a Naïve Bayes algorithm. The results for the Naïve Bayes implementation were better than those from the network module. Wu et al. [40] proposed a residual CNN-LSTM with attention (RCLA) model to capture both local and long-range contextual information and also incorporated additional features such as POS tags and sentiment features. This model achieved a 30.25% macro-averaged F-score in the emoji prediction task at SemEval-2018. Çöltekin and Rama [172] experimented with SVMs and recurrent neural networks and the SVM classifier outperformed every other team at the SemEval-2018 Task 2 with a macro-averaged F1-measures of 35.99% for English data sets. Their experiments showed that linear models, particularly SVMs, yield better results than (deep) neural models. Kopev et al. [182] also achieved the best results using an SVM-based classifier. They also incorporated a Hierarchical Attention Neural Network. Wang and Pedersen [183] developed a Multi-channel Convolutional Neural Network based on subword embeddings which improve character embedding by 2.1% and word embedding by 1.8%. Beaulieu and Owusu [184] used a Bag of Words model with a Linear SVM classifier and achieved a macro F1 score of 32.73% in the emoji prediction task at SemEval-2018. Wang et al. [185] utilized a bi-directional gated recurrent unit with an attention mechanism to build their base model, trained and multi-models with or without class weights for the ensemble methods. This method demonstrated an improvement of approximately 3% of the macro F1 score at SemEval-2018 Task 2. Barbieri et al. [171] investigated the relationship between words and emojis, studying the novel task of predicting which emojis are evoked by text-based tweet messages. They employed a state-of-the-art classification framework based on Bidirectional Long Short-term Memory Networks (BLSTMs), and showed that it outperforms a bag of words baseline, a baseline based on semantic vectors, and human annotators. Tomihira et al. [186] verified and compared multiple models based on RNN and CNN that learn from sentences using emojis as labels, collecting Japanese tweets from Twitter as the corpus. Wu et al. [187] proposed to predict multi-label emoji prediction in tweets using a hierarchical neural model with an attention mechanism. Their model contained a character encoder to learn hidden representations of words from original characters using a CNN layer.

Chapter 3

Proposed Methodology

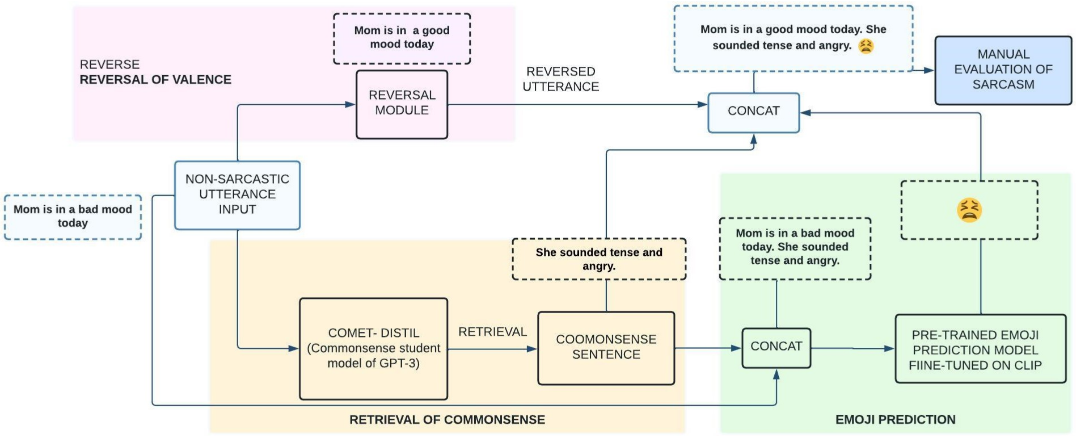
REVERSE
REVERSAL OF VALENCE
Mom is in a good mood today
REVERSAL MODULE
REVERSED UTTERANCE
Mom is in a good mood today. She sounded tense and angry. 😫
CONCAT
MANUAL EVALUATION OF SARCASM
NON-SARCASTIC UTTERANCE INPUT
Mom is in a bad mood today
😫
She sounded tense and angry.
Mom is in a bad mood today. She sounded tense and angry.
COMET- DISTIL (Commonsense student model of GPT-3)
RETRIEVAL
COOMONSENSE SENTENCE
CONCAT
PRE-TRAINED EMOJI PREDICTION MODEL FIINE-TUNED ON CLIP
RETRIEVAL OF COMMONSENSE
EMOJI PREDICTION

FIGURE 3.1: Proposed Architecture

Burgers et al. [20] proposed that for an utterance to be sarcastic, it has to at least have 5 of these characteristics- 1) be evaluative; 2) be based on a reversal of valence between the literal and intended meaning; 3) be based on a semantic incongruity with the context, which can include shared commonsense or world knowledge between the speaker and the addressee; 4) be aimed at some target, and 5) be relevant to the communicative situation in some way. We will be considering Chakrabarty et al. [1] as our baseline for the sarcasm generation task which works with two of these sarcasm factors- reversal of valence and semantic incongruity.

As discussed in the Literature Review section 2.0.5, their architecture is composed of three modules - 1. Reversal of Valence, 2. Retrieval of Commonsense Context, and 3. Ranking of Semantic Incongruity. For the reversal module, the evaluative word is identified and negation or replacement with lexical antonyms is performed. This module fulfills the reversal of valence factor where the literal meaning of the utterance is the reverse of the intended meaning. For example- for the input “I inherited unfavorable genes from my mother”, this module will give the output “I inherited great genes from my mother” which is the reverse of what was intended. But only doing the reverse is not always enough to express sarcasm, we need context with the reverse text, that will give us some idea about the sarcastic situation.

The second module utilized the fourth factor- semantic incongruity with the context where COMET Bosselut et al. [80] is used to generate relevant commonsense knowledge which is concatenated with the reversed sentence achieved from the reverse module to give some context. For example, for the previous input, “I inherited unfavorable genes from my mother”, COMET will produce “Ugly goes down to the bone.” So the total output will be, “I inherited great genes from my mother. Ugly goes down to the bone.” which gives a sarcastic notion of how the speaker inherited bad genes from their mother. Finally, in the last module, the semantic incongruity of the retrieved outputs after the second module is calculated and ranked using a fine-tuned RoBERTa-large Liu et al. [109] on Multi-NLI dataset Williams et al. [188].

Just like them, our architecture will include a deep learning Reversal of Valence module which will take in a negative utterance and give a positive utterance as an output. It will also have a Retrieval of Common sense module which will output additional common sense context which will be incongruous with the reversed output from the previous reversal module. We concatenate the reverses sentence and the common sense context together and incorporate an emoji with this output where the emoji will provide additional context which further elevates the sarcastic situation. For emoji prediction, we use a pre-trained emoji prediction model which is fine tuned on the CLIP (Radford et al. [8]) deep learning model by OpenAI to predict an emoji from a given input which is fine-tuned on a dataset containing a set of 32 emojis. So, our emoji prediction model will be a 32-class classification problem.

Our proposed methodology mainly includes three modules, 1) Reversal of Valence module, 2. Retrieval of Commonsense Context and 3) Emoji Prediction.

3.0.1 Reversal of Valence

In the work of Chakrabarty et al. [1], for the reversal of valence module, they have used a rule-based approach to manually reverse the sentiment of the negative sentence. But a rule-based model cannot reverse sentences that do not follow the traditional structure of sentences such as those used in social media. We can see some of the examples of negative inputs in table 3.1 which went through the Chakrabarty et al. [1]'s reverse module which is said to produce outputs with reversed sentiment, here for negative outputs it should produce positive sentiment sentences. As we can see in table 3.1, the rule-based model's simple rules cannot process the complex input sentences.

negative input	**reverse output by Chakrabarty et al. [1]**
Wishing i could watch another cullen family baseball game but dont think itll happen again sigh looking cast again.	Wishing i could watch another cullen family baseball game but dont don't think itll happen again sigh looking cast again.
Home with the flu.	Home with the not flu.
If i cant post an episode today i might as well sleep now so annoyed with my stupid computer sorry guys.	If i don't cant post an episode today i might as well sleep now so annoyed with my stupid computer sorry guys.

Table 3.1: Example of reversal module outputs of Chakrabarty et al. [1]

We have worked on this limitation of this current state-of-the-art sarcasm generation model where we replace their rule-based reversal module with a deep-learning reversal module inspired by the work of Mishra et al. [3]. We decided to use a deep learning model because deep learning models will not just solely focus on the input sentence's syntax or structure but also will try to assess the whole sentiment of the sentence in order to meaningfully reverse the negative sentences into positive ones. Though Mishra et al. [3] performs worse than Chakrabarty et al. [1], but they proposed a clever methodology to reverse the sentiment of the sentence. We utilize this idea to create a modified version of the Mishra et al. [3]'s reversal mechanism in our own work to see if this module works well with the other modules of the framework or not.

This module is divided into two parts: Sentiment Neutralization and Positive Sentiment Induction. The module's architecture is shown in figure 3.2. Sentiment Neutralization module tries to remove sentiment indicative words from the sentence to make it a neutral sentence and Positive Sentiment Induction Module tries to incorporate positive sentiment words into a neutral sentence input to turn into a positive sentence.

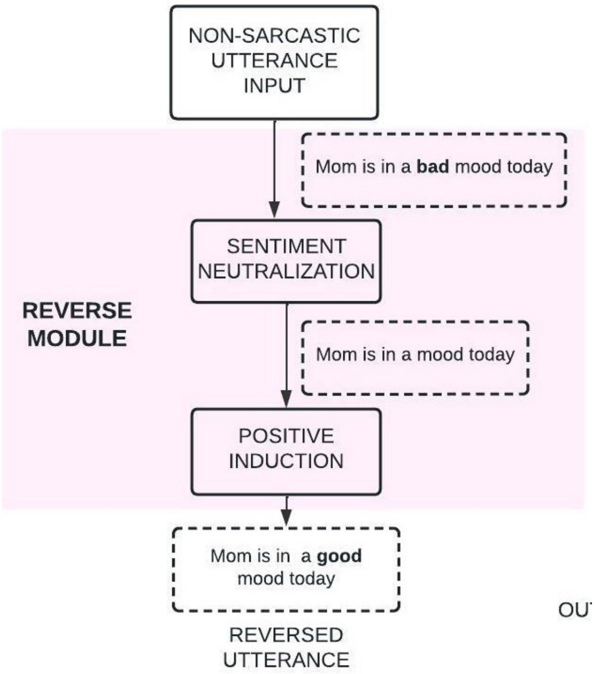

FIGURE 3.2: Reversal of Valence module Architecture

3.0.1.1 Sentiment Neutralization

We implement the Sentiment Neutralization module to filter out the sentiment words from the input utterance, which results into a neutral sentence from a negative one. An example is shown in table 3.2. The neutralization model is essentially a sentiment classification model which first detects the sentiment of the given utterance (positive/negative). This model consists of several LSTM layers and a self-attention layer. During testing, the self-attention vector is extracted as done by Xu et al. [189] which is then inversed and discretized as follows:

$$\hat{a_i} = \begin{cases} 0, & \text{if } a_i > 0.95 * max(a) \\ 1, & \text{otherwise} \end{cases} \tag{3.1}$$

where a_i is the attention weight for the i^{th} word, and $max(a)$ gives the highest attention value from the current utterance. A word is filtered out if the discretized attention weight for that word is 0. The sentiment detection model architecture is shown in figure 3.3.

