Master Thesis Proposal

THE IMPACT OF CONVERSATION CONTEXT ON IDENTIFYING IRONY IN SOCIAL MEDIA

Student:
Caner Coban
Institute:
Eberhard Karls Universität Tübingen
E-mail:
caner.coban@student.uni-tuebingen.de

Advisor:
Cagri Coltekin

Department of Computational Linguistics
April 8, 2024

Contents

List of Figures

List of Tables

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst habe, dass ich keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe, dass ich alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist, dass ich die Arbeit weder vollständig noch in wesentlichen Teilen bereits veröffentlicht habe, dass das in Dateiform eingereichte Exemplar mit dem eingereichten gebundenen Exemplar übereinstimmt.

I hereby declare that this Master's thesis titled "The Impact of Conversation Context on Identifying Irony in Social Media" is the result of my own independent scholarly work and has been obtained and presented in accordance with academic rules and ethical conduct. I have acknowledged all the other author's ideas and referenced direct quotations from their work.

Name: Caner COBAN

Signature:

To my family.

Master Thesis Proposal

April 8, 2024

Abstract

In this thesis, we explore the critical role of context in detecting irony in social media communications, focusing on the challenges posed by written forms of sarcasm and irony. Irony, a pervasive element in human interaction, often involves a juxtaposition of positive expressions and negative situations, especially apparent on platforms like Twitter. Accurately interpreting such ironic statements is crucial for natural language processing (NLP) systems to prevent the misinterpretation of irony as literal speech.

Our research centers on using a state-of-the-art Turkish BERT model, applied to a uniquely compiled dataset of Turkish social media content collected through the Twitter API. This approach aims to discern the impact of incorporating conversational context in automatic irony detection. Additionally, we extended our study to include the application of the BERT transformer model on an English dataset, allowing for a comparative analysis.

While our findings reveal that contextual information can significantly enhance model accuracy in both the Turkish and English datasets, the improvement in the Turkish dataset, although not as uniform across all metrics as in the English dataset, still demonstrates a notable advancement in certain areas. This differential performance illuminates intriguing aspects of the Turkish and English datasets and opens avenues for specialized improvements and targeted research in this area.

Ultimately, this study contributes to the broader understanding of how advanced computational models, including transformer models like BERT, can be fine-tuned for more nuanced language tasks. The insights gained here underscore the complexity of irony detection in digital communication and pave the way for future advancements in this domain.

1 Introduction

Irony has always been a significant aspect of human communication, making it essential to understand for enhancing our grasp of human language and interaction (Van Hee et al., 2018a). Researchers like Grice (1975) have contributed to this understanding by suggesting that irony and similar forms of figurative language are violations of the conversational maxim of quality. According to Van Hee (2017), such violations by the speaker are intentional, aimed at focusing the listener's attention and encouraging deeper exploration to recognize the irony. Similar theories have been developed, such as Echoic Mention Theory by Sperber and Wilson (1981). A distinct view is offered by Giora (1995), who describes irony as a form of indirect negation, including understatement and exaggerations. Giora's approach is notable for suggesting that an ironic statement's meaning coexists with its literal interpretation, underscoring the contrast between them, which diverges from previous theories. A common proposal is that sarcasm is a variant of verbal irony (Colston, 2000). Sarcasm, as a subtype of verbal irony, involves expressing a negative and critical attitude toward a target or group of victims, setting it apart from basis irony. The Oxford Dictionary[1] defines sarcasm as a sharp, bitter, or cutting expression or remark; a bitter gibe or taunt. Accurate detection of sarcasm is crucial for natural language processing systems to prevent misinterpretation of sarcastic remarks as literal statements. On Social platforms like Twitter, a common form of sarcasm is evident in the contrast of positive sentiments with descriptions of unfavorable situations, providing a clear indicator for computational models. Sarcasm detection is framed as a binary classification task, predicting whether a given sentence is sarcastic or not. However, it is not always possible to decide if a sentence is sarcastic or not without knowing its context. The recent advancements in the field leverage contextual information beyond the target text to enhance accuracy in identifying sarcasm across diverse expressions.

The challenge of detecting irony and sarcasm in both spoken and written language has garnered significant interest in the fields of Computational Linguistics and Computer Science. While in spoken discourse, irony is relatively easier to identify through vocal cues and word emphasis, as noted by authors like (Cutler, 1974) and Haiman (1998), it becomes far more complex in written text. The subtleties of written language, devoid of explicit cues, require sophisticated computational models to solve layers of meaning. This complexity is especially evident in written irony, where the text often conveys the opposite of an apparent fact without clear indicators. In a

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

similar vein, understanding sarcasm in text, particularly in text-only social media content, is crucial for natural language understanding. Sarcasm often expresses a sentiment contrary to its literal meaning and relies heavily on contextual cues, such as tone of voice or facial expressions, which are absent in written forms (Gregory et al. (2020) and Gupta et al. (2017)). Wallace et al. (2014) emphasized the importance of context in sarcasm detection, noting that annotators frequently require additional information to accurately label texts. Thus the challenges in detecting sarcasm and irony in written forms, particularly on social media, have led to the use of advanced models like LSTM (Long short-term memory) and BERT (Bidirectional Encoder Representations), with transformer models showing promising results. These models often require a comprehensive understanding of context, a factor we investigate in this study by employing automated methods to gather contextual elements from conversations containing potentially sarcastic tweets.

In light of the complexities in detecting irony and sarcasm in written forms, especially on social media, this paper delves into the efficacy of advanced transformer models like BERT (Devlin et al., 2018) in automatically identifying these linguistic nuances. Our research particularly emphasizes the crucial role of context in understanding such subtleties, a factor often overlooked in previous systems. We aim to conduct a series of experiments to assess how effectively these models can interpret irony and sarcasm when provided with adequate conversational context, as suggested by the importance of context in the correct interpretation of sarcastic and ironic statements, highlighted by researchers like Gregory et al. (2020),Gupta et al. (2017), and Wallace et al. (2014). This approach aims to bridge the gap identified in earlier systems where a lack of contextual information impeded accurate model training and interpretation.

The studies on this task encompass a broad spectrum of languages, including English (Potamias et al., 2020), Dutch (Maladry et al., 2022), and Portuguese (Jiang et al., 2021), among others. While some methods for irony detection are universally applied across languages, there are also unique linguistic features specific to each language that are leveraged for more accurate detection of irony.

Turkish is notable for its morphological complexity, mainly due to its agglutinative nature. In this language, longer words are formed by attaching morphemes, each with distinct grammatical or semantic roles, to a root word. This addition of various suffixes creates nuanced meanings.

The agglutinative feature of Turkish allows for the creation of almost limitless new words. For example, the Turkish word “gelmemeliydim” translates to the sentence “I should not have come” in English, showcasing the language’s ability to express complex ideas in a single word. Understanding these intricate word constructions in Turkish requires morphological analysis, which is crucial for deconstructing words and uncovering their layered meanings.

Another distinction we can talk about comparing the Turkish to English where negation is often signaled by the inclusion of “not” within the sentence. However, Turkish employs a different strategy. Negation in Turkish frequently involves the addition of suffixes such as “-me” or “-ma” to verbs (e.g., “Ironi yaptım/yapmadım”). While the word “değil” serves a similar purpose, it is not as commonly used as “not” in English. This highlights the importance of acknowledging language-specific differences when engaging in Natural Language Processing (NLP) tasks. For example, in sentiment analysis for English, it is common to remove suffixes to simplify analysis. However, this approach does not consistently apply to Turkish because of the mentioned morphological complexities that characterize the language. Thus NLP practitioners must embrace a different understanding of Turkish morphology to create effective and culturally specific language models and applications.

The growing popularity of social media, driven by a need to connect and stay informed, has positioned platforms such as Twitter at the forefront of contemporary life. As of December 2022, Twitter boasts over 368 million monthly users and sees 500 million tweets daily, reflecting its status as a major information hub (Dixon, 2022). This great amount of varied content, from simple language to complex linguistic expressions, is invaluable for Natural Language Processing (NLP) research. Twitter’s diverse user-generated content offers a unique resource for understanding language nuances in digital communication, making it prime for NLP studies. The platform’s rich linguistic data is key for advancing natural language understanding. Building on the significance of social media as a major information hub, with Twitter being a prime example, this thesis delves into the complexities of human communication on such platforms. Through this, we aim to develop a theoretical framework and practical tools for irony detection in both English and Turkish social media content.

Sarcasm/Irony detection is highly dependent on access to substantial amounts of labeled data (Shmueli et al., 2020). In the English language domain, various studies have made significant contributions by creating datasets specifically for feeding into machine learning models. For instance, Bamman and Smith (2015) developed a comprehensive English dataset containing around 20K samples, enriched with author and conversational context. Oprea and Magdy (2019) introduced a unique approach by defining the author's context as the embedded representation of their historical Twitter posts. They created two datasets: one manually labeled for sarcasm and another using tag-based distant supervision. More recently, Shmueli et al. (2020) succeeded in developing a dataset encompassing both perceived and intended sarcasm. They employed their novel method of Reactive Supervision, which included nearly 15,000 tweets along with their conversational context.

However, this abundance of datasets is not reflected in the Turkish language context. The notable exception is the IronyTR dataset, developed by Ozturk et al. (2021). This dataset was specifically developed for irony detection in Turkish, sourced from Twitter and other social media platforms. It utilized hashtags and event-related identifiers to pinpoint instances of irony. Unlike the mentioned English irony datasets, the IronyTR dataset lacks conversational context, focusing solely on individual ironic tweets.

The scarcity and need for such datasets in Turkish motivated us to develop a sarcasm-specific dataset for the language. Our dataset not only includes individual sarcastic tweets but also other tweets from the same conversational threads. Thus, we aim to contribute to the field of automatic irony and sarcasm detection for the Turkish language by creating a new and reliable dataset, which we have named the Turkish Sarcasm Corpus (TSC). Consequently, this dataset not only serves as a valuable resource for understanding and analyzing sarcasm within the Turkish language but also addresses the need for contextually rich datasets in the domain of sarcasm detection.

Irony and sarcasm are rhetorical devices commonly found in everyday communication. With the advent of social media platforms, the interest increases to differentiate ironic and sarcastic textual contributions from regular ones (Ling and Klinger, 2016). However, many research efforts in this area treat irony and sarcasm as sufficiently similar concepts, thereby not necessitating a distinct separation in their identification (Clift, 1999). In line with this approach, our study does not differentiate between the two. Throughout this paper, the terms 'irony' and 'sarcasm' will be used interchangeably to maintain clarity and avoid confusion for the reader. Consequently, the methodologies and procedures discussed are applicable to both irony and sarcasm detection research.

Our research focuses on understanding how the inclusion of conversational context in the training data influences the efficacy of transformer models in the binary classification of irony within written Turkish text. To answer this, we first employ the Reactive Supervision method, as discussed by Shmueli et al. (2020), to develop a comprehensive Turkish Sarcasm Corpus. This corpus is unique as it includes sarcastic and non-sarcastic tweets, each paired with its conversational context, and is organized into four versions with varying distributions of sarcastic and non-sarcastic content. Our experimental approach involves two phases. In the initial phase, we train transformer models exclusively on the text of the tweets, omitting any conversational context. This assesses the models' baseline ability to identify irony. Subsequently, in the second phase, we retrain these models with the same parameters but incorporate the conversational context into the training data. This approach allows us to directly compare the impact of including conversational context on the models' performance in irony detection, using a consistent test dataset across both experiments for each version of the corpus.

This thesis is structured as follows: Chapter 2 provides background information on irony and the task of automatic irony detection. It outlines the data sources, collection tools, classification techniques, and model algorithms. Chapter 3 reviews previous studies in the field of irony detection, detailing the approaches and datasets employed historically. Our methodology for data collection, annotation, and classification is elaborated in Chapter 4. Chapter 5 presents our experiments and offers a detailed examination of the results derived from these experiments. Finally, Chapter 6 discusses the conclusions and potential directions for future research.

2 Background Information

2.1 Irony

According to Van Hee 2017, Irony, a subtle literary device, appears in different forms, and there are three distinct types: verbal, situational, and dramatic irony. These contribute depth and complexity to literary works. Among these, verbal irony takes a prominent position, involving the deliberate expression of statements where intended meaning sharply contrasts with the literal words spoken. This intentional contradiction often takes the form of sarcasm, a biting subset of verbal irony that adds a layer of sharp wit to communication. For example, if someone consistently arrives late to meetings, a sarcastic comment might be, "Oh look, the early bird has finally decided to grace us with their presence!".

Situational irony, another aspect of irony, occurs when the result of a situation differs significantly from what was anticipated. This form relies on the contrast between expected outcomes and the actual unfolding of events, introducing twists that can evoke surprise or humor. A classic illustration of situational irony is a fire station itself catching fire, defying expectations as a place dedicated to fire prevention becomes an unexpected victim of flames.

Dramatic Irony is the third type of irony that moves beyond the tangible events, dramatic irony introduces a distinctive narrative dynamic. In this form, readers or viewers possess information unknown to the characters, creating suspense as the characters navigate unfolding events unaware of crucial details. An iconic example takes place in a Titanic-themed narrative when a character admires the beauty of the scene just before the ship collides with an iceberg, a moment filled with dramatic irony as the audience understands the impending disaster.

Among these three types of irony, our study takes a deeper interest in verbal irony, particularly in the context of social media. This focus is driven by the unique characteristics of communication in digital platforms, where text-based interactions often lack non-verbal cues, making the detection and interpretation of irony more challenging.

Within the domain of linguistic comprehension, both machine learning methods and human abilities face the complex task of accurately identifying irony. In spoken language, cues such as tone, facial expressions, and the duration of pauses provide vital indicators of ironic intent, facilitating a more immediate perception of irony in conversation (Taşlıoğlu, 2014). However, Irony in written text presents a challenge, with the focus of this research centering on verbal irony, particularly sarcasm. This research emphasizes the need for developing linguistic indicators specifically designed for Turkish to identify sarcasm in written text.

2.2 Automatic Detection of Irony/Sarcasm

Sarcasm, an intricate aspect of human language and culture, extends beyond the broader umbrella of irony, embodying a distinct form that often relates to irony with a touch of mockery or criticism. Specifically categorized as a subtype of verbal irony, sarcasm is characterized by the expression of a negative and critical attitude directed at a victim or group of victims. Unlike basic irony, sarcasm necessitates the presence of a target, amplifying its impact as a linguistic device that conveys not only incongruity but also a sense of disrespect.

This unique combination of irony, negativity, and a specific audience distinguishes sarcasm, making it a challenging yet important task for natural language processing systems to identify. This side of sarcasm underscores the need for accurate detection to prevent misinterpretation of such remarks as literal statements. A frequent form of sarcasm, especially noticeable on platforms like Twitter, includes the contrast of positive sentiments like “love” or “enjoy” with descriptions of unfavorable situations (Riloff et al., 2013). This contrast serves as a clear indicator for computational models as they strive to differentiate between sarcastic and non-sarcastic expressions.

Sarcasm detection involves framing it as a binary classification task, predicting whether a given sentence is sarcastic or not. The challenge lies in effectively representing sarcasm within the context of a binary classification system. This computational challenge arises from the nuanced ways in which sarcasm can be expressed, often involving a deliberate contradiction between the intended meaning and the literal interpretation of the words used. For instance, in a seemingly positive statement like “Life’s good, you should get one!” the speaker is sarcastically suggesting that their life is good and fulfilling while implying that the person they are speaking to does not have a ‘good life’ and should seek to improve their situation. In that sense, failure to recognize sarcasm could lead a model to mistakenly interpret such statements as expressing positive sentiments. However, recent progress in this field has seen the creation of more advanced models, some of which utilize contextual information beyond the target text, demonstrating improved accuracy in identifying sarcasm across diverse expressions. This progress will be examined throughout Chapter 3, where we will discuss and analyze related work from the past.

2.3 Twitter

Twitter, a digital platform launched in 2006, has since become a worldwide space for social interaction. Twitter is mainly about expressing thoughts, ideas, and information in a concise but impact way through “tweets”, which are limited to 280 characters. This restriction not only encourages briefness but also promotes a form of digital communication that accommodates a diverse user base spanning from celebrities and political figures to students and homemakers.

Over its 17-year existence, Twitter has grown on a steady and significant number of users. The platform’s availability on various mediums, including web browsers and mobile applications, ensures accessibility for users worldwide. The platform’s easy access made it very popular, and users keep coming back to interact with the wide variety of content shared by others.

On a daily basis, Twitter witnesses an extraordinary volume of activity, with an astounding half a billion tweets being sent out. The mix of users on Twitter contributes to a rich collection of perspectives and opinions, creating a dynamic space where information, trends, and discussion emerge and evolve.

Beyond its role as a social hub, Twitter has proven to be a valuable resource for researchers in the field of natural language processing (NLP). For many NLP tasks such as sarcasm detection, Twitter becomes the most popular social media platform for collecting data (Băroiu and Trăușan-Matu, 2022). The platform’s vast and real-time dataset of diverse user-generated content presents a unique opportunity for studying language patterns, sentiment analysis, and linguistic phenomena. Researchers can leverage Twitter’s API to access a wealth of textual data encompassing a wide range of topics, making it an invaluable source for training and testing NLP algorithms.

Within the Twitter lexicon, hashtags (#) play a pivotal role in categorizing and indexing discussions, providing researchers with a mechanism to focus on specific themes or analyze public sentiment around trending topics. In sarcasm detection datasets, sarcastic tweets commonly contain hashtag keywords such as #sarcasm, #sarcastic, #not (Riloff et al., 2013), (Davidov et al., 2010a), (Bamman and Smith, 2015). The retweet feature, signaling the viral spread of content, enables researchers to track the dissemination of information and analyze how language spreads across the platform.

2.4 Twitter API

Twitter offers access to its data through the Twitter API, a tool that permits developers to programmatically retrieve various types of information from the platform. Twitter's accessibility through its API facilitates the extraction of large-scale datasets, enabling researchers to conduct comprehensive studies on language use across different demographics, geographic locations, and cultural contexts. This wealth of data can contribute to advancements in machine learning models, sentiment analysis tools, and language understanding algorithms.

By utilizing the Twitter API, you can also gather tweets posted in a specific language such as Turkish. This process includes querying the API for tweets that meet specific criteria, such as language settings, keywords, or hashtags commonly used in Turkish-language content. The collected data can then be employed for various purposes, including analysis, research, or integration into other applications that demand real-time or historical tweet data.

In February 2023, Twitter commercialized its API, altering its previous policy of allowing academic and research access. This shift impacted our data collection process. Consequently, we had to adjust and optimize our methodology to align with the new access constraints.

2.5 Text Classification

In the field of Natural Language Processing (NLP), considerable research effort has been devoted to tackling text classification challenges (Kowsari et al., 2019). Text classification, also referred to as text categorization, is a fundamental aspect of NLP that aims to assign labels or tags to different textual components, including sentences, queries, paragraphs, and entire documents. Notably, deep learning models have emerged as powerful performers, outperforming traditional machine learning approaches across various text classification tasks such as sentiment analysis, news categorization, question answering, and natural language inference (Minaee et al., 2021).

Text classification and document categorization systems generally involve four essential phases: feature extraction, dimension reduction, classifier selection, and evaluation. This structured approach ensures a thorough and efficient process for classifying textual data originating from various sources. These sources encompass a broad range, including web content, emails, chats, social media posts, tickets, insurance claims, user reviews, and customer service queries. Despite the valuable insights embedded in textual data, its unstructured nature presents a significant challenge, emphasizing the crucial importance of text classification methodologies.

Text classification can be carried out through either manual annotation or automatic labeling, with the latter gaining prominence as the volume of text data in industrial applications continues to expand. The importance of automatic text classification is paramount, driven by the necessity to efficiently process and derive insights from the overwhelming abundance of textual information.

Approaches to automatic text classification can be broadly categorized into two main methods: rule-based methods and machine learning (data-driven)–based methods. Rule-based methods depend on predefined linguistic rules and patterns to categorize text, while machine learning approaches utilize data-driven models trained on labeled datasets to make predictions. Each approach has its strengths and limitations, and the choice between them depends on the specific characteristics of the data and the objectives of the classification task.

As research in NLP progresses, the exploration of text classification methodologies remains a dynamic and evolving field. The ongoing development of robust models and techniques contributes not only to the enhancement of text classification systems but also to the broader domain of NLP. This fosters a deeper comprehension of language processing and the extraction of information from unstructured textual data.

2.6 BERT

In 2018, Google AI introduced BERT (Devlin et al., 2018), which revolutionized Natural language processing (NLP) and changed the field significantly. BERT, short for Bidirectional Encoder Representations from Transformers, is notable because it can understand context bidirectionally using transformers. This feature makes BERT a major milestone in NLP, greatly improving language comprehension and performing well in various NLP tasks like sentiment analysis, text classification, question answering, and overall language understanding.

Built upon the transformer architecture, BERT is a versatile and context-aware language model trained on extensive text corpora. It comes in four pre-trained versions, distinguished by the scale of their model architecture. BERT-Base (Cased or Uncased) features 12 layers, 768 hidden nodes, 12 attention heads, and 110 million parameters, while BERT-Large (Cased or Uncased) boasts 24 layers, 1024 hidden nodes, 16 attention heads, and 340 million parameters. The choice between “cased” and “uncased” depends on the task-specific preference for maintaining letter casing.

BERT uses unsupervised pre-training followed by supervised fine-tuning. Unlike a traditional Transformer, which uses both an encoder and decoder for reading input and making predictions, BERT only uses the encoder to create a language representation model. The input to the BERT encoder is token sequences converted into vectors, with three main types of embeddings: Token Embeddings (including special tokens like [CLS] and [SEP]), Segment Embeddings (showing tokens from different sentences), and Positional Embeddings (showing the position of each token in the sentence).