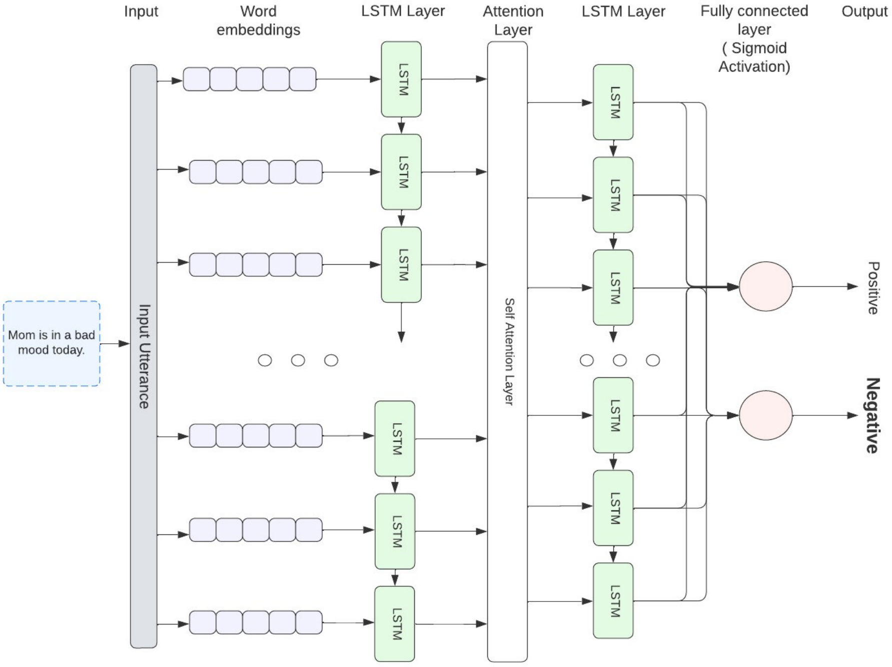

FIGURE 3.3: Sentiment detection model architecture for the Sentiment neutralization module

negative input	**neutral output**
is feeling absolutely bloated and fat from lack of a proper workout	is feeling absolutely and from a proper workout

TABLE 3.2: Example of sentiment neutralization from input sentence

Mishra et al. [3] used mean and variance instead of max to filter out the words where a word is filtered out if the discretized attention vector value for that is between the mean and variance value for that whole sentence. This process was filtering out most of the words from the sentences for our dataset. So we decided to only filter the words which have the highest attention values in the sentence. That is why we used the max function instead.

3.0.1.2 Positive Sentiment Induction

The output from the Sentiment Neutralization module is fed into the Positive Induction module as input. The module takes in a neutral utterance and incorporates positive sentiment into the utterance and returns a sentence with positive sentiment. An example is shown in table 3.3. For this, we use Neural Machine Translation method built on OpenNMT framework Klein et al. [190] where we first train our model with a set of $< source, target >$ pairs where the source is a neutral sentence and target is its positive counter part. We use the Positive dataset provided by Mishra et al. [3] which includes a set of positive sentences. We pass this dataset through the sentiment neutralization module to get the neutral source sentence to its positive target sentence and use these $< source, target >$ pairs to train the positive induction module. The input sentences are transformed into embeddings that go through the translation encoders and decoders. The encoders and decoders are both built with LSTM layers. The model architecture is shown in figure 3.4.

neutral input	**positive output**
is feeling absolutely and from a proper workout	is feeling absolutely amazing and high got away from a proper workout

Table 3.3: Example of positive sentiment induction from neutralized sentence

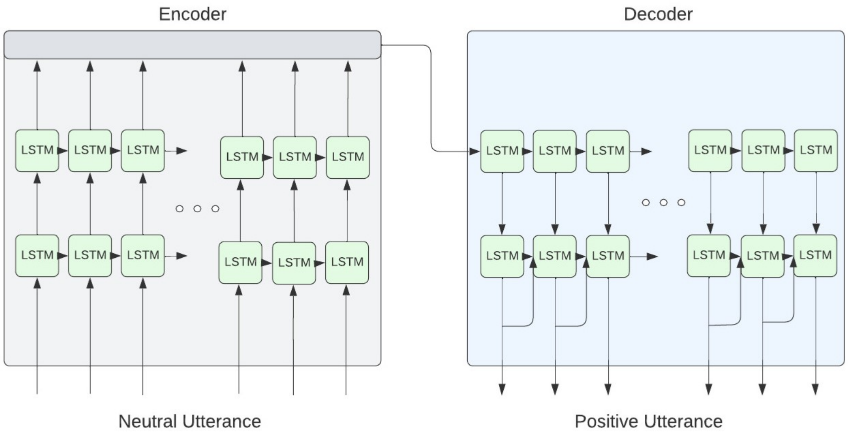

Figure 3.4: Model Architecture for Positive Induction module

3.0.2 Retrieval of Commonsense

This module is used to retrieve additional context for the sarcastic sentence based on commonsense knowledge. Figure 3.5 demonstrates a schematic view of this module. In order to generate additional context, firstly, a phrase regarding commonsense knowledge in accordance to the non-sarcastic input is generated using $\text{COMET}^{\text{DIS}}_{\text{TIL}}$. Secondly, to get an actual sentence from the retrieved phrase, 2 methods are applied, where one method searches and fetches sentences containing the keyword of the phrase from a corpus and the other method includes generating a commonsense sentence from the generated commonsense phrase using a language generation model. Finally, the semantic incongruities of the retrieved sentences with the reversed sentence that we got from the Reversal of Valence module are calculated. The commonsense sentence having the most incongruity is selected. We discuss the detailed process in the following sections. Additionally, we show an example input-output pair for this module in table 3.4.

input	**commonsense sentence**
his presentation was bad	the manager is criticized by his boss after a presentation

Table 3.4: Example of commonsense sentence generation from input sentence

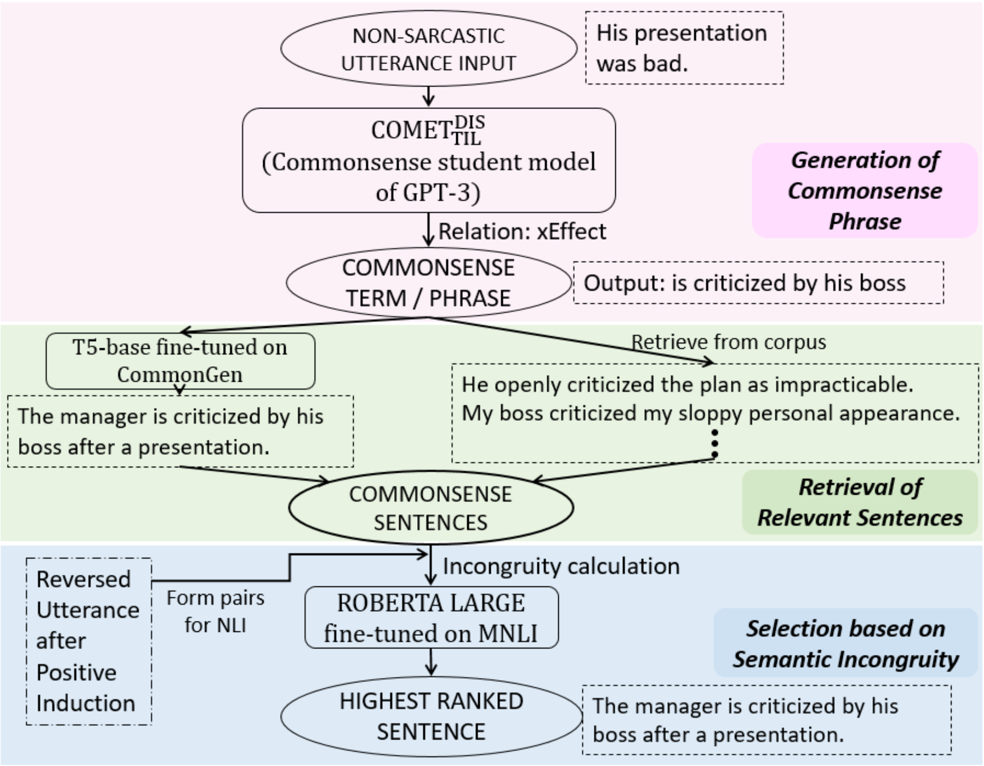
NON-SARCASTIC UTTERANCE INPUT
His presentation was bad.
$\text{COMET}^{\text{DIS}}_{\text{TIL}}$ (Commonsense student model of GPT-3)
Generation of Commonsense Phrase
Relation: xEffect
COMMONSENSE TERM / PHRASE
Output: is criticized by his boss
T5-base fine-tuned on CommonGen
Retrieve from corpus
He openly criticized the plan as impracticable.
My boss criticized my sloppy personal appearance.
The manager is criticized by his boss after a presentation.
COMMONSENSE SENTENCES
Retrieval of Relevant Sentences
Reversed Utterance after Positive Induction
Form pairs for NLI
Incongruity calculation
ROBERTA LARGE fine-tuned on MNLI
Selection based on Semantic Incongruity
HIGHEST RANKED SENTENCE
The manager is criticized by his boss after a presentation.

Figure 3.5: Model Architecture for Retrieval of Commonsense module

3.0.2.1 Generation of Commonsense Knowledge

For generating commonsense knowledge context, $\text{COMET}^{\text{DIS}}_{\text{TIL}}$ (West et al. [156]) is used. First, we feed the input sentence to $\text{COMET}^{\text{DIS}}_{\text{TIL}}$. $\text{COMET}^{\text{DIS}}_{\text{TIL}}$ is a machine trained 1.5B parameters commonsense model generated by applying knowledge distillation Hinton et al. [155] on a general language model, GPT-3. Among the 23 relation types, West et al. [156] evaluated COMET$^{\text{DIS}}_{\text{TIL}}$ on 7 relation types that correspond to *casual* commonsense knowledge: **xAttr** (how X is perceived after *event*), **xReact** (how X reacts in response to *event*), **xEffect** (what X does after *event*), **xIntent** (X's intent in *event*), **xWant** (what X wants after *event*), **xNeed** (what X needed for *event* to take place) and **xHinderedBy** (what might hinder the *event*). For our study, we have used the **xEffect** relation. From the three variants of COMET$^{\text{DIS}}_{\text{TIL}}$ (COMET$^{\text{DIS}}_{\text{TIL}}$, COMET$^{\text{DIS}}_{\text{TIL}}$ + critic$_{\text{low}}$ and COMET$^{\text{DIS}}_{\text{TIL}}$ + critic$_{\text{high}}$), we have chosen COMET$^{\text{DIS}}_{\text{TIL}}$ + critic$_{\text{high}}$ for our work. The model returns a contextual phrase pertaining to the **xEffect** relation with the extracted words of the non-sarcastic sentence. For a non-sarcastic sentence "His presentation was bad", COMET$^{\text{DIS}}_{\text{TIL}}$ predicts the contextual phrase with **xEffect** relation – 'is criticized by his boss'.

3.0.2.2 Retrieval of Relevant Sentences

Once we have the inferred contextual phrase, we retrieve relevant sentences. For doing so, we imply 2 methods - 1. Retrieval from corpus and 2. Generation from the inferred phrase.

- **Retrieval from corpus:** First, from the contextual phrase, we extract the keyword. Then using the keyword, we search for related sentences in a corpus. We use Sentencedict.com [1] as the retrieval corpus. For filtering the retrieved sentences, two constraints are set - (a) the commonsense concept should appear at the beginning or at the end of the retrieved sentences; (b) to maintain consistency between the length of the non-sarcastic input and its sarcastic variant, sentence length should be less than twice the number of tokens in the non-sarcastic input. Next, we check the consistency of the pronoun in the retrieved sentence and the pronoun in the input sentence. If the pronoun does not match, we modify it to match the non-sarcastic text input. If the non-sarcastic input lacks a pronoun while the retrieved sentence does not, it is simply changed to "I". These constraints for retrieving the sentences and the assessment of grammatical consistency are done following the work of Chakrabarty et al. [1].
- **Generation from the inferred phrase:** Unlike the previous method, we keep the inferred phrase intact in this case. We first extract the *Subject* of the non-sarcastic input. If the sentence contains no *Subject*, we set it to 'I'. Then the auxiliary verb in the inferred context is checked and modified to match with that of the *Subject*. Then we feed the *Subject* and contextual phrase to a pre-trained

[1] `https://sentencedict.com/`

sentence generation model[2]. The model fine-tunes Google's T5 on CommonGen (Lin et al. [191]). The model returns us a commonsense sentence based on the *Subject* and contextual inference. For example - the *Subject-inference* pair for the input "His presentation was bad" becomes ['His', 'is criticized by his boss'], and from this collection of words, the sentence "The manager is criticized by his boss after a presentation." is generated.

3.0.2.3 Selection based on Semantic Incongruity

Section 3.0.2.2 returns several sentences containing the context. Among them, we choose the sentence having the highest semantic incongruity with the sentence generated after the Reversal of Valence module. For calculating the semantic incongruity, following Chakrabarty et al. [1], we have used the RoBERTa-large (Liu et al. [109]) model fine-tuned on the Multi-Genre NLI dataset (Williams et al. [188]). Considering the non-sarcastic input "His presentation was bad", section 3.0.2.2 yields a list of sentences such as - "The manager is criticized by his boss after a presentation", He openly criticized the plan as impracticable", and "My boss criticized my sloppy personal appearance". From these sentences, the highest ranked sentence, "The manager is criticized by his boss after a presentation", is returned as the final output to this module as it contains the most semantic incongruity with the reversed sentence.

3.0.3 Emoji Prediction

In this module, we use a pre-trained emoji prediction model which is fine tuned on the CLIP Radford et al. [8] deep learning model by OpenAI to predict an emoji from a given input. After concatenating the non-sarcastic input and the context retrieved from the Retrieval of Commonsense module, we predict an emoji based on this concatenated sentence. The module architecture is shown in figure 3.6. The model employs a masked self-attention Transformer as a text encoder and a ViT-B/32 Transformer architecture as an image encoder. By using a contrastive loss, these encoders are trained to optimize the similarity of (image, text) pairs. One version of the implementation used a Vision Transformer and the other a ResNet image encoder. The variation with the Vision Transformer is used in this case. The dataset[3] used for fine-tuning the model consists of two columns: raw tweets and emoji labels. The emoji labels correspond to the appropriate one among a set of 32 emojis shown in figure 3.7.

[2] https://huggingface.co/mrm8488/t5-base-finetuned-common_gen

[3] https://huggingface.co/datasets/vincentclaes/emoji-predictor

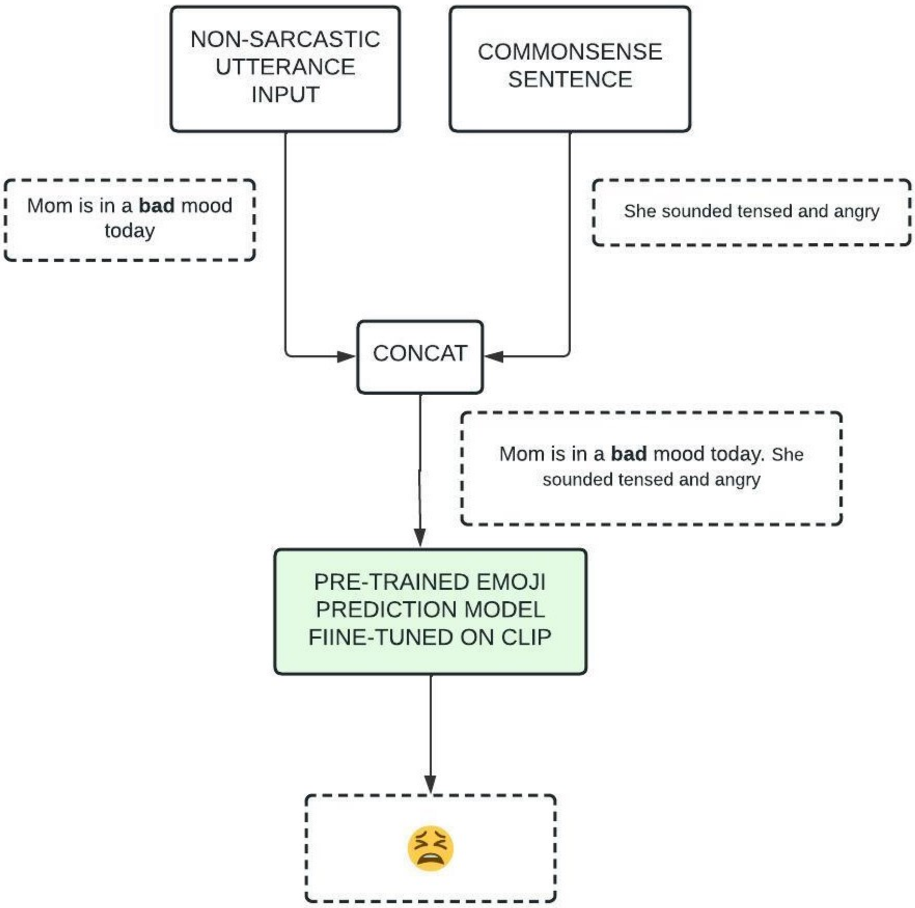
NON-SARCASTIC UTTERANCE INPUT
COMMONSENSE SENTENCE
Mom is in a **bad** mood today
She sounded tensed and angry
CONCAT
Mom is in a **bad** mood today. She sounded tensed and angry
PRE-TRAINED EMOJI PREDICTION MODEL FIINE-TUNED ON CLIP
😫

FIGURE 3.6: Emoji Prediction module Architecture

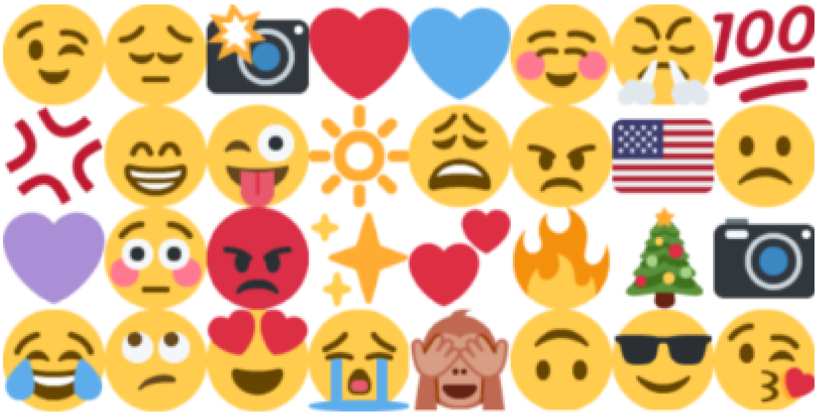

FIGURE 3.7: Set of 32 emojis

Chapter 4

Experimental Setup

The dataset, model configurations for the different modules, and the evaluation criteria for our work are all discussed in the following sub sections.

4.0.1 Dataset

For our experiments, we utilize the Positive and Negative sentiment corpora by Mishra et al. [3]. These two corpora consist of short sentences or snippets and tweets from four sources which are: Stanford Sentiment Treebank Dataset by Socher et al. [192], Amazon Product Reviews[1], Yelp Reviews[2] and Sentiment 140 dataset by Kotzias et al. [193]. Tweets have been normalized by eliminating hashtags, usernames, and conducting spell checking and lexical normalization using NLTK (Loper and Bird [194]). Each set contained about 47827 samples each where the Positive corpora consists of sentences with positive sentence and Negative corpora consists of negative emotion sentences.