BERT performs exceptionally well in two pre-training tasks in NLP: masked language modeling and next-sentence prediction. In masked language modeling, BERT predicts the masked token in a sequence of words, which enhances contextual understanding. Next sentence prediction assesses whether the second sentence in a pair logically follows the first.

In this study, the aim is to use a BERT model to classify tweets as either sarcastic or non-sarcastic. The results of this classification will be carefully evaluated and analyzed, showcasing the flexibility and efficiency of BERT in the domain of text classification.

3 Literature Review

Automatic Sarcasm Detection has become a significant sub-area within Natural Language Processing research. Moreover, the field of sarcasm detection has emerged as a topic of significant interest, attracting attention from diverse research communities, including Neuroscience (Shamay-Tsoory et al., 2005), Artificial Intelligence (Muaad et al., 2022), Psychology (Ghosh and Veale, 2017), Linguistics (Skalicky and Crossley, 2018), and others. This interdisciplinary interest puts emphasis on the complexity and nuanced nature of sarcasm as a linguistic and cognitive phenomenon, assuring extensive scholarly exploration and analysis. This section draws upon insights from prior surveys (Joshi et al., 2017) and systematic reviews (Băroiu and Trăușan-Matu, 2022), aiming to provide an overview of historical developments in the field of automatic sarcasm detection. Section 3.1 is dedicated to exploring various linguistic studies pertaining to the phenomenon of sarcasm. Section 3.2 provides an analysis of the types of datasets and data collection strategies employed in previous research. In Section 3.3, attention is given to the range of methodologies adopted in different studies for tackling the challenge of automatic sarcasm detection. In Section 3.4, we explore several past shared tasks focused on sarcasm detection.

3.1 Sarcasm studies in Linguistics

One of the great efforts that deeply study sarcasm and its features comes from the field of linguistics, playing an important role in understanding this complex language phenomenon. Linguistics approaches sarcasm as a distinctly linguistic phenomenon. Throughout the years, Linguists have developed comprehensive representations and taxonomies that enhance our understanding of sarcasm, providing deeper insights into its nuances and underlying mechanism from a human-centric perspective with relatively minimal emphasis on computational perspectives.

Giora (1995) explores the concept of irony as a mode of indirect negation. The author suggests that irony or sarcasm functions as a form of negation where an explicit negation marker is absent. This implies that when one expresses sarcasm, there is an intended negation, yet it is conveyed without using a direct negation word such as ‘not’.

Ivanko and Pexman (2003) explores the significance of context in interpreting and processing literal and ironic statements. Their research proposes a six-component representation of sarcasm, encapsulated in the tuple $\langle S, H, C, u, p, p' \rangle$. This framework can be interpreted as follows: 'Speaker S produces utterance u in Context C with literal proposition p, but with the intention that hearer H comprehends the implied meaning p'.

Eisterhold et al. (2006) delves into the Non-Cooperative Principle by examining a corpus of naturally occurring ironic and sarcastic utterances. Their paper posits that sarcasm can be effectively understood through the responses it elicits. The researchers observed a range of reactions of sarcastic comments, including laughter, zero response, smiling, reciprocal sarcasm, a change of topic, a literal reply, and various non-verbal reactions.

Wilson (2006) critically examines two post-Gricean approaches to understanding verbal irony, subsequently proposing the 'Situational Disparity Theory.' This theory posits that sarcasm originates from a situational disparity between the text and contextual information.

Campbell and Katz (2012) conducted thorough research into the contextual elements employed in expressing sarcastic verbal irony. Their study critically examined whether the theoretical components traditionally considered essential for generating a sense of irony are indeed indispensable. According to their findings, sarcasm occurs along multiple dimensions, including failed expectations, pragmatic insincerity, negative tension, and the presence of a victim, offering an understanding of its complex nature.

Camp (2012) explores conventional theories of sarcasm, and argues for a more comprehensive analysis that encompasses illocutionary force and evaluative attitudes. She addresses the challenges posed by different subclasses of sarcasm to the standard implicature analysis. Camp categorizes sarcasm into four distinct types:

Propositional Sarcasm: Characterized by what appears to be a non-sentiment proposition, yet implicitly contains an evaluative sentiment.

Embedded Sarcasm: This variant features embedded sentiment incongruity, manifesting through specific words and phrases.

Like-Prefixed Sarcasm: In this form, a 'like' phrase is used to imply a denial of the proposition being made, suggesting an underlying sarcasm.

Illocutionary Sarcasm: This type involves non-verbal cues that signify an attitude contrary to what a sincere utterance would suggest. In such instances, prosodic variations are critical in conveying the sarcastic expression.

While these theories offer insightful frameworks for understanding sarcasm, they also present certain challenges in practical application, particularly in the realm of sarcasm. The primary challenges include the identification of Common Knowledge, Identification of What Constitutes Ridicule, and Speaker-Listener Context (Băroiu and Trăușan-Matu, 2022).

With technology becoming more accessible and computers widely used to address academic problems, in the subsequent sections, we will delve into automatic sarcasm detection involving computational systems and approaches. The focus will primarily be on the 'Identification of Common Knowledge' and 'Speaker-Listener Context'. These aspects are particularly relevant for computational models, as they attempt to capture and analyze the context using various techniques. This shift towards computational analysis underscores the evolving nature of sarcasm research, where traditional linguistic and theoretical approaches are being supplemented by advanced technological methods.

In that regard, the interest in sarcasm detection, while rooted in theoretical linguistics, has gained significant momentum within the field of Natural Language Processing (NLP) as well. According to the paper ny Joshi et al. (2017), the initial approaches to sarcasm detection can be traced back to Tepperman et al. (2006) work, which utilized speech-based features. This early exploration set the stage for the growing interest in sarcasm detection within sentiment analysis and NLP. In the years that followed, the inherently challenging nature of sarcasm as a linguistic phenomenon garnered attention as a research problem in more advanced domains, particularly within Artificial Intelligence (AI) and Natural Language Processing. This shift reflects the growing recognition of sarcasm detection's complexity and its importance in understanding nuanced human communication for AI systems.

By the mid-2010s, automatic sarcasm detection had achieved three significant milestones (Joshi et al., 2017), the first was the development of semi-supervised pattern extraction which enabled algorithms to identify implicit sentiments in texts where they were not explicitly stated. Following years, the rise of social media, especially platforms like Twitter, introduced the use of hashtags (e.g., #sarcasm) as a form of distant supervision. This technique provided a wealth of labeled data, greatly enhancing the accuracy of sarcasm detection algorithms. In more recent years, researchers have focused on incorporating context into these models. Recognizing that the meaning of a statement often heavily relies on its context, they began to include external information such as the broader conversation, historical interaction patterns of the users, and general world knowledge. This advancement allowed the models to more effectively determine whether a statement was sarcastic, addressing a major challenge faced by earlier approaches.

In this section, we provide a brief literature review of past works in the field of sarcasm detection, so far we have explained the complexity of the sarcasm problem and highlighted the research areas that have shown interest in this complex problem over the years.

In the following parts, we begin by detailing the datasets created specifically for sarcasm detection, describing the types of these datasets, their sources (such as Twitter, Reddit and online forums), and the methodologies employed in data collection. In the second part, we focus on the various approaches proposed for automatic sarcasm detection, including rule-based and supervised methods, and trace their chronological development. In the third part, we analyze the trends that have influenced the development of the field, underscoring the transiition from basic algorithms to the current state-of-the-art deep learning techniques. Finally, we discuss the challenges and limitations encountered in past research and propose potential solutions and directions for future work in the domain of sarcasm detection.

Furthermore, we will explore the shared tasks and collaborative efforts that have significantly contributed to the development of sarcasm detection. These shared tasks, often in the form of competitions or collaborative projects, have played a crucial role in advancing the field by providing common platforms and datasets for researchers to test and benchmark their models.

3.2 Datasets

Sarcasm detection requires large amounts of labeled data (Shmueli et al., 2020). In this section, we delve into the historical roadmap of datasets used in the research of sarcasm detection, examining how they and the principles guiding their construction have evolved.

3.2.1 Data collection Techniques

When discussing dataset creation and the methods for gathering sarcastic data, we initially refer to two main techniques: **Manual annotation** (Riloff et al., 2013) and **Distant supervision** (Bouazizi and Ohtsuki, 2015). In manual annotation, human judges analyze each utterance and label it accordingly. This process often involves crowdsourcing, making human labor an important part of the technique. On the other hand, Distant supervision automates this labeling process. For example, in the context of social media, annotators might use signals like Twitter hashtags to label data. In this scenario, the hashtag, chosen by the author of the tweet, acts as a self-determined label indicating attributes such as sarcasm. Several studies have employed the technique of hashtag-based supervision. Notable examples include the works of Davidov et al. (2010a) and Reyes et al. (2013), which explore the use of hashtags in dataset compilation and analysis.

Ozturk et al. (2021) investigates the task of irony detection in Turkish informal texts, focusing on the comparison of traditional supervised learning methods and neural network-based solutions. To facilitate this analysis, the researchers collected a dataset specifically for irony detection in Turkish. The dataset was sourced from Twitter and other microblogs/social media platforms, utilizing hashtags and other event-related identifiers to identify instances of potential irony. The data annotation process involved a group of seven native Turkish speakers, who established ground truth labels through qualified majority voting to ensure accuracy and reliability. The final dataset comprised 600 instances, evenly split between ironic and non-ironic samples, providing a balanced foundation for evaluating the performance of various detection methods.

While Distant Supervision is generally quicker and more efficient than Manual Annotation for building large datasets, it has its own set of challenges and limitations. These concerns regarding the quality of datasets generated through Distant Supervision have led some researchers to analyze this technique closely. For instance, Bamman and Smith (2015) implemented a method to enhance the reliability of their data. In their approach, tweets containing hashtag #sarcasm were categorized as positive examples of sarcasm, while those without this hashtag were assumed to be non-sarcastic, thus serving as negative examples. This method aimed to establish a more controlled and accurate dataset.

Similarly, Fersini et al. (2015) developed a dataset comprising approximately 8,000 tweets, initially labeled based on their hashtags. To further ensure the quality and accuracy of these labels, these tweets were subsequently reviewed and annotated by human judges. This additional step of manual annotation served to validate and refine the initial hashtag-based labels, highlighting the importance of combining automated and manual techniques in dataset creation for improved quality assurance.

Manual collection represents an alternative approach to dataset creation, particularly in the context of identifying sarcasm. This methodology relies on human effort to gather and report sarcastic texts. Different strategies have been employed, such as asking individuals to contribute texts that they have authored themselves, as seen in the study by Oprea and Magdy (2020), or to collect examples of sarcasm written by others, as demonstrated in Filatova (2012). However, it is important to note that both of these manual methods tend to be slower and more costly compared to distant supervision. As a result, they often yield smaller datasets. This trade-off highlights the balance between the quality and quantity of data in the field of dataset creation.