To train our sentiment detection model, we utilized both the corpora and labeled each sentence 1 for negative sentiment and 0 for positive sentiment. Then we concatenate the two corpora and train our sentiment model with these samples. For the positive induction module, after filtering out sentences longer than 30 words these sentences went through the sentiment neutralizer module. From the output of this module, we filtered out the neutralized sentences whose length were less than 50% of the actual input length. After the filtering process, approximately 27380 samples remained from the positive corpora with its neutral counterpart. We train our Positive Induction module with these 27380 $< neutralutterance, positiveutterance >$ sample pairs.

[1] https://www.amazon.com/

[2] https://www.yelp.com/

Next, we took the negative corpora which went through the sentiment neutralizer module and again we filtered out the sentences whose length were 80% less from the original input sentence. These left us with exactly 8399 samples. These samples then went through the Positive Induction module and finally we got our desired reversed sentences.

4.0.1.1 Data Annotation & Filtering

Due to time shortage we randomly took 6000 samples from here and upon manual filtering and annotating, 2,000 sarcastic sentences are picked as the sarcastic dataset. To get the 2,000 samples out of the 6000 samples returned by the model, 11 annotators labeled the samples as either sarcastic or non-sarcastic. These 2,000 samples finally make up our sarcastic dataset. After much consideration, 11 annotators were chosen based on their efficiency over English language, clear idea about what sarcasm is, how to distinguish between a sarcastic sentence and a normal sentence and lastly the ability to recognize creativity and humor in sentences. Next, the annotators were provided with a set of instructions in the form of a manual where we explained our research work and what we are trying to achieve. They were thoroughly guided on how to recognize sarcasm in sentences via written guidelines and meetings. Then each of them were provided with a previously agreed upon number of samples which included $< negativeinpututterance, generatedoutputfromourmodel >$ pairs and were asked to annotate if the output sentence seemed sarcastic or not. If the output sentence seemed to grammatically not make sense at all, those samples were filtered out by the annotators.

4.0.2 Model Configuration

The sentiment model of the neutralization module is trained on the sentiment dataset given by Mishra et al. [3] where the negative sentences are labeled as 1 and the positive sentences are labeled as 0. Each word in the input sentence is first encoded with one-hot encoding and turned into a K-dimensional embedding. Then, these embeddings go through an LSTM layer with 200 hidden units, a self-attention layer, another LSTM layer with 150 hidden units and finally a softmax layer. The classifier is trained for 10 epochs with a batch size of 32, and achieves a validation accuracy of 96% and a test accuracy of 95.7%.

Mishra et al. [3] used an older version of the OpenNMT framework where some of the functions were deprecated, so we had to retrain the model with our own processed data on a newer OpenNMT framework. We used a smaller set of samples than them as with our length criteria a lot of sentences were filtered out. The positive sentiment induction module is built on top of the OpenNMT 3.0 framework, and following Mishra et al. [3], the embedding dimensions of the encoder and decoder is set to 500, with 2 LSTM layers each consisted of 500 hidden units. Training iteration is set to 100000 and early stopping is incorporated to prevent overfitting. After training, the model produced a corpus-BLEU score of 51.3%.

4.0.3 Environmental Setup

We run our experiments on several environmental setups. For our sentiment detection model training from section 3.0.1.1, Emoji prediction task from section 3.0.3 and Positive Induction model training from section 3.0.1.2, we used the Google Colaboratory[3] Notebook enviorment with T4 type GPU enabled. For generating positive sentiment sentence from the Positive induction module, we used our own setup with a AMD Ryzen 3700x 8-core Processor,32 GB Ram, AMD Radeon RX 5700 XT PC where we run the experiment in the Anaconda 4.12.0[4]. For generating Common sense context and ranking them, we used two setups, one with a AMD Ryzen 3700x 8-core Processor,32 GB Ram, AMD Radeon RX 5700 XT PC where we run the experiment in the Anaconda 4.12.0 and other with an Intel core i9 12900k, 64 GB Ram, RTX 3090 x1 or x2, Z690 series MOBO pc also in Anaconda environment. As there was a lot of output samples, we used the two systems to distribute the task evenly.

4.0.4 Evaluation Criteria

For evaluating the performance of our proposed architecture we incorporate Human judgement. To assess the quality of the generated dataset we compare among four systems.

1. **Full Model** The Full Model contains all three modules of the proposed framework and generates the final dataset.
2. **Without Emoji** The Without Emoji system includes the context sentences along with the outputs from the reversal of valence module but does not contain any emoji that goes with each sarcastic sentence in the final dataset.
3. **Without Context** The Without Context system consists of generated sentences from the reversal of valence module as well as the associated emoji for the utterance. However, It does not include any context.

[3] `https://colab.research.google.com/`

[4] `https://www.anaconda.com/`

4. **R**^{3} The R^3 system is the state-of-the-art sarcasm generation model proposed by Chakrabarty et al. [1].

For comparing on the basis of the four above mentioned systems, we evaluate 400 generated sentences in total. From the 2,000 sarcastic data, 100 samples are chosen randomly. Each system is assessed on these 100 randomly chosen utterances. Following the evaluation approach proposed by Chakrabarty et al. [1] in their work, we evaluate the generated sentences on these criteria:

1. Sarcasticness ("How sarcastic is the output?"),
2. Creativity ("How creative is the output?"),
3. Humour ("How funny is the output?"),
4. Grammaticality ("How grammatically correct is the output?").

Three human judges have been chosen to rate the outputs from the four systems on the four criteria mentioned. The rating is done on a scale of 5 where 1 indicates not at all and 5 indicates very. All of the three judges rate each of the 400 sentences from the 4 systems. The human judges have been chosen based on their high efficiency in English, good grasp in understanding and differentiating between Creativity, Humor and Sarcasticness in English sentences.

4.0.4.1 Criteria Selection Justification

Though BLEU score is used for evaluating generation tasks, but as sarcasm is even hard for humans to understand and can come in different formats and situations, simply a n-gram overlap between human generated output and machine translated output is not enough for the evaluation. For this purpose, just like Chakrabarty et al. [1], our human evaluators evaluated each sentence based on their sarcasticness, creativity, humor and correct grammar. Sarcasm is often associated with intelligence, creativity, and a quick wit. It requires cognitive skills to understand and manipulate language in unconventional ways, allowing for clever critiques and humorous wordplay. That is why, Chakrabarty et al. [1] proposed these 4 criteria to evaluate a sarcastic sentence. To see how our model is performing in all those criteria compared to the baseline, we have also incorporated these evaluations in our study.

Chapter 5

Results and Discussions

Table 5.1 shows the comparison among a few sample sarcastic outputs across the four systems which are our full model, output without the context but with emoji, output without any emoji but with the context and lastly the state-of-the-art model by Chakrabarty et al. [1]. The comparisons are on four different measures mentioned earlier such as Sarcasticness, Creativity, Humor and Grammaticality. Each score in the table is the average rating given by the three human judges for each sample. Table 5.2 shows the average ratings on 100 randomly chosen samples by the human judges for generated sarcastic sentences from the four systems based on the four categories. Figure 5.1, 5.2, 5.3 and 5.4 show us the score comparison through the 100 samples between the 4 different criteria (Sarcasticness, Humor, Creativity and Grammaticality) between our proposed framework and our baseline model, Chakrabarty et al. [1].

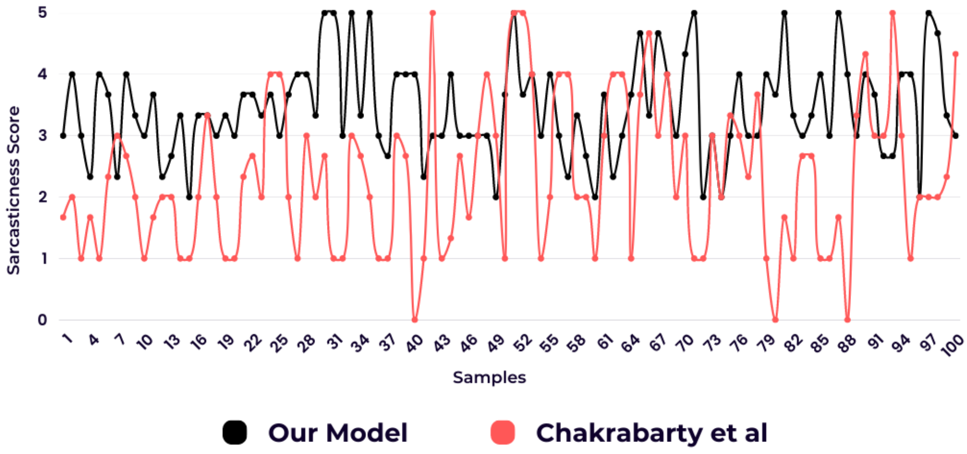

Figure 5.1: Sarcasticness score comparison for 100 samples between our full model and Chakrabarty et al. [1]

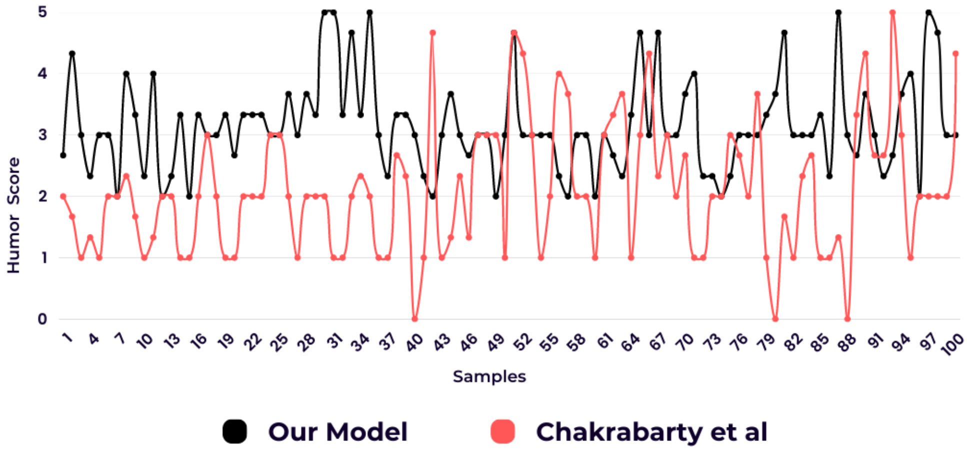

Figure 5.2: Humor score comparison for 100 samples between our full model and Chakrabarty et al. [1]

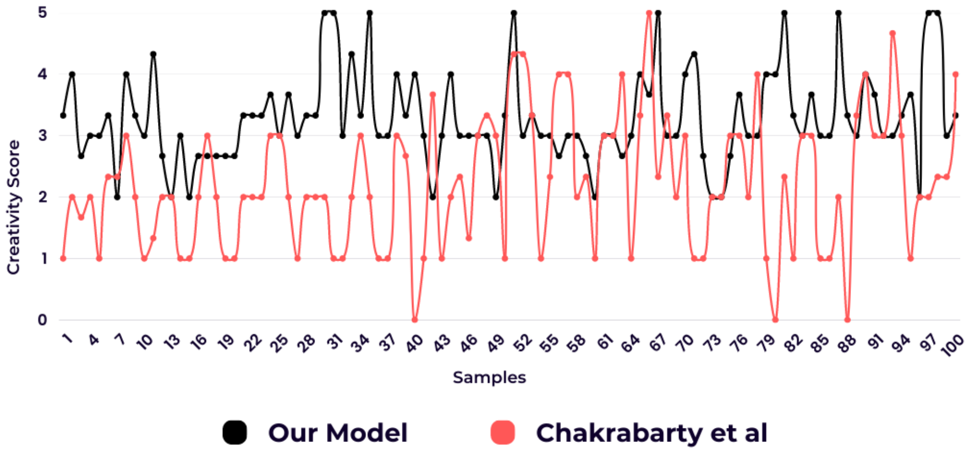

FIGURE 5.3: Creativity score comparison for 100 samples between our full model and Chakrabarty et al. [1]

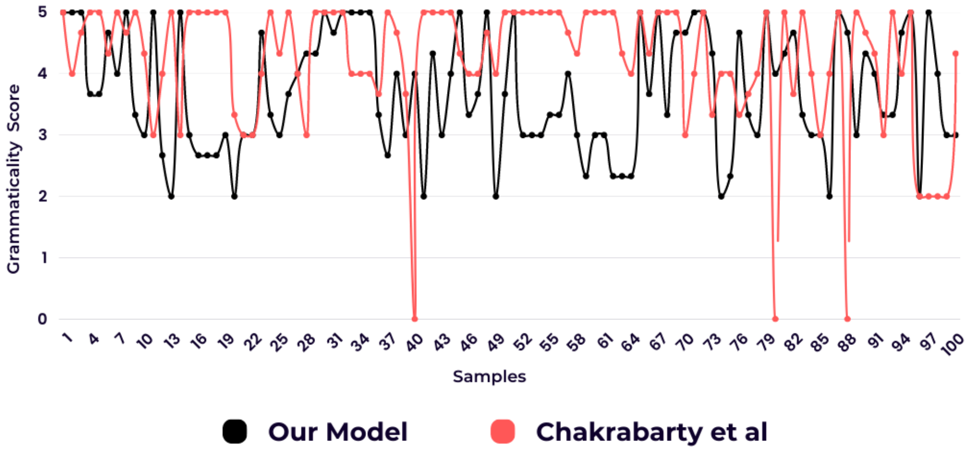

FIGURE 5.4: Grammaticality score comparison for 100 samples between our full model and Chakrabarty et al. [1]

5.0.1 Quantitative Analysis

From table 5.1, we can see that our proposed model got the highest score in Sarcasticness (3.29), Creativity (3.44) and Humor (3.16) among all the other models. The next best performing model in Sarcasticness is the Without Context model where it has only the reversed sentence concatenated with the emoji and no context. The next best performing model in Sarcasticness is the Without Emoji model (reversed sentence + context), but it comes ahead in Creativity and Humor of the Without Emoji model. This is because adding common sense context adds more creativity and humor to the output than simply just adding an emoji. The reason why Without Context gives more Sarcastic outputs than Without Emoji because sometimes the common sense cannot retrieve relevant context sentences, so the emoji makes up for the additonal context. The lowest performing in Sarcasticness (2.2), Creativity (2.32) and humor (2.1) is the state-of-the-art baseline model, but it outputs the most grammatically correct sentences (Grammaticality score of 4.29 against our model's grammaticality score of 3.72) due to Chakrabarty et al. [1] only making rule-based changes to the grammatically correct non-sarcastic sentence and fetching grammatically correct context sentences from a retrieval corpus. From the Figures 5.1, 5.2, 5.3 and 5.4, we can see the difference between the scores of our full model and Chakrabarty et al. [1]. Our model constantly got better results than the baseline model in most of the samples. We can also notice that the baseline model scores are fluctuating a lot as it performs poorly for the sentences with complex structures.

Non-Sarcastic Utterance	**System**	**Sarcastic Utterance**	**Sarcasticness**	**Creativity**	**Humor**	**Grammaticality**
Home with the flu.	Full Model	Happy to be home with the fam. Being incarcerated-under the label of being mentally ill.😫	3.67	4.33	4	5
	Without Emoji	Happy to be home with the fam. Being incarcerated-under the label of being mentally ill.	3.67	4.33	3.67	5
	Without Context	Happy to be home with the fam.😫	3.33	3	3	5
	R^3 (Chakrabarty et al. [1])	Home with the not flu.	1.67	1.33	1.33	3
The boss just came and took the mac away.	Full Model	The boss just ended and took the mac away awesome. Angry is not the word for it - I was furious.😠	5	5	4.67	4.33
	Without Emoji	The boss just ended and took the mac away awesome. Angry is not the word for it - I was furious.	4	3.67	3	4.67
	Without Context	The boss just ended and took the mac away awesome.😠	5	5	4.67	4.33

	R^3 (Chakrabarty et al. [1])	The boss just came and took the mac away. Angry is not the word for it - I was furious.	1.67	2.33	1.67	5
Friday nights are so boring when the boyfriend is working late and then i have to work at on saturday mornings.	Full Model	Friday nights are so cute when the boyfriend is working rearrange and then i have to work at on mornings. At least they weren't bored. 😔	4	4	3.67	4
	Without Emoji	Friday nights are so cute when the boyfriend is working rearrange and then i have to work at on mornings. At least they weren't bored.	4	4	3.67	4
	Without Context	Friday nights are so cute when the boyfriend is working rearrange and then i have to work at on mornings. 😔	4	4	3.67	4
	R^3 (Chakrabarty et al. [1])	Friday nights are so boring when the boyfriend is working early and then I have to work at on saturday mornings. Friday saw the latest addition to darlington's throbbing night life packed to the rafters.	1.33	2	1.33	5

Just finished workin bed feeling sick.	Full Model	Just finished workin feeling good. My stomach heaved and I felt sick.😫	5	5	4.67	5
	Without Emoji	Just finished workin feeling good. My stomach heaved and I felt sick.	5	5	4.67	5
	Without Context	Just finished workin feeling good.😫	3	3	3	5
	R^3 (Chakrabarty et al. [1])	Just finished workin bed feeling healthy. My stomach heaved and I felt sick.	5	4.33	4.67	5

TABLE 5.1: Score comparison among the generated outputs from the different systems (Full model, Output without context, Output without emoji and the State-of-the-art model) on four categories

5.0.2 Qualitative Analysis

As we have seen, our full model achieves the highest average score among all the systems including the state-of-the-art sarcasm generation model by Chakrabarty et al. [1] on three of the four categories except Grammaticality. Besides the full model, the without emoji and without context systems also outperform the state-of-the-art model on Sarcasticness, Creativity and Humor. Our system lacks in Grammaticality (Figure 5.2) due to the fact that we replace the rule based approach followed by Chakrabarty et al. [1] with a deep learning model which results in a slightly more significant information loss. However, the rule based model performs worse in case of the other three categories as it fails to generalize on all types of sentence structures. It is apparent from the scores that context plays an important role in recognising a sarcastic sentence. Additionally, the notable improvement in the score for full model compared to the without emoji model suggests that emojis obviously help better detect the incongruity that exist in sarcastic utterances. From the Figures 5.1, 5.2, 5.3 and 5.4, we can see that The baseline model got 0 in few of the samples because the model could not produce any output for these samples due to its some of the internal conditions, for example, they avoided the sentences starting with words like "can't", "don't", "won't" etc. Their model also distinguishes between the words "can't" and "cant" due to their rule-based approach. Using a deep learning model, our model is easily able to solve these types of problems. It is safe to say that, the baseline model is not quite suitable for informal language. Our dataset mainly consists posts from twitter users who casually expressed their emotions through informal language. Our deep learning model adapted with this informality of English language where it is not possible to impose a rule-based approach to deal with these unpredictable structure of sentences. As sarcasticness is mainly a human trait and used heavily in informal settings, it is quite necessary to direct the research of sarcasm generation which focuses on both formal and informal language.

System	**Sarcasticness**	**Creativity**	**Humor**	**Grammaticality**
Full Model	**3.29**	**3.44**	**3.16**	3.72
Without Emoji	2.83	2.77	2.69	3.7
Without Context	2.98	3.09	2.87	3.71
R^3 (Chakrabarty et al. [1])	2.2	2.32	2.1	**4.29**

TABLE 5.2: Average ratings by human judges for outputs from the four systems

5.1 Limitations

The proposed framework works well for short sentences. However, it is difficult for the model to identify and change the sentiment word in long sentences. That is why Sentence Neutralization is difficult in these cases and may result in loss of information. Even if neutralized properly, long sentences may need more words neutralized and induced to make it sound positive.

Although our proposed architecture successfully generates emoji-based sarcastic sentences from non-sarcastic texts, in some cases, particularly longer sentences, adding commonsense context does not add much to make it more sarcastic as in such cases, the longer sentences already contain the contextual information.

In our work, we have used $\text{COMET}_{\text{TIL}}^{\text{DIS}}$ to generate additional commonsense context. So the performance of our proposed architecture heavily depends on the accuracy of $\text{COMET}_{\text{TIL}}^{\text{DIS}}$.

The low grammaticality score by our final model is likely to be caused by the insufficient training data for the Positive Sentiment Induction module for which the model could not generalize properly. We believe that there is still room for improvement here by collecting and adding more training samples to improve the model's performance.

Another concern may arise which is why we used a round about way to reverse a sentence and why we did not just simply train a machine translation model which directly translates a negative sentence to a positive one. It is because there is no available dataset which holds a bunch of positive sentences and their negative counterpart and no available works that does so. Constructing a new dataset and evaluate its correctness and effectiveness is time consuming, costly and a separate research area by itself. So we tried to incorporate the already established method proposed by Mishra et al. [3] instead.