One significant drawback of earlier data collection methods was the lack of context for sarcastic utterances. In recent years, a more innovative approach to creating sarcastic datasets has emerged: **Reactive Supervision** (Shmueli et al., 2020) introduced this novel data collection method, which leverages the dynamics of online conversations. This approach effectively overcomes the limitations of previous techniques, such as manual annotation and distant supervision, by providing a richer context for understanding sarcasm. We have also adopted this technique in our research. The methodology will be explained in more detail in Section 4.1 of this research.

3.2.2 Dataset types

In this paper, we categorize datasets from some previous research into three types: short-text datasets (such as tweets), long-text datasets (for example, news articles or Reddit conversations), and datasets that include non-textual data, such as images or videos. Table 1 presents these characteristics for each dataset mentioned in previous works. As evident from the table, the majority of the datasets consist of short texts (e.g., Barbieri et al. 2014), with 26 papers focusing on this type. Of these, 8 were created through manual annotation, 16 were compiled using the distant supervision technique and the remaining two datasets employed other methods such as Reactive Supervision. Within this group, 8 were developed through manual annotation, 3 uti-

Category	**Number of Papers**	**Example Paper**
Short Text Datasets (e.g., Tweets)	26	Barbieri et al. (2014)
Long Text Datasets (e.g., News Articles, Reddit Conversations)	16	Wallace et al. (2015)
Other Datasets (Non-textual, e.g., Images, Videos)	5	Rakov and Rosenberg (2013)

Table 1: Characteristics of datasets mentioned in previous research

lized distant supervision for compilation, and the remaining 5 employed various other techniques for data collection. Five studies have contributed to the field with their non-text datasets, including images and videos (e.g., Rakov and Rosenberg 2013). Of these, three datasets were compiled through manual annotation, one employed a combination of manual and distant annotation methods, the other one (Hao and Veale, 2010) used a different technique.

As mentioned earlier in Section 2.3, Twitter is the most popular source for collecting sarcastic data, and there are several reasons for this. Firstly, the availability of the Twitter API facilitates easy access to a vast amount of data. Twitter's popularity, particularly among English speakers, also contributes to its widespread use as a data source. The platform's format, which encourages users to express their thoughts and feelings in concise text, is particularly suitable for studying sarcasm. Additionally, before its commercialization in early 2023, the Twitter API enabled straightforward data extraction through distant supervision of hashtags. The techniques for obtaining labels on Twitter are consistent with those mentioned previously. Notable examples of research utilizing manual annotation techniques on Twitter data include works by Riloff et al. (2013) and Maynard and Greenwood (2014).

Twitter is not the only social media platform utilized by researchers as a data source for sarcasm detection and analysis. Other platforms have also been explored for this purpose. For instance, Wallace et al. (2014) used Reddit comments as a source of short sarcastic texts for their research. Similarly, Davidov et al. (2010b) turned to Amazon product reviews as a unique and rich source of data.

Several datasets have been created for the purpose of sarcasm detection in longer text formats, beyond brief tweets or Reddit comments. For instance, Davidov et al. (2010b) compiled a substantial collection of 66,000 Amazon reviews. Reyes and Rosso (2012) focused on products that suddenly received a wave of sarcastic reviews. In a similar vein, Reyes and Rosso (2014) developed a dataset encompassing movie and book reviews, as well as news articles. Lukin and Walker (2013) introduced the Internet Argument Corpus, a dataset of discussion forum posts tagged with various labels, including sarcasm. Taking a different approach, Liu et al. (2014) gathered data from a range of sources such as Amazon, Twitter, Netease, and Netcana.

In addition to traditional datasets, there are innovative collections that encompass a variety of text types, extending beyond brief tweets or extensive movie reviews. Spoken and multimodal datasets, such as those analyzing call center conversations and TV series dialogue, serve as prime examples of these innovative collections. Tepperman et al. (2006) conducted an analysis on the usage of the phrase 'yeah right' within 131 call center conversations. Joshi et al. (2016a) explored a manually annotated dataset from the TV series 'Friends', where each line in a scene was labeled as either sarcastic or non-sarcastic. Hao and Veale (2010) concentrated on discerning which similes are used sarcastically. They began by searching the web for instances of the pattern '* as *', resulting in a compilation of 20,000 unique similes, each subsequently classified as either sarcastic or not.

3.3 Approaches

The approaches used to tackle automatic sarcasm detection have evolved throughout the years. The transition from rule-based approaches to machine learning-based and most recently deep learning models.

3.3.1 Ruled Based Approach

Rule-based methods for detecting sarcasm automatically focus on identifying sarcasm through specific signs, which are defined by rules and heuristics that look for sarcasm indicators. According to Maynard and Greenwood (2014), the sentiment of hashtags is a crucial clue for sarcasm. They suggest that if the sentiment conveyed by a hashtag in a tweet contradicts the rest of the tweet's content, the tweet is likely sarcastic. Bharti et al. (2015) developed two rule-based classifiers. The first uses a parse-based lexicon generation algorithm to create sentence parse trees and pinpoint sentiment-bearing situational phrases. A sentence is considered sarcastic if it contains a negative phrase within a positive context. The second classifier focuses on identifying hyperbole, such as "I'm so hungry, I could eat a horse," by detecting the co-occurrence of interjections and intensified.

3.3.2 Machine Learning Approach

In the early stages of sarcasm detection, rule-based methods were predominant, but the field has gradually shifted towards machine learning (ML) techniques. These ML approaches to sarcasm detection employ a range of features and algorithms, with the bag-of-words model being a common technique. However, other methods are also in use. For instance, Davidov et al. (2010a) introduced features based on patterns, which are effective in identifying specific sarcasm-indicative patterns. These patterns are extracted from a large corpus of sarcasm-tagged data. The strength of these pattern-based features lies in their ability to identify three types of pattern matches: exact, partial, and none. Moreover, words within these patterns are classified as High-Frequency Words (HFWs) or Content Words (CWs), depending on how often they appear in the corpus. Ibanez et al. (2011) employed features based on sentiment lexicons. Additionally, pragmatic features like emoticons and user mentions are also utilized in sarcasm detection. Lunando and Purwarianti (2013) incorporated aspects such as negative sentiment and the frequency of interjections, alongside word context, in their research. They employed supervised machine learning techniques like Naive Bayes, Maximum Entropy, and Support Vector Machine (SVM) for classification, due to their high accuracy rates.

3.3.3 Deep Learning Approach

In recent years, there has been a noticeable rise in the application of deep learning methodologies in the realm of automatic sarcasm detection. A pioneering example of this trend is the work by Joshi et al. (2016b), who introduced the utilization of word embedding similarity as a feature in sarcasm detection. Their approach involves enriching the feature set with similarities derived from the most congruent and incongruent word pairs. This enrichment is crucial, as they found that solely relying on these features is insufficient for effective sarcasm detection. Their findings demonstrate a significant performance improvement, highlighting the importance of feature augmentation in this domain.

Ghosh and Veale (2016) utilized an integrated approach that includes a Convolutional Neural Network (CNN), followed by a Long Short-Term Memory (LSTM) network. In their study, they conducted a comparative analysis with a recursive Support Vector Machine (SVM). The results of this comparison showed a significant improvement in performance when using the deep learning architecture, highlighting the effectiveness of their proposed method in the relevant domain.

Ren et al. (2020) introduced a novel multi-layer neural network, specifically trained on emotion-semantics, for sarcasm detection. Their model incorporates a dual-layer memory network architecture. The initial layer captures emotional nuances in individual phrases, while the subsequent layer identifies contrasts between these nuances and the sentence's overall context. Additionally, they integrated a modified Convolutional Neural Network (CNN) to boost the memory network's performance, without requiring additional domain-specific data. Their approach's effectiveness was empirically validated through rigorous testing on datasets like the Twitter dataset and the Internet Argument Corpus.

Other deep learning techniques have been utilized in automatic sarcasm detection, especially for analyzing the context surrounding sarcastic utterances. By studying the impact of context, researchers strive to achieve more promising results in sarcasm detection.

Jena et al. (2020) tackles the challenging task of accurately identifying sarcasm in conversations. The authors highlight the inherent difficulty in classifying an utterance as sarcastic or non-sarcastic in isolation, emphasizing the crucial role of contextual information in determining the sarcastic nature of a sentence. In their paper, the proposed model, C-Net, is designed to effectively utilize the conversation context of an utterance in a sequential manner to capture sarcastic utterances. The approach achieves F-1 scores of 75% on the Twitter dataset and 66.3% on the Reddit dataset, which were presented in the Sarcasm Detection shared task as part of the Second Workshop on Figurative Language Processing, colocated with ACL 2020. This paper's argument aligns with the evolving trends in sarcasm detection research, which increasingly recognizes the importance of contextual information in accurately identifying sarcasm.

Similar to our study, Ghosh et al. (2018) explored sarcasm detection in social media posts, focusing on contextual conversation rather than analyzing tweets in isolation. They gathered data from three distinct sources. The first was a subset of the Internet Argument Corpus (IACv2),[2] known as The Sarcasm Corpus V2,[3] which includes posts tagged as either sarcastic or non-sarcastic. The second source utilized was the Self-Annotated Reddit Corpus (SARC) dataset introduced in Khodak et al. (2017), characterized by posts marked as sarcastic with a "/s" at the end. Lastly, they compiled a corpus of tweets from Twitter, where tweets were self-labeled for sarcasm

[2] https://nlds.soe.ucsc.edu/iac2

[3] https://nlds.soe.ucsc.edu/sarcasm2

using specific hashtags, collected through Twitter's developer APIs.

Kumar and Anand (2020) also presents a comprehensive study on the role of context in sarcasm detection within social media conversations. The study explores the significance of context in identifying sarcasm, acknowledging the challenges in identifying sarcasm without considering the surrounding dialogue. The method leverages various pre-trained language models and transformer architectures as BERT, RoBERTa and spanBERT to investigate the effectiveness of different deep-learning techniques for sarcasm detection with context. The paper introduces a novel architecture comprising LSTM and Transformers, offering insights into the potential of hybrid models for enhancing sarcasm detection in contextual conversations.

Babanejad et al. (2020) developed two models for detecting sarcasm, modifying the BERT architecture to include both affective and contextual elements. These models, named ACE 1 and ACE 2 (where ACE stands for Affective and Contextual Embeddings), showed promising performance. In comparative evaluations, ACE 1 outperformed its counterpart, demonstrating higher efficiency with an F1 score of 0.8457 on the SemEval 2018 Twitter dataset (Van Hee et al., 2018b) and 0.9314 on the IAC dataset (Abbott et al., 2016).