Lastly, our emoji prediction module only predicts one emoji per sentence. However, to make a sentence sarcastic, it is not uncommon to use more than one emoji.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We propose a novel multi-modular framework for sarcasm generation with emoji considering two key characteristics of sarcasm: reversal of valence and semantic incongruity between the sarcastic remark and the context. To generate sarcastic sentences, we first neutralize the input sentence's sentiment and then add a positive sentiment to reverse its meaning. We also incorporate a relevant emoji and additional contextual information to improve its sarcasticness. We conclude by evaluating our model using human judgement. In our work, we tried to show that adding emoji after a sarcastic sentence can elevate its sarcasticness. As sarcasm is hard to understand even by humans, the emoji cues can help further detect the sarcasm in a sentence. Our findings show that, adding emoji indeed increases the sarcasticness of a sentence. As, the trend of using emoji is increasing day by day, incorporating the use of emoji in sarcasm generation can open ways to further improve the generation process. Our results regarding the addition of contextual information improving the sarcasticness are also in line with the findings of Chakrabarty et al. [1]. In our work, we also tried to focus on informal language and work with sentences with informal and complex structures and showed that a rule-based approach is not efficient enough to reverse a sentence and training a powerful machine translation model with more data can further improve our results.

6.2 Future Work

To address the limitations of our system mentioned earlier, we plan to make some improvements to our work in the future. To solve the issue with longer sentences performing poorly, we will train with more data to make it more inclusive in future. For solving the redundant context information issue in future, we plan to modify our architecture in a way such that it can identify whether or not adding commonsense context would be necessary. Additionally, we would like to find and incorporate better models for generating commonsense context.

Similar to the problem with longer sentences, the information loss that leads to the low grammaticality score can be improved by adding more training data in future to generalize on input sentences even better. We also plan to explore multi-label emoji prediction in the future.

Bibliography

[1] Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. R3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[2] Aditya Joshi, Anoop Kunchukuttan, Pushpak Bhattacharyya, and Mark James Carman. Sarcasmbot: An open-source sarcasm-generation module for chatbots. In *WISDOM Workshop at KDD*, 2015.

[3] Abhijit Mishra, Tarun Tater, and Karthik Sankaranarayanan. A modular architecture for unsupervised sarcasm generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6144–6154, 2019.

[4] Aditya Joshi, Diptesh Kanojia, Pushpak Bhattacharyya, and Mark Carman. Sarcasm suite: a browser-based engine for sarcasm detection and generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[5] Spencer Cappallo, Thomas Mensink, and Cees GM Snoek. Image2emoji: Zero-shot emoji prediction for visual media. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1311–1314, 2015.

[6] Santosh Kumar Bharti, Bakhtyar Vachha, RK Pradhan, Korra Sathya Babu, and Sanjay Kumar Jena. Sarcastic sentiment detection in tweets streamed in real time: a big data approach. *Digital Communications and Networks*, 2(3):108–121, 2016.

[7] Jayashree Subramanian, Varun Sridharan, Kai Shu, and Huan Liu. Exploiting emojis for sarcasm detection. In *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation*, pages 70–80. Springer, 2019.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al.

Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[9] Basant Agarwal, Namita Mittal, Pooja Bansal, and Sonal Garg. Sentiment analysis using common-sense and context information. *Computational intelligence and neuroscience*, 2015, 2015.

[10] Aditya Joshi, Samarth Agrawal, Pushpak Bhattacharyya, and Mark Carman. Expect the unexpected: Harnessing sentence completion for sarcasm detection. *arXiv preprint arXiv:1707.06151*, 2017.

[11] Milagros Fernández-Gavilanes, Jonathan Juncal-Martínez, Silvia García-Méndez, Enrique Costa-Montenegro, and Francisco Javier González-Castaño. Creating emoji lexica from unsupervised sentiment analysis of their descriptions. *Expert Systems with Applications*, 103:74–91, 2018.

[12] Muhammad Abulaish and Ashraf Kamal. Self-deprecating sarcasm detection: an amalgamation of rule-based and machine learning approach. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 574–579. IEEE, 2018.

[13] Karthik Sundararajan and Anandhakumar Palanisamy. Multi-rule based ensemble feature selection model for sarcasm type detection in twitter. *Computational intelligence and neuroscience*, 2020, 2020.

[14] Ashraf Kamal and Muhammad Abulaish. An lstm-based deep learning approach for detecting self-deprecating sarcasm in textual data. In *Proceedings of the 16th International Conference on Natural Language Processing*, pages 201–210, 2019.

[15] Silviu Oprea and Walid Magdy. Exploring author context for detecting intended vs perceived sarcasm. *arXiv preprint arXiv:1910.11932*, 2019.

[16] Richard J Gerrig and Yevgeniya Goldvarg. Additive effects in the perception of sarcasm: Situational disparity and echoic mention. *Metaphor and Symbol*, 15(4): 197–208, 2000.

[17] Deirdre Wilson. The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116 (10):1722–1743, 2006.

[18] John D Campbell and Albert N Katz. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes*, 49(6):459–480, 2012.

[19] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. Sarcasm as contrast between a positive sentiment and negative

situation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 704–714, 2013.

[20] Christian Burgers, Margot Van Mulken, and Peter Jan Schellens. Verbal irony: Differences in usage across written genres. *Journal of Language and Social Psychology*, 31(3):290–310, 2012.

[21] Rachel Giora. On irony and negation. *Discourse processes*, 19(2):239–264, 1995.

[22] Suzana Ilić, Edison Marrese-Taylor, Jorge A Balazs, and Yutaka Matsuo. Deep contextualized word representations for detecting sarcasm and irony. *arXiv preprint arXiv:1809.09795*, 2018.

[23] Jona Dimovska, Marina Angelovska, Dejan Gjorgjevikj, and Gjorgji Madjarov. Sarcasm and irony detection in english tweets. In *International Conference on Telecommunications*, pages 120–131. Springer, 2018.

[24] Rolandos Alexandros Potamias, Georgios Siolas, and Andreas-Georgios Stafylopatis. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32(23):17309–17320, 2020.

[25] Usman Naseem, Imran Razzak, Peter Eklund, and Katarzyna Musial. Towards improved deep contextual embedding for the identification of irony and sarcasm. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[26] Jennifer Ling and Roman Klinger. An empirical, quantitative analysis of the differences between sarcasm and irony. In *European semantic web conference*, pages 203–216. Springer, 2016.

[27] Maria Khokhlova, Viviana Patti, and Paolo Rosso. Distinguishing between irony and sarcasm in social media texts: Linguistic observations. In *2016 International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT)*, pages 1–6. IEEE, 2016.

[28] Diana G Maynard and Mark A Greenwood. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Lrec 2014 proceedings*. ELRA, 2014.

[29] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*, 2017.

[30] Yitao Cai, Huiyu Cai, and Xiaojun Wan. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515, 2019.

[31] Santosh Kumar Bharti, Korra Sathya Babu, and Sanjay Kumar Jena. Parsing-based sarcasm sentiment recognition in twitter data. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1373–1380. IEEE, 2015.

[32] Santosh Kumar Bharti, Ramkrushna Pradhan, Korra Sathya Babu, and Sanjay Kumar Jena. Sarcasm analysis on twitter data using machine learning approaches. *Trends in Social Network Analysis*, pages 51–76, 2017.

[33] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*, 2016.

[34] Aniruddha Ghosh and Tony Veale. Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 161–169, 2016.

[35] Tanvi Dadu and Kartikey Pant. Sarcasm detection using context separators in online discourse. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 51–55, 2020.

[36] Avinash Kumar, Vishnu Teja Narapareddy, Pranjal Gupta, Veerubhotla Aditya Srikanth, Lalita Bhanu Murthy Neti, and Aruna Malapati. Adversarial and auxiliary features-aware bert for sarcasm detection. In *8th ACM IKDD CODS and 26th COMAD*, pages 163–170. 2021.

[37] Aditya Joshi, Pranav Goel, Pushpak Bhattacharyya, and Mark Carman. Automatic identification of sarcasm target: An introductory approach. *arXiv preprint arXiv:1610.07091*, 2016.

[38] Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*, 2018.

[39] Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. Cascade: Contextual sarcasm detection in online discussion forums. *arXiv preprint arXiv:1805.06413*, 2018.

[40] Chuhan Wu, Fangzhao Wu, Sixing Wu, Zhigang Yuan, Junxin Liu, and Yongfeng Huang. Thu_ngn at semeval-2018 task 2: Residual cnn-lstm network with attention for english emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 410–414, 2018.

[41] Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. Are word embedding-based features useful for sarcasm detection? *arXiv preprint arXiv:1610.00883*, 2016.

[42] Paras Dharwal, Tanupriya Choudhury, Rajat Mittal, and Praveen Kumar. Automatic sarcasm detection using feature selection. In *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 29–34. IEEE, 2017.

[43] Ameeta Agrawal and Aijun An. Affective representations for sarcasm detection. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1029–1032, 2018.

[44] Natalie Parde and Rodney Nielsen. Detecting sarcasm is extremely easy;-. In *Proceedings of the workshop on computational semantics beyond events and roles*, pages 21–26, 2018.

[45] Elena Filatova. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Lrec*, pages 392–398. Citeseer, 2012.

[46] Shereen Oraby, Vrindavan Harrison, Amita Misra, Ellen Riloff, and Marilyn Walker. Are you serious?: Rhetorical questions and sarcasm in social media dialog. *arXiv preprint arXiv:1709.05305*, 2017.

[47] Xinyu Wang, Xiaowen Sun, Tan Yang, and Hongbo Wang. Building a bridge: a method for image-text sarcasm detection without pretraining on image-text data. In *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*, pages 19–29, 2020.

[48] Hongliang Pan, Zheng Lin, Peng Fu, and Weiping Wang. Modeling the incongruity between sentence snippets for sarcasm detection. In *ECAI 2020*, pages 2132–2139. IOS Press, 2020.

[49] Nan Xu, Zhixiong Zeng, and Wenji Mao. Reasoning with multimodal sarcastic tweets via modeling cross-modality contrast and semantic association. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3777–3786, 2020.

[50] Rossano Schifanella, Paloma De Juan, Joel Tetreault, and Liangliang Cao. Detecting sarcasm in multimodal social platforms. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1136–1145, 2016.

[51] Tomáš Ptáček, Ivan Habernal, and Jun Hong. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 213–223, 2014.

[52] Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. Modelling sarcasm in twitter, a novel approach. In *proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 50–58, 2014.

[53] David Bamman and Noah Smith. Contextualized sarcasm detection on twitter. In *proceedings of the international AAAI conference on web and social media*, volume 9, pages 574–577, 2015.

[54] Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*, 2016.

[55] Meishan Zhang, Yue Zhang, and Guohong Fu. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers*, pages 2449–2460, 2016.

[56] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*, 2017.

[57] Aniruddha Ghosh and Tony Veale. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 482–491, 2017.

[58] Anukarsh G Prasad, S Sanjana, Skanda M Bhat, and BS Harish. Sentiment analysis for sarcasm detection on streaming short text data. In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, pages 1–5. IEEE, 2017.

[59] Christos Baziotis, Nikos Athanasiou, Georgios Paraskevopoulos, Nikolaos Ellinas, Athanasia Kolovou, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 2: Predicting emojis using rnns with context-aware attention. *arXiv preprint arXiv:1804.06657*, 2018.

[60] Debanjan Ghosh, Alexander R Fabbri, and Smaranda Muresan. Sarcasm analysis using conversation context. *Computational Linguistics*, 44(4):755–792, 2018.

[61] Cynthia Van Hee, Els Lefever, and Véronique Hoste. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, 2018.

[62] Navonil Majumder, Soujanya Poria, Haiyun Peng, Niyati Chhaya, Erik Cambria, and Alexander Gelbukh. Sentiment and sarcasm classification with multitask learning. *IEEE Intelligent Systems*, 34(3):38–43, 2019.

[63] Akshi Kumar, Saurabh Raj Sangwan, Anshika Arora, Anand Nayyar, Mohamed Abdel-Basset, et al. Sarcasm detection using soft attention-based bidirectional

long short-term memory model with convolution network. *IEEE access*, 7:23319–23328, 2019.

[64] Amit Kumar Jena, Aman Sinha, and Rohit Agarwal. C-net: Contextual network for sarcasm detection. In *Proceedings of the second workshop on figurative language processing*, pages 61–66, 2020.

[65] Lakshya Kumar, Arpan Somani, and Pushpak Bhattacharyya. " having 2 hours to write a paper is fun!": Detecting sarcasm in numerical portions of text. *arXiv preprint arXiv:1709.01950*, 2017.

[66] Abhijeet Dubey, Lakshya Kumar, Arpan Somani, Aditya Joshi, and Pushpak Bhattacharyya. "when numbers matter!": Detecting sarcasm in numerical portions of text. In *Proceedings of the tenth workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 72–80, 2019.

[67] Dipto Das and Anthony J Clark. Sarcasm detection on facebook: A supervised learning approach. In *Proceedings of the 20th International Conference on Multimodal Interaction: Adjunct*, pages 1–5, 2018.

[68] Aytuğ Onan. Topic-enriched word embeddings for sarcasm identification. In *Computer science on-line conference*, pages 293–304. Springer, 2019.

[69] Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. The role of conversation context for sarcasm detection in online interactions. *arXiv preprint arXiv:1707.06226*, 2017.

[70] K Sreelakshmi and PC Rafeeque. An effective approach for detection of sarcasm in tweets. In *2018 International CET Conference on Control, Communication, and Computing (IC4)*, pages 377–382. IEEE, 2018.

[71] Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In *The world wide web conference*, pages 2115–2124, 2019.

[72] Ramish Jamil, Imran Ashraf, Furqan Rustam, Eysha Saad, Arif Mehmood, and Gyu Sang Choi. Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model. *PeerJ Computer Science*, 7:e645, 2021.

[73] Tanya Jain, Nilesh Agrawal, Garima Goyal, and Niyati Aggrawal. Sarcasm detection of tweets: A comparative study. In *2017 Tenth International Conference on Contemporary Computing (IC3)*, pages 1–6. IEEE, 2017.

[74] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*, 2016.

[75] Siti Khotijah, Jimmy Tirtawangsa, and Arie A Suryani. Using lstm for context based approach of sarcasm detection in twitter. In *Proceedings of the 11th International Conference on Advances in Information Technology*, pages 1–7, 2020.

[76] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

[77] Pulkit Mehndiratta and Devpriya Soni. Identification of sarcasm using word embeddings and hyperparameters tuning. *Journal of Discrete Mathematical Sciences and Cryptography*, 22(4):465–489, 2019.

[78] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

[79] Damian Borth, Tao Chen, Rongrong Ji, and Shih-Fu Chang. Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 459–460, 2013.

[80] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

[81] Shubhadeep Mukherjee and Pradip Kumar Bala. Sarcasm detection in microblogs using naïve bayes and fuzzy clustering. *Technology in Society*, 48:19–27, 2017.

[82] Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, 2011.

[83] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcasm in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116, 2010.

[84] Santosh Kumar Bharti, Ramkrushna Pradhan, Korra Sathya Babu, and Sanjay Kumar Jena. Sarcastic sentiment detection based on types of sarcasm occurring in twitter data. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 13(4):89–108, 2017.

[85] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, 2015.

[86] Gavin Abercrombie and Dirk Hovy. Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations. In *Proceedings of the ACL 2016 student research workshop*, pages 107–113, 2016.

[87] Shubhadeep Mukherjee and Pradip Kumar Bala. Detecting sarcasm in customer tweets: an nlp based approach. *Industrial Management & Data Systems*, 2017.

[88] Sakshi Thakur, Sarbjeet Singh, and Makhan Singh. Detecting sarcasm in text. In *International Conference on Intelligent Systems Design and Applications*, pages 996–1005. Springer, 2018.

[89] Akshay Khatri et al. Sarcasm detection in tweets with bert and glove embeddings. *arXiv preprint arXiv:2006.11512*, 2020.

[90] Christopher Ifeanyi Eke, Azah Norman, Liyana Shuib, Faith B Fatokun, and Isaiah Omame. The significance of global vectors representation in sarcasm analysis. In *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, pages 1–7. IEEE, 2020.

[91] Karthik Sundararajan, J Vijay Saravana, and Anandhakumar Palanisamy. Textual feature ensemble-based sarcasm detection in twitter data. In *Intelligence in Big Data Technologies—Beyond the Hype*, pages 443–450. Springer, 2021.

[92] Arghasree Banerjee, Mayukh Bhattacharjee, Kushankur Ghosh, and Sankhadeep Chatterjee. Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media. *Multimedia Tools and Applications*, 79(47):35995–36031, 2020.

[93] Mattia Antonino Di Gangi, Giosué Lo Bosco, and Giovanni Pilato. Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection. *Natural Language Engineering*, 25(2):257–285, 2019.

[94] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[95] Saurabh Porwal, Gaurav Ostwal, Anagha Phadtare, Mohini Pandey, and Manisha V Marathe. Sarcasm detection using recurrent neural network. In *2018 second international conference on intelligent computing and control systems (ICICCS)*, pages 746–748. IEEE, 2018.

[96] Sayed Saniya Salim, Agrawal Nidhi Ghanshyam, Darkunde Mayur Ashok, Dungarpur Burhanuddin Mazahir, and Bhushan S Thakare. Deep lstm-rnn with word embedding for sarcasm detection on twitter. In *2020 international conference for emerging technology (INCET)*, pages 1–4. IEEE, 2020.

[Guo and Shah] Nick Guo and Ruchir Shah. Finding sarcasm in reddit postings: A deep learning approach.

[97] Yufeng Diao, Hongfei Lin, Liang Yang, Xiaochao Fan, Yonghe Chu, Kan Xu, and Di Wu. A multi-dimension question answering network for sarcasm detection. *IEEE Access*, 8:135152–135161, 2020.

[98] Yafeng Ren, Donghong Ji, and Han Ren. Context-augmented convolutional neural networks for twitter sarcasm detection. *Neurocomputing*, 308:1–7, 2018.

[99] Y Alex Kolchinski and Christopher Potts. Representing social media users for sarcasm detection. *arXiv preprint arXiv:1808.08470*, 2018.

[100] Rishabh Misra and Prahal Arora. Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*, 2019.

[101] Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti. Sarcasm detection using multi-head attention based bidirectional lstm. *Ieee Access*, 8:6388–6397, 2020.

[102] Yiyi Liu, Yequan Wang, Aixin Sun, Zheng Zhang, Jiafeng Guo, and Xuying Meng. A dual-channel framework for sarcasm recognition by detecting sentiment conflict. *arXiv preprint arXiv:2109.03587*, 2021.

[103] Ramya Akula and Ivan Garibay. Explainable detection of sarcasm in social media. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 34–39, 2021.

[104] Edoardo Savini and Cornelia Caragea. A multi-task learning approach to sarcasm detection (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13907–13908, 2020.

[105] Rolandos-Alexandros Potamias, Georgios Siolas, and Andreas Stafylopatis. A robust deep ensemble classifier for figurative language detection. In *International Conference on Engineering Applications of Neural Networks*, pages 164–175. Springer, 2019.

[106] Rahul Gupta, Jitendra Kumar, Harsh Agrawal, et al. A statistical approach for sarcasm detection using twitter data. In *2020 4th international conference on intelligent computing and control systems (ICICCS)*, pages 633–638. IEEE, 2020.

[107] Jens Lemmens, Ben Burtenshaw, Ehsan Lotfi, Ilia Markov, and Walter Daelemans. Sarcasm detection using an ensemble approach. In *proceedings of the second workshop on figurative language processing*, pages 264–269, 2020.

[108] Angela Fan, Thibaut Lavril, Edouard Grave, Armand Joulin, and Sainbayar Sukhbaatar. Addressing some limitations of transformers with feedback memory. *arXiv preprint arXiv:2002.09402*, 2020.

[109] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[110] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[111] Himani Srivastava, Vaibhav Varshney, Surabhi Kumari, and Saurabh Srivastava. A novel hierarchical bert architecture for sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 93–97, 2020.

[112] Hunter Gregory, Steven Li, Pouya Mohammadi, Natalie Tarn, Rachel Draelos, and Cynthia Rudin. A transformer approach to contextual sarcasm detection in twitter. In *Proceedings of the second workshop on figurative language processing*, pages 270–275, 2020.

[113] A Kalaivani and D Thenmozhi. Sarcasm identification and detection in conversion context using bert. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 72–76, 2020.

[114] Soroush Javdan, Behrouz Minaei-Bidgoli, et al. Applying transformers and aspect-based sentiment analysis approaches on sarcasm detection. In *Proceedings of the second workshop on figurative language processing*, pages 67–71, 2020.

[115] Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. Lcf: A local context focus mechanism for aspect-based sentiment classification. *Applied Sciences*, 9(16): 3389, 2019.