Razali et al. (2021) presents a comprehensive approach to sarcasm detection in tweets by combining deep learning extracted features with carefully handcrafted contextual features. This study contributes to the field of automatic sarcasm detection by supplementing the traditional rule-based techniques with advanced deep learning methods as It integrates contextual clues and linguistic markers. The study also addresses the challenge of identifying sarcasm, a form of language that often involves a positive utterance with underlying negative intention and emphasizes the importance of contextual understanding in detecting sarcasm, as tweets often contain subtle clues such as hashtags and hyperboles to convey sarcastic intent. The integration of deep learning and contextual handcrafted features represents a significant advancement in the field, offering a more nuanced and effective approach to sarcasm detection in social media discourse

Du et al. (2022) address the challenge of sarcasm detection in social media, highlighting the limitations of traditional sentiment analysis methods in capturing the indirect and emotional nature of sarcasm. They argue that context, particularly the sentiments of replies and user expression habits, plays a crucial role in sarcasm detection. To overcome this, the authors propose a novel neural network model that integrates sentimental context and individual expression habits. The model aims to enhance sarcasm detection by emphasizing the significance of contextual information and user-specific expression patterns, thereby contributing to the field of automatic sarcasm detection and advancing research in natural language processing and sentiment analysis. The results showed that the proposed approach can significantly improve the performance of sarcasm detection tasks.

While not the primary focus of the current study, multi-modal approaches have gained popularity in automatic sarcasm detection tasks in recent years.

Qiao et al. (2023) tackle the challenging task of detecting sarcasm in multi-modal content, which combines both text and image data. They introduce the Mutual-enhanced Incongruity Learning Network (MILNet), a novel approach that incorporates a local semantic-guided incongruity learning module and a global incongruity learning module. This integration enables MILNet to effectively capture both inter and intra-modal incongruities within the same context, thereby enhancing sarcasm detection. The authors underscore the relevance of their work in the context of the increasing prevalence of multi-modal communication on social media platforms, making their contribution timely and significant to the field of automatic sarcasm detection.

3.4 Shared Tasks

Shared tasks in conferences allow a common dataset to be shared across multiple teams for a comparative evaluation (Joshi et al., 2017). Several shared tasks focusing on detecting sarcasm have been carried out previously. One such task is by Ghosh et al. (2015), a part of SemEval-2015, which is centered on analyzing sentiment in figurative language. The task's organizers presented participants with a dataset containing statements that were either ironic or metaphorical, each categorized under positive, negative, or neutral sentiments. The challenge for participants was to effectively identify the sentiment's polarity, especially in figurative expressions like irony. The methods used by the teams varied, including the use of effective resources and character n-grams. Notably, the most successful team employed a combination of four lexicons, with one developed through automation and the others created by hand.

Van Hee et al. (2018b) hosted a SemEval task centered on identifying irony in individual Twitter posts. This task involved not only distinguishing whether a tweet was ironic or not (binary classification) but also categorizing the irony into specific types, such as verbal, situational, or other forms, through a more detailed multi-classification system.

The Second Workshop on Figurative Language Processing, featuring a sarcasm detection shared task (Ghosh et al., 2020). This task focused on understanding how conversational context influences the detection of sarcasm. It encompassed two different types of social media sources for its two distinct tracks: content from a microblogging site like Twitter and posts from an online discussion forum such as Reddit.

4 Methodology

The main concern of this study is Irony Detection in Turkish and investigating the effect of context on the performance of classification models. The key task of such models is to classify these texts as either ironic or non-ironic. Given the limited number of Turkish Irony datasets, for this study, it is essential as the first step to collect the data, where we utilize a significant Turkish dataset obtained from Twitter via the Twitter API. The data was collected using former Twitter academic access through the Twitter Streaming API from March 2018 to until February 2023 when Twitter stopped the academic access. The Twitter stream was queried using the most frequent Turkish words and filtered using Twitter's language identifier to only include tweets identified as Turkish. The data contains a large part of tweets in Turkish during this period. In total, the dataset comprises 2.9 billion tweets, with an average of 1.6 million tweets per day and 50 million tweets per month. The entire dataset contains approximately 35.6 billion tokens, averaging about 12 tokens per tweet.

Unlike common practices in many Natural Language Processing projects on different tasks such as text classification, and sentiment analysis, we deliberately avoid extensive preprocessing of the data. We believe that excessive preprocessing may detract from the inherent spontaneousness and novelty of the text, potentially reducing its reliability to real-world linguistic nuances. Therefore, our goal is to make minimal changes to the collected corpus, preserving its inherent authenticity to enable a more accurate assessment of model performance under conditions that reflect real-world text environments.

We use a dual methodology to obtain ironic and non-ironic tweets. While sarcastic data undergo automatic annotation named Reactive Supervision (Shmueli et al., 2020), non-sarcastic counterparts receive careful manual annotation, ensuring the integrity and accuracy of our dataset. After annotation, we statistically analyze the data, uncovering characteristics embedded within our Turkish sarcasm dataset. This analytical exercise aligns with one of the contributions of the study: creating a reliable Turkish Irony/Sarcasm Dataset.

Leveraging our dataset, we make minor adjustments to our model architecture to optimize performance. The final phase of our study is the classification stage, where our analysis relies on the Transformer model, BERT. Our objective is to analyze and understand the complex aspects of irony as they are manifested in the Turkish language.

In this chapter, we outline the methodology of our study. Section 4.1 describes the collection of our datasets, which involved leveraging a larger, pre-existing Turkish Twitter dataset acquired through the Twitter API. Section 4.2 details the annotation process applied to these raw datasets to distinguish between sarcastic and non-sarcastic tweets. In Section 4.3, we present statistics on the number of tweets processed for the Turkish Sarcasm Corpus (TSC). Finally, Section 4.4 discusses the fine-tuning of BERT transformer models for a binary classification task.

4.1 Data Collection

Within the spectrum of data collection methodologies explained in the Literature review part (see Section 3), we adopt a relatively novel approach known as the Reactive Supervision methodology (Shmueli et al., 2020). The method's appeal lies not only in its ability to avoid the laborious manual annotation process but also in its capacity to capture the contextual text surrounding sarcastic expressions. By using Reactive Supervision, we aim to explore the complex relationship between context and irony more deeply, which has significant implications for our classification effort.

Before delving into our data collection methodology, it is important to highlight the crucial role of context in our classification task. Understanding context might be essential for recognizing the complexities of identifying sarcastic expressions within a given corpus. By disclosing the complex web of contextual cues, we can better understand the irony and sarcasm thereby enhancing the effectiveness of our classification framework.

For the effectiveness of computational systems, especially in areas like sentiment analysis and dialogue systems, It becomes crucial to autonomously detect irony, both from the author's and the reader's perspective, to avoid potential misinterpretations (Shmueli et al., 2020). Research on the distinction between author-intended sarcasm and reader-perceived sarcasm was notably advanced by Oprea and Magdy (2019) within the domain of sarcasm detection endeavors.

Tweet Type	User	Tweet Example
Sarcastic Tweet	User C	"The app we use for work emails is not working. I feel terrible about this!"
Oblivious Tweet	User B	"Not your fault. Do not feel guilty."
Cue Tweet	User A	"She was just being sarcastic!"

Table 2: Examples of perceived tweet types in Reactive Supervision method

One of the contributions of this study is to create a reliable Turkish sarcasm corpus using social media discussions, following a method similar to the one outlined in a data collection technique. Due to compatibility issues with the terminology used in the original methodology, where the English dataset was named Sarcasm, Perceived and Intended, by Reactive Supervision (SPIRS), we use the term Sarcasm instead of Irony. The Reactive supervision method, as introduced in Shmueli et al. (2020), utilizes the dynamics of online conversations to overcome the limitations of existing data collection techniques such as manual annotation and distant supervision. Although we aimed to use a similar approach, we faced some external limitations that stopped us from following the precise "4-step pipeline" framework mentioned in the original paper.

Reactive Supervision methodology is focused on three kinds of tweets that appear in typical Twitter conversations. Cue Tweets are reply tweets that highlight sarcasm in a prior tweet (e.g., "Don't worry, it's sarcasm!"). Oblivious tweets are responses to sarcastic tweets made without realizing they are sarcastic. Sarcastic Tweets are tweets that represent sarcastic text in a Twitter thread. To better understand this, we can look at the examples from the original paper (Shmueli et al., 2020) in Table 2, which depicts a common Twitter interaction: User C posts a sarcastic tweet. Unaware of C's sarcastic intent, B responds with an oblivious tweet. Then, User A clarifies things for B with a cue tweet, saying, "She was just being sarcastic!". Given that A addresses B but mentions the sarcastic author (C) in the third person as "She", it is evident that C is the author of the perceived sarcastic tweet. Similarly, Table 3 illustrates how a first-person cue, like "I was just being sarcastic!", can definitively label intended sarcasm. For a comprehensive understanding and further contributions related to this data collection technique, you can check the official GitHub repository and the paper from the link made publicly accessible.[4]

In February 2023, Twitter implemented a new monetization strategy for its API, marking the end of the era of free access that had been available previously. As a

[4] https://github.com/bshmueli/SPIRS

Tweet Type	**User**	**Tweet Example**
Trigger Tweet	User C	“Just watched Forrest Gump. Great film!”
Sarcastic Tweet	User A	“So Tom Hanks can act! Who knew???”
Oblivious Tweet	User B	“Literally everyone!!!”
Cue Tweet	User A	“I was just being sarcastic!”

Table 3: Examples of intended tweet types in Reactive Supervision method

result of this significant change, we find ourselves compelled to reassess our approach to data collection, particularly for gathering sarcastic content.

We have opted for a less automatic version of the Reactive supervision method for our data collection efforts, which involves less utilization of the Twitter API during the data collection process. This approach, which we had initially employed for compiling sarcastic data, involves a systematic and methodical examination of Twitter content, particularly focusing on the nuances of language and user interactions.

To make sure our dataset is comprehensive, we decided to use our method for the whole time that the database covers (From March 2018 to February 2023). This means we will look at and collect data from each of the sixty months in the dataset. With this full approach, we want to see all the different Twitter activities and trends. This will help us understand sarcasm better in the Turkish Twitter community.

It is worth noting that the choice of ‘ironi’ (Irony) over the literal translation ‘iğneleme’ (sarcasm) was made due to their frequent interchangeability and the broader usage of the former term within the context of social media. This decision allows us to capture a wider range of potentially sarcastic content within the Turkish Twitter Community.

In the next step of our analysis, we employ a rule-based classifier to identify the personal suffixes of the verb “yapmak” within the mentioned search query “ironi yap”. Unlike English, where pronouns determine the personal markers (e.g. “I was being sarcastic” for 1st person and “He was being sarcastic” for 3rd person), Turkish operates differently as an agglutinative language. In Turkish, it is the suffix that plays a crucial role in specifying the personal marker associated with the verb.

For example, in the phrase “ironi yaptım”, the suffix “-tım” indicates the 1st person singular, and in “ironi yapmış” (he/she was being sarcastic), “-mış” reveals that it is the 3rd person singular form.

To accurately identify these personal suffixes, we utilize a rule-based classifier that examines the structure of the verb phrase and identifies the specific suffixes associated with each person. This classifier relies on linguistic rules and patterns inherent to the Turkish language to determine the correct personal marker for each verb form.

Following the classification of cue tweets, our next step involves filtering out cue tweets that are not classified into any person category. From the remaining cue tweets, we extract the IDs of oblivious tweets, which represent unsuspecting replies to the sarcastic tweets. This extraction is done using the ‘in_reply_to_status_id’ column presented in the cue tweet dataset.

In the fourth and final step of our pipeline, we aim to identify the sarcastic tweets based on the responses they receive, considering the nuances of Turkish Language usage and social interaction on Twitter. This steps involves matching Twitter IDs to a list of texts corresponding to the sarcastic tweets, which allows us to discern the context and the intent behind each tweet accurately. To achieve this, we implement two distinct strategies to identify sarcastic tweets.

Firstly, we trace back to Sarcasm signaled by the first-person cue, for example when a tweet’s author explicitly signals sarcasm using a first-person cue, such as “ironi yaptım” (I was being sarcastic), we label the author’s preceding tweet, in the same thread as the sarcastic comment. This straightforward, approach allows us to directly attribute sarcasm to the preceding tweet, providing unambiguous annotations.

Secondly, we evaluate context and intervening replies for indirect sarcasm in cases where sarcasm is inferred from second or third-person cues, we evaluate both the cue tweet and any intervening ‘oblivious’ replies that occur between the cue and the original sarcastic tweet. This method considers the conversational context and examines the sequence of replies to identify the underlying sarcasm accurately. By analyzing the thread of replies and considering the linguistic and contextual cues, we can discern the subtle nuances of sarcasm and attribute it to the appropriate tweet within the conversation thread.

This context-aware approach is crucial for our study’s purpose as it ensures that sarcastic tweets are accurately identified and contextualized within the broader con-

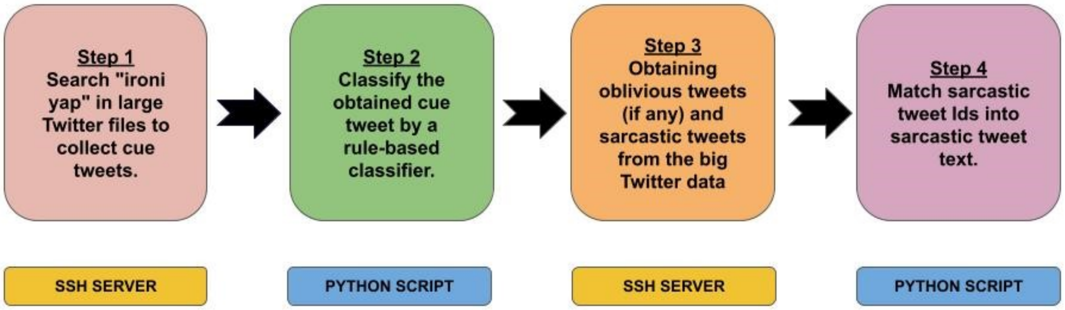
Step 1
Search "ironi yap" in large Twitter files to collect cue tweets.
Step 2
Classify the obtained cue tweet by a rule-based classifier.
Step 3
Obtaining oblivious tweets (if any) and sarcastic tweets from the big Twitter data
Step 4
Match sarcastic tweet Ids into sarcastic tweet text.
SSH SERVER
PYTHON SCRIPT
SSH SERVER
PYTHON SCRIPT

Figure 1: The 4-step pipeline used in the data collection process

versation. Additionally, it addresses challenges faced by external annotators who may struggle to understand the conversational context due to cultural and social differences, as highlighted in previous studies (Joshi et al., 2016a). For clear understanding, we provide Figure 1 below which shows the adopted version of the "4- step pipeline" in our context.

In the original study (Shmueli et al., 2020), Fetch is the initial phase, and the Twitter API is utilized to gather cue tweets by searching for the query "being sarcastic". However, in our case, we essentially searched for the term "ironi yap." in the big Twitter dataset. This dataset was collected using the Twitter streaming API. We are not actively using the Twitter API service now but are working with a dataset gathered using it in the past.

Classification is the next step in which a rule-based classifier prioritizing precision is employed. This classifier categorizes cues as 1st-, 2nd- or 3rd- person based on the referenced pronoun (e.g., I, you, he/she). If the cue cannot be distinctly categorized (for instance, when a pronoun is absent, the cue includes multiple pronouns, or there negation words), it is labeled as 'unknown' and subsequently disregarded. In our case with the Turkish dataset, we classify cues based on the suffixes of the verb "yapmak". Similar to the original procedure, we disregard cases without any suffixes and those with negation suffixes (-ma, -me) in the verb itself

After classification, the traverse step utilizes the Twitter Lookup API to trace back the thread, beginning from the cue tweet. It consistently fetches the parent tweet until it reaches the root or original tweet of the thread. This implementation is not feasible in our case since we have no access to Twitter API. However, this implementation is not feasible in our case since we do not have access to the Twitter API. Instead, we traverse preceding tweets before cue tweets manually by searching for 'in_reply_to_status_id' in the large Twitter dataset until we find the sarcastic tweet. Since these big Twitter datasets were gathered a while ago and frozen with no updates, many of these tweets with corresponding IDs might be absent there, so the number of oblivious and sarcastic tweets we obtained is inevitably less than the cue tweets we gathered in the beginning. We will address this data loss when discussing data statistics in the next sections.

Finally, in the match phase, the sequence of authors in the thread is compared to the relevant regular expression. Threads that do not match the pattern are disregarded. If a match is found, the sarcastic tweet is pinpointed and stored, accompanied by the cue tweet and when applicable, the eliciting and oblivious tweets.

In essence, In our modified approach to Reactive supervision (Shmueli et al., 2020) we managed to gather three types of tweets along with their metadata. We employ a pipeline similar to the strategy previously outlined. However, our adaptation is not as automated as the original version and does not rely on tools provided by Twitter services.

After obtaining the three types of tweets in a typical Twitter thread, we take one step further in the thread and additionally retrieve the tweets to which these original sarcastic tweets are posted as replies. Essentially, these newly obtained tweets serve as trigger tweets and are most likely victims of these sarcastic utterances. Since our main focus is the effect of context on automatic irony identification systems, the effect of these trigger tweets on our transformer model will be explicitly studied in the next sections.

In our pursuit of non-sarcastic utterances, we employ a multi-step methodology to ensure systematic and comprehensive gathering of relevant data. Unlike the method used to identify sarcastic utterances, which depended on cue tweets and thread context, our strategy for identifying non-sarcastic utterances involves a different set of procedures customized to the data's characteristics.

We began by gathering user IDs linked to the sarcastic utterances collected through the methodology described in the preceding sections. These user IDs formed the basis for our subsequent endeavors to identify non-sarcastic utterances within the Twitter dataset.

Next, we categorized these user IDs based on the month/year dataset in which they were originally included. This categorization helped us organize the user IDs according to the temporal dimension, enabling targeted searches within specific timeframes.

With the categorized user IDs in hand, we then searched for random tweets within the monthly datasets corresponding to the respective timeframes. For example, when searching within the April 2020 dataset, we specifically considered user IDs from which we had obtained sarcastic tweets within the same monthly dataset, namely April 2020. By restricting our search to user IDs associated with sarcastic utterances within the specific monthly dataset under consideration, we aimed to maintain consistency and relevance in our data collection process.

Following the initial collection of additional tweet samples from the authors of sarcastic tweets, our next step involved a thorough process to ensure the relevance and contextual coherence of the gathered data. We began by excluding samples that lacked any context tweet to which they were replying. This filtering step was crucial for preserving the integrity and validity of the dataset, ensuring that each tweet sample was situated within a meaningful conversational context.

After the exclusion process, we proceeded to gather context tweet IDs for specific months. This involved identifying the tweets that served as the contextual setting for the selected tweet samples, providing the necessary context for understanding within a meaningful conversational context.

Once we compiled the context tweet IDs for each month, we conducted a thorough search within our large dataset to find and retrieve these tweets. This search ensured that each context tweet was accurately identified and included in the dataset.

The ultimate goal of this process was to compile a dataset similar in structure to our sarcastic dataset, albeit comprising context tweets instead. We aimed to improve the BERT model by adding tweets that provide context. This helps BERT understand sarcasm and irony better, especially in Turkish tweets. By doing this, we aim to better understand language and how people talk online.

Finally, we ended up with two distinct datasets representing sarcastic and non-sarcastic tweets. The sarcastic dataset was gathered using a self-annotation process facilitated by the reactive supervision methodology. Meanwhile, the non-sarcastic tweets were random selections from sarcastic authors and were manually annotated by human annotators. These datasets laid the groundwork for our ambitious endeavor of creating a dependable resource known as the Turkish Sarcasm Corpus.

The Turkish Sarcasm Corpus represents a unique compilation of tweet samples occurring within the conversational context of Turkish Twitter discourse. It includes different kinds of linguistic expressions, social interactions, and cultural differences, captured in real-time without any interpretation or preprocessing from our end.

By combining the sarcastic and non-sarcastic datasets, we sought to create a comprehensive repository that reflects the diverse spectrum of communication styles and linguistic phenomena observed in online conversations. This unified corpus serves as a valuable resource for researchers, linguists, and practitioners interested in studying sarcasm and irony within the Turkish Language context.

Through careful analysis and annotation processes, we ensured the integrity and reliability of the Turkish Sarcasm Corpus, empowering researchers to explore and analyze the intricacies of digital communication with confidence and precision. To further assess the reliability and utility of our dataset, we conducted several experiments, which we will elaborate on in the subsequent sections. These experiments will aid us in gaining a better understanding of irony and sarcasm within the digital landscape and evaluating how well they are understood by computer systems such as Transformers.

4.2 Data Annotation

We automatically identify sarcastic tweets using the Reactive Supervision method (Shmueli et al., 2020). This approach leverages the natural flow of conversations on Twitter, which we described in the previous section. Essentially, we use specific tweets, known as cue tweets, as markers to spot potential sarcasm in a conversation.

For the annotation of non-sarcastic samples we gathered by multi-step methodology (see section 4.1), we used the manual annotation using humans to annotate these examples. we annotated approximately 2000 different samples 20% of which were found to be sarcastic which was further excluded in the last version. The remaining non-sarcastic tweet samples were then gathered by other sarcastic tweets previously annotated by the reactive supervision technique.