[116] Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, and David Eyers. Bert's the word: Sarcasm target detection using bert. In *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*, pages 185–191, 2021.

[117] Chenwei Lou, Bin Liang, Lin Gui, Yulan He, Yixue Dang, and Ruifeng Xu. Affective dependency graph for sarcasm detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1844–1849, 2021.

[118] Boaz Shmueli, Lun-Wei Ku, and Soumya Ray. Reactive supervision: A new method for collecting sarcasm data. *arXiv preprint arXiv:2009.13080*, 2020.

[119] Raj Kumar Gupta and Yinping Yang. Crystalnest at semeval-2017 task 4: Using sarcasm detection for enhancing sentiment classification and quantification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 626–633, 2017.

[120] Hankyol Lee, Youngjae Yu, and Gunhee Kim. Augmenting data for sarcasm detection with unlabeled conversation context. *arXiv preprint arXiv:2006.06259*, 2020.

[121] Arup Baruah, Kaushik Das, Ferdous Barbhuiya, and Kuntal Dey. Context-aware sarcasm detection using bert. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 83–87, 2020.

[122] Adithya Avvaru, Sanath Vobilisetty, and Radhika Mamidi. Detecting sarcasm in conversation context using transformer-based models. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 98–103, 2020.

[123] Nikhil Jaiswal. Neural sarcasm detection using conversation context. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 77–82, 2020.

[124] Xiangjue Dong, Changmao Li, and Jinho D Choi. Transformer-based context-aware sarcasm detection in conversation threads from social media. *arXiv preprint arXiv:2005.11424*, 2020.

[125] Amardeep Kumar and Vivek Anand. Transformers on sarcasm detection with context. In *Proceedings of the second workshop on figurative language processing*, pages 88–92, 2020.

[126] Debanjan Ghosh, Avijit Vajpayee, and Smaranda Muresan. A report on the 2020 sarcasm detection shared task. *arXiv preprint arXiv:2005.05814*, 2020.

[127] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[128] Jasabanta Patro, Srijan Bansal, and Animesh Mukherjee. A deep-learning framework to detect sarcasm targets. In *proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6336–6342, 2019.

[129] Abhijeet Dubey, Aditya Joshi, and Pushpak Bhattacharyya. Deep models for converting sarcastic utterances into their non sarcastic interpretation. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 289–292, 2019.

[130] Joseph Tepperman, David Traum, and Shrikanth Narayanan. " yeah right": sarcasm recognition for spoken dialogue systems. In *Ninth international conference on spoken language processing*, 2006.

[131] Byron C Wallace, Laura Kertz, Eugene Charniak, et al. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–516, 2014.

[132] Aditya Joshi, Prayas Jain, Pushpak Bhattacharyya, and Mark Carman. Who would have thought of that!': A hierarchical topic model for extraction of sarcasm-prevalent topics and sarcasm detection. *arXiv preprint arXiv:1611.04326*, 2016.

[133] Joan Plepi and Lucie Flek. Perceived and intended sarcasm detection with graph attention networks. *arXiv preprint arXiv:2110.04001*, 2021.

[134] Zelin Wang, Zhijian Wu, Ruimin Wang, and Yafeng Ren. Twitter sarcasm detection exploiting a context-based model. In *international conference on web information systems engineering*, pages 77–91. Springer, 2015.

[135] Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. Hidden markov support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 3–10, 2003.

[136] Andrea Vanzo, Danilo Croce, and Roberto Basili. A context-based model for sentiment analysis in twitter. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*, pages 2345–2354, 2014.

[137] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[138] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[139] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[140] Rongcheng Lin, Jing Xiao, and Jianping Fan. Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[141] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[142] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020.

[143] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. Fusion of detected objects in text for visual question answering. *arXiv preprint arXiv:1908.05054*, 2019.

[144] Silviu Oprea, Steven Wilson, and Walid Magdy. Chandler: An explainable sarcastic response generator. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 339–349, 2021.

[145] Sourav Das, Soumitra Ghosh, Anup Kumar Kolya, and Asif Ekbal. Unparalleled sarcasm: a framework of parallel deep lstms with cross activation functions towards detection and generation of sarcastic statements. *Language Resources and Evaluation*, pages 1–38, 2022.

[146] Jie Ruan, Yue Wu, Xiaojun Wan, and Yuesheng Zhu. How to describe images in a more funny way? towards a modular approach to cross-modal sarcasm generation. *arXiv preprint arXiv:2211.10992*, 2022.

[147] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[148] Joe Davison, Joshua Feldman, and Alexander M Rush. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 1173–1178, 2019.

[149] Zhenjie Zhao, Evangelos Papalexakis, and Xiaojuan Ma. Learning physical common sense as knowledge graph completion via bert data augmentation and constrained tucker factorization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3293–3298, 2020.

[150] Prajjwal Bhargava and Vincent Ng. Commonsense knowledge reasoning and generation with pre-trained language models: a survey. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12317–12325, 2022.

[151] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.

[152] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[153] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE: Confederated International Conferences CoopIS, DOA, and ODBASE 2002 Proceedings*, pages 1223–1237. Springer, 2002.

[154] Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392, 2021.

[155] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[156] Peter West, Chandra Bhagavatula, Jack Hessel, Jena D Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178*, 2021.

[157] Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2002–2012, 2019.

[158] Jian Guan, Yansen Wang, and Minlie Huang. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6473–6480, 2019.

[159] Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108, 2020.

[160] Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. Think before you speak: Using self-talk to generate implicit commonsense knowledge for response generation. *arXiv preprint arXiv:2110.08501*, 2021.

[161] Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. Think before you speak: Explicitly generating implicit commonsense knowledge for response generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1252, 2022.

[162] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018.

[163] Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5811–5820, 2020.

[164] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. Grounded conversation generation as guided traverses in commonsense knowledge graphs. *arXiv preprint arXiv:1911.02707*, 2019.

[165] Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. Topicka: Generating commonsense knowledge-aware dialogue responses towards the recommended

topic fact. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3766–3772, 2021.

[166] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[167] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[168] Tianqiao Liu, Qiang Fang, Wenbiao Ding, Hang Li, Zhongqin Wu, and Zitao Liu. Mathematical word problem generation from commonsense knowledge graph and equations. *arXiv preprint arXiv:2010.06196*, 2020.

[169] Aditi Chaudhary, Shirley Anugrah Hayati, Naoki Otani, and Alan W Black. What a sunny day: toward emoji sensitive irony detection. *Proc. W-NUT*, page 212, 2019.

[170] Dushyant Singh Chauhan, Gopendra Vikram Singh, Aseem Arora, Asif Ekbal, and Pushpak Bhattacharyya. An emoji-aware multitask framework for multimodal sarcasm detection. *Knowledge-Based Systems*, 257:109924, 2022.

[171] Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. Are emojis predictable? *arXiv preprint arXiv:1702.07285*, 2017.

[172] Çağrı Çöltekin and Taraka Rama. Tübingen-oslo at semeval-2018 task 2: Svms perform better than rnns in emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 34–38, 2018.

[173] Joël Coster, Reinder Gerard van Dalen, and Nathalie Adriënne Jacqueline Stierman. Hatching chick at semeval-2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 445–448, 2018.

[174] Daphne Groot, Rémon Kruizinga, Hennie Veldthuis, Simon de Wit, and Hessel Haagsma. Pickleteam! at semeval-2018 task 2: English and spanish emoji prediction from tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 454–458, 2018.

[175] Jing Chen, Dechuan Yang, Xilian Li, Wei Chen, and Tengjiao Wang. Peperomia at semeval-2018 task 2: Vector similarity based approach for emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 428–432, 2018.

[176] Gaël Guibon, Magalie Ochs, Patrice Bellot, G Guibon, M Ochs, P Bellot, et al. From emoji usage to categorical emoji prediction. In *19th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING 2018). Springer Lecture Notes in Computer Science, Switzerland*, volume 10, 2018.

[177] Dunlu Peng and Huimin Zhao. Seq2emoji: A hybrid sequence generation model for short text emoji prediction. *Knowledge-Based Systems*, 214:106727, 2021.

[178] Francesco Barbieri, Luis Marujo, Pradeep Karuturi, William Brendel, and Horacio Saggion. Exploring emoji usage and prediction through a temporal variation lens. *arXiv preprint arXiv:1805.00731*, 2018.

[179] Francesco Barbieri, Miguel Ballesteros, Francesco Ronzano, and Horacio Saggion. Multimodal emoji prediction. *arXiv preprint arXiv:1803.02392*, 2018.

[180] Angelo Basile and Kenny W Lino. Tajjeb at semeval-2018 task 2: Traditional approaches just do the job with emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 470–476, 2018.

[181] Larisa Alexa, Alina Beatrice Lorent, Daniela Gifu, and Diana Trandabat. The dabblers at semeval-2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 405–409, 2018.

[182] Daniel Kopev, Atanas Atanasov, Dimitrina Zlatkova, Momchil Hardalov, Ivan Koychev, Ivelina Nikolova, and Galia Angelova. Tweety at semeval-2018 task 2: Predicting emojis using hierarchical attention neural networks and support vector machine. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 497–501, 2018.

[183] Zhenduo Wang and Ted Pedersen. Umdsub at semeval-2018 task 2: Multilingual emoji prediction multi-channel convolutional neural network on subword embedding. *arXiv preprint arXiv:1805.10274*, 2018.

[184] Jonathan Beaulieu and Dennis Asamoah Owusu. Umduluth-cs8761 at semeval-2018 task 2: Emojis: Too many choices? In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 400–404, 2018.

[185] Nan Wang, Jin Wang, and Xuejie Zhang. Ynu-hpcc at semeval-2018 task 2: Multi-ensemble bi-gru model with attention mechanism for multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 459–465, 2018.

[186] Toshiki Tomihira, Atsushi Otsuka, Akihiro Yamashita, and Tetsuji Satoh. What does your tweet emotion mean? neural emoji prediction for sentiment analysis.

In *proceedings of the 20th international conference on information integration and web-based applications & services*, pages 289–296, 2018.

[187] Chuhan Wu, Fangzhao Wu, Sixing Wu, Yongfeng Huang, and Xing Xie. Tweet emoji prediction using hierarchical model with attention. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 1337–1344, 2018.

[188] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[189] Jingjing Xu, Xu Sun, Qi Zeng, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. *arXiv preprint arXiv:1805.05181*, 2018.

[190] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*, 2017.

[191] Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*, 2019.

[192] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[193] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 597–606, 2015.

[194] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.