In data annotation, it is beneficial to have multiple annotators label the same training instances to verify the accuracy of the labels. By having multiple annotators review the same data, we can calculate the inter-observer agreement or IAA. IAA is crucial for determining the clarity of annotation guidelines, the consistency with which annotators understood these guidelines, and the reproducibility of the annotation task. It is an essential component in validating and reproducing classification results. IAA also accounts for the expected chance agreement that might occur when individuals annotate instances. The measures considering expected chance agreement are:

Cohen's kappa: This is used when two annotators classify each instance into a category. (McHugh, 2012)

Fleiss' kappa: This applies when each instance is classified into a category by multiple annotators. (Fleiss, 1971)

To ensure the quality of our manually annotated non-sarcastic dataset, we utilized inter-annotator agreement (IAA) as a metric. This metric measures the consistency of annotation decisions across different annotators for a given category. Each annotator received a portion of these tweets and the same set of annotation guidelines. Two of the annotators (Annotator 1 and Annotator 2) had expertise in Computational Linguistics, whereas Annotator 3 did not have a background in linguistics. However, all were native speakers of Turkish, ensuring a clear understanding of the texts' meanings and contexts.

The Cohen's kappa score, an indicator of agreement, was 0.64 between Annotator 1 and Annotator 2, suggesting substantial agreement according Kappa statistics interpretations defined in Viera et al. (2005). This level of concordance likely stems from their shared linguistic background, aiding in identifying the subtleties of ironic statements in the text which can already be tricky for most human annotators. On the other hand, Annotator 3, lacking a linguistic background, showed lower Cohen's kappa scores of 0.37 and 0.52 when paired with Annotator 1 and Annotator 2, respectively. These scores reflect a lesser degree of understanding and discernment of sarcasm in the utterances. The Fleiss' kappa score (Fleiss, 1971) scores for all three annotators is 0.30 which indicates a fair agreement.

Type	Count
Cue	103 483
Oblivious	27 243
Sarcastic	13 743
Context(Tigger)	2157
Cue/Oblivious/Sarcastic	5923
Cue/Oblivious/Sarcastic/Trigger	1025

Table 4: Breakdown of tweet counts by four types and conversation threads in our data collection process

4.3 Data Statistics

In this section, we statistically examine our Turkish dataset and the original English dataset described Shmueli et al. (2020) obtained through the Reactive Supervision Method, as detailed in section 4.1.

Table 4 shows the obtained tweet counts by using our data collection pipeline. The last two rows include the number of conversation threads with respective tweet types. As we discussed, our data collection pipeline began with the collection of Cue tweets containing the phrase “ironi yap,” which translates to “being sarcastic/ironic,” resulting in a total of 103,483 tweets. Next, we extracted 27,143 oblivious tweets from a large Twitter dataset. These tweets are responses to sarcastic ones without recognizing the sarcasm. By analyzing these oblivious tweets, we identified potential sarcastic tweet IDs and then located them in the extensive Turkish Twitter dataset comprising millions of tweets. This process yielded 13,743 sarcastic tweets prior to any processing or quality assessment. Furthermore, using these IDs, we were able to extract 2,157 context tweets from the larger dataset. The final step involved concatenating cue, oblivious, sarcastic, and context tweets from the same conversation threads. Ultimately, we compiled 1,025 complete conversation threads, which serve as our automatically annotated sarcastic dataset for model training purposes. The loss of data in our data collection pipeline happened because Twitter streaming API does not stream all tweets but only a sample. They were initially created using Twitter's API before it was commercialized in February 2023. Since then, these files have not been updated, there was uncertainty about whether they include every piece of relevant tweet data we require for our analysis. Also, the method in Shmueli et al.'s paper (2020) initially found 65,000 tweets but ended with only 15,000 sarcastic tweets. The varying degrees of data loss experienced in both techniques can be attributed to the availability of the Twitter API and the unchanging state of our extensive Twitter files.

4.4 Classification

After gathering and annotating the datasets, we progressed to the text classification phase. Text classification is a common NLP task that assigns a label or class to text. For that purpose, we utilized the Turkish BERT model (Schweter, 2020), leveraging a transformer-based model pre-traned on a large corpus of Turkish text. This model, based on transformer architecture and pre-trained on a substantial corpus of Turkish text, was fine-tuned for binary classification, specifically to distinguish between sarcastic and non-sarcastic comments.

We adopted the Adam optimizer with a learning rate of 2e-5, aligning with standard practices for fine-tuning BERT models. Sparse categorical cross-entropy served as our loss function, fitting for tasks with sparse labels like ours. We focused on accuracy and various F1-scores like F1-Binary and F1-Macro to evaluate our model's effectiveness.

Our training process involved batches of 512 samples over 10 epochs. To ensure we captured the most effective version of the model, we implemented checkpoints at regular points during training. This allowed for the recovery or analysis of the model at these specific stages.

As a pre-requirement for such BERT models, the textual data was converted into a format suitable for training using the aforementioned tokenization process. This was then fed into the model as a TensorFlow dataset object. We then compiled the model with our chosen optimizer, loss function, and evaluation metrics, and conducted the training on the preprocessed dataset.

For computational resources, we utilized Google Colab Pro+ GPUs, specifically the A100 GPU for each model. The duration of the training varied between 5 to 15 minutes, contingent on the dataset size (see section 5.1).

5 Experiments

In our research, we conducted two experiments to assess the role of conversational context in accurately detecting irony in social media. In the first experiment, we trained our BERTurk Transformer models (as explained in section 4.4) using four variants of the Turkish Sarcasm Corpus (TSC) and the external dataset IronyTR, without incorporating any external contextual information. For the second experiment, we focused solely on each version of the TSC, as the IronyTR dataset does not include supplementary contextual text, and hence was excluded. By comparing various evaluation metrics across these experiments, we aimed to determine the true influence of contextual information in improving the performance of our models.

In the following section, we will conduct a detailed analysis of each dataset used to train our model. This examination aims to understand the impact of these datasets on the model's ability to discern between sarcastic and non-sarcastic expressions. We will present and discuss various performance metrics, including accuracy and F1 scores. This will help us measure the influence of both the quality and quantity of training data on the efficacy of pre-trained transformer models. Further, in Section 5.4, we will evaluate and compare the performance of models trained on identical datasets, with and without the inclusion of contextual tweets. This comparison is expected to provide valuable insights into our primary research question: Does incorporating contextual tweet information during training enhance the model's proficiency in detecting sarcasm in social media?

5.1 Datasets

In all our experiments, we utilize four distinct versions of the Turkish Sarcasm Corpus. This section aims to provide an overview of how these datasets differ both qualitatively and quantitatively. We will delve into the specific characteristics of each version, detailing aspects such as the size of the datasets, the diversity of content, the annotation methodologies employed, etc. Additionally, we will explore the balance of sarcastic versus non-sarcastic utterances in each dataset and discuss how these proportions may influence the performance of our models. In addition to the Turkish Sarcasm Corpus, our study incorporates the IronyTR dataset as an external resource. This section will offer a concise yet thorough overview of IronyTR, focusing on its composition, unique characteristics, and relevance to our research.

Name	**Sarcastic**	**Non-sarcastic**	**Unique tweets**	**Context Tweets**
TSC_v1	1025	1032	No	Yes
TSC_v2	1025	1560	No	Yes
TSC_v3	300	300	Yes	Yes
TSC_v4	1007	500	Yes	Yes
IronyTR	300	300	Yes	No

Table 5: Comparison of different Twitter Sarcasm Corpus (TSC) versions and the IronyTR dataset

TSC_v1 (Turkish Sarcasm Corpus version 1): This version consists of 1025 sarcastic tweets, which were compiled employing a reactive supervision methodology (Shmueli et al., 2020) and 1032 random non-sarcastic tweet samples manually annotated from the tweets of same sarcastic authors. Each tweet in this dataset, both sarcastic and non-sarcastic, is paired with a contextual tweet to which the primary tweet was a response.

TSC_v2 (Turkish Sarcasm Corpus version 2): Similar to TSC v1, this version includes 1025 sarcastic tweets collected through the same methodology. However, it features a different set of 1560 non-sarcastic tweets, also randomly selected from the authors of the sarcastic tweets and manually annotated. As with TSC v1, each tweet is associated with a contextual tweet.

TSC_v3 (Turkish Sarcasm Corpus version 3): The third version of the dataset represents a more balanced collection, consisting of 300 sarcastic tweets, randomly sampled from the previously annotated corpus. To enhance data quality and align it with the IronyTR dataset, each sample underwent a secondary manual annotation, focusing on selecting more definitive examples of sarcasm and non-sarcasm. Each tweet, both sarcastic and non-sarcastic, is accompanied by a contextual tweet.

TSC_v4 (Turkish Sarcasm Corpus version 4): This version maintains the 1025 identical sarcastic tweets from versions 2 and 3 but limits the non-sarcastic tweets to 500, all manually annotated. Notably, this is the only dataset variant with a greater number of sarcastic than non-sarcastic tweets. Similar to the other versions, each tweet is coupled with a contextual tweet.

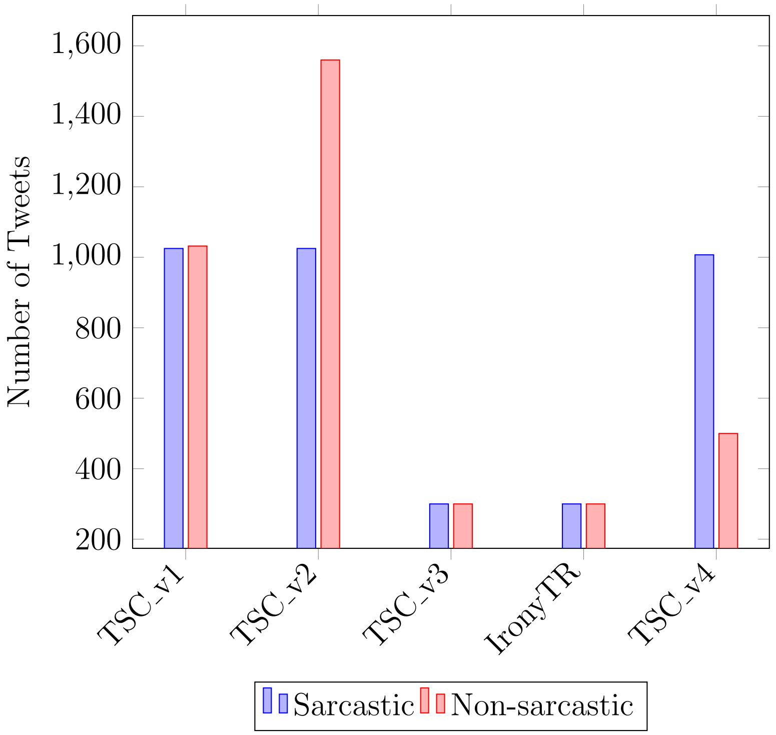

Figure 2: Sarcastic and non-sarcastic tweet distribution in Turkish datasets

IronyTR: An Extended Turkish Social Media Dataset for Irony Detection, initially introduced at GitHub.[5] Comprising 300 sarcastic and 300 non-sarcastic tweets, this dataset is distinct in that it does not include additional contextual text. While maintaining a balanced sarcasm distribution, the content of this dataset has undergone preprocessing, resulting in a more refined and less natural text compared to our tweet samples.

Figure 2 presents a comparison of the distribution of sarcastic and non-sarcastic tweets across five different datasets. In both TSC_v1 and TSC_v2, the number of non-sarcastic tweets exceeds that of sarcastic ones. While TSC_v1 has a more balanced distribution, TSC_v2 has 535 more non-sarcastic tweets than sarcastic ones. However, For both IronyTR and TSC_v3, the chart shows that these datasets have an equal number of sarcastic and non-sarcastic tweets, with each category containing 300 tweets. This indicates a balanced distribution of tweet types in these two datasets. TSC_v4 exhibits a different pattern, with a higher count of sarcastic tweets

[5] https://github.com/teghub/IronyTR

(1007) compared to non-sarcastic ones (500). This visual representation helps in understanding the imbalance or balance in the representation of sarcasm within these datasets. The inclusion of varied distributions of sarcastic and non-sarcastic samples across these datasets serves to evaluate the impact of both balanced and unbalanced datasets on the model's effectiveness.

Using these datasets, we create two experiment settings to measure the impact of conversation context on identifying irony in social media. In the first experiment setting we train our models on four versions of Turkish Sarcasm corpus (TSC) and IronyTR without any additional context inputs. In the second experiment, we train models on each TSC dataset, since IronyTR does not have any additional contextual text, we exclude this dataset in the second experiment. Comparing the different evaluation metric scores, we seek the real impact of context to enhance the performance of our models.

5.2 Experiments without Context

In the initial experimental setup, we consistently trained our BERTurk[6] Model, maintaining identical parameters, across three distinct train-test splits of four datasets. These splits were proportioned as 80-20, 70-30, and 60-40, respectively. Our initial step involved calculating the accuracy score for each version of the dataset split, to identify the most effective split ratio for optimizing model performance.

5.2.1 Accuracy

As seen in Table 6, the TSC_v1 dataset shows a moderate level of accuracy with the highest score at the 80-20 split (72.33) and the lowest at the 70-30 split (69.20). The average accuracy across all splits is 70.77, suggesting that relatively stable performance across split ratios.

There is a slight improvement in the accuracy of TSC_v2 compared to TSC_v1, with the highest score at the 80-20 split (74.02) and the lowest at the 60-40 split (71.37). The average accuracy is 72.44 (+1.6), indicating a bit more consistency in model performance with this dataset version.

TSC_v3 version shows a further increase (+1.6) in accuracy, with the highest at

[6] https://github.com/stefan-it/turkish-bert

Datasets	80-20	70-30	60-40	Average
TSC_v1	72.33	69.20	70.80	70.77
TSC_v2	74.02	71.95	71.37	72.44
TSC_v3	75.66	74.81	72.50	74.32
TSC_v4	77.70	75.42	75.17	76.09
IronyTR	77.33	70.00	77.63	74.98

Table 6: Accuracy scores for models trained on datasets without context inputs

the 80-20 split (75.66) and the lowest at the 60-40 split (72.50). The average accuracy is 74.32, suggesting this version of the dataset might be more suitable for the model (+3.5 from v1 and +1.9 from v2). However, we believe this improvement is due to the incorporation of a manual annotation process for enhancing the annotation quality of the sarcasm component of the dataset, a methodology not employed in the other versions (v1, v2, v4).

TSC_v4 shows the highest accuracy among the TSC versions, it peaks at the 80-20 (77.70) and has its lowest at the 60-40 split (75.17). The average accuracy is 76.09 (+1.8 from v3, +3.7 from v2, and +5.3 from v1), indicating this version of the dataset provides the best performance for the model among the TSC datasets. However, these higher scores are likely influenced by the model's bias towards the majority class, suggesting that metrics such as the F-1 score would offer a more conclusive assessment.

IronyTR, as an external (non-original) dataset in our study, shows high accuracy at both the 80-20 (77.33) and 60-40 (77.63) splits, but as seen in the Figure 3 there is a notable drop at the 70-30 split (70.00). The average accuracy is 74.98 which is high but suggests some variability in performance across different splits. However, considering its more refined version (preprocessed) of isolated text (no real context) compared to other datasets, which provides more straightforward clues of irony than the tweets presented in a more realistic context, it is not surprising that the model trained exclusively on this dataset yields better results.

TSC_v4 and IronyTR datasets demonstrate the highest overall accuracies, especially in the 80-20 split suggesting this train-test split is a relatively better option for such models. However, the variability in the scores, particularly for IronyTR, indicates the model's performance can fluctuate based on the dataset version and the split

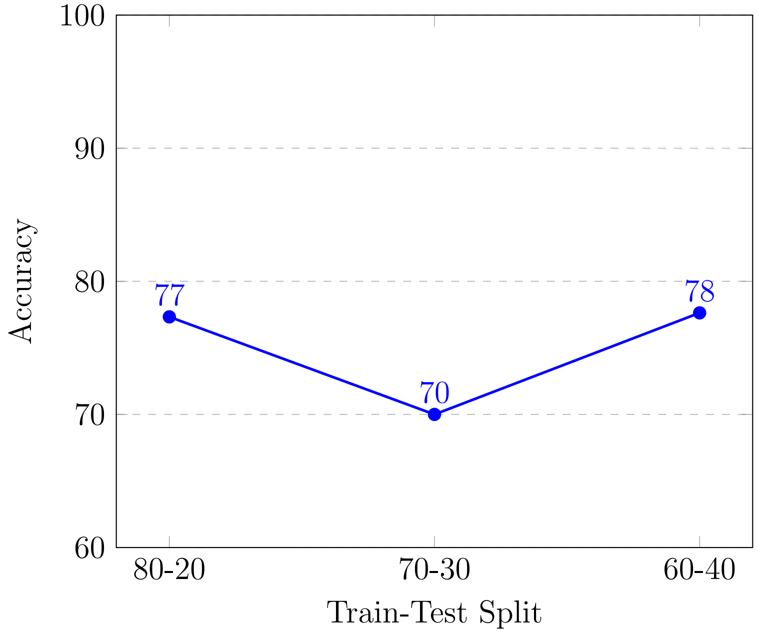

Figure 3: IronyTR performance between different train-test splits

ratio used. The average accuracy scores for each train-test split across all datasets are as follows: approximately 75.41 for the 80-20 split, 72.28 for the 70-30 split, and 73.49 for the 60-40 split. This pattern suggests that the 80-20 split generally provides the best performance, followed by the 60-40 and then the 70-30 splits. Such variability highlights the importance of carefully selecting both the dataset version and the split ratio to achieve optimal performance in the model. Based on these results, we opted to use 80-20 split for training and testing our models, evaluating them using various metrics.

5.2.2 Precision and Recall

Table 7 presents the binary precision and recall scores for various datasets. These metrics are important for understanding how well a model can identify sarcasm (in terms of both relevance and completeness).

With a precision of 0.54 and a recall of 0.56, TSC v1 model is moderately accurate and fairly comprehensive in detecting sarcasm. It correctly labels more than half of the sarcastic tweets, but it is also incorrectly labels a significant of non-sarcastic tweets as sarcastic.

Datasets	**Precision**	**Recall**
TSC_v1	0.54	0.56
TSC_v2	0.37	0.30
TSC_v3	0.45	0.65
TSC_v4	0.71	0.80
IronyTR	0.42	0.35

Table 7: Binary Precision and Recall scores (80-20 Split)

TSC v2 dataset shows a lower performance with precision of 0.37 and recall of 0.30. Despite its 75% accuracy rate, It means model often misclassifies tweets, and it also misses a substantial number of sarcastic tweets. These low scores are likely due to the imbalance in the dataset, resulting in a conservative approach to classifying tweets as sarcastic.

For TSC v3 the precision is 0.45 and recall is higher at 0.65. This suggests that while the model correctly identifies a majority of sarcastic tweets, it also has a relatively high rate of false positive (non-sarcastic tweets classified as sarcastic).

TSC v4 dataset shows the highest performance with a precision of 0.71 and recall of 0.80. This might indicate that the model not only correctly identifies most sarcastic tweets but also maintains a low rate of false positives. Despite having almost twice as many sarcastic tweets as non-sarcastic, the model exhibits strong precision and recall. This indicates a high capability in accurately identifying sarcasm and a lower rate of misclassification, making it robust even in an imbalanced dataset.

IronyTR demonstrates moderate effectiveness in identifying sarcasm, with a precision of 0.42 and recall of 0.35. It somewhat struggles both in terms of correctly identifying sarcastic tweets and in avoiding false positives. Despite being a balanced dataset in terms of the distribution of sarcastic and non-sarcastic tweets (identical numbers with TSC v3) the model shows lower performance in both precision and recall than TSC v3.

From this analysis, we can say that the distribution of sarcastic and non-sarcastic tweets in each dataset significantly impacts the model's performance in terms of precision and recall. A balanced distribution (like in TSC v3 and IronyTR) does not

Datasets	F1-Macro	F1-Binary
TSC_v1	0.51	0.56
TSC_v2	0.56	0.43
TSC_v3	0.52	0.52
TSC_v4	0.54	0.76
IronyTR	0.50	0.44

Table 8: Macro and Binary F1 scores (80-20 Split)

always lead to better performance, as seen in the case of TSC v4, which performs well despite its imbalance. Conversely, TSC v2's struggle is more understandable given its larger number of non-sarcastic tweets.

5.2.3 Evaluation on F1 Scores

In this section, we present the F1 scores for various models, each trained using different versions of datasets with a consistent split of 80% for training and 20% for testing. In the initial experiment, the context was not incorporated as part of the training input.

Table 8 shows the performance of various datasets in a classification task, focusing on F1-macro and F1-binary metrics used to evaluate how well a model performs in classifying sarcasm. The F1-Score, which combines Precision and Recall, is crucial in evaluating the performance of models, particularly in the context of classifying tweets as either sarcastic or not. This metric effectively accounts for both false positives and false negatives. F1-Macro averages the metrics for each label without considering label imbalance. For some datasets like TSC v2 and IronyTR, there is a discrepancy between the F1 Macro and F1 Binary indicating that the model's performance is not consistent across the classes in the dataset.

The F1-Macro score, which considers all classes equally, is sometimes much higher or lower than the F1-Binary score, indicating potential imbalances or biases in the dataset. Although this bias could be observed in TSC v2 due to the unequal distribution of classes (Table 8), the lower binary score in IronyTR, which has a perfectly balanced class distribution (Figure 2), is unexpected. This suggests that the issue may be related to the quality of the samples in these datasets. Notably, this lower binary score issue does not arise in TSC v3, even though its class distribution reflects that of IronyTR.

For the binary task of classifying tweets as sarcastic or not, the F1-binary score is crucial. Higher scores here mean better performance. TSC_v4 is notable with the highest F1-Binary score (0.76), affirming its effectiveness in this specific task. On the other hand, TSC_v2 has the lowest score (0.43) in this category, possibly indicating it is overly cautious in predicting sarcasm, leading to many missed sarcastic tweets.

In this experiment setting in which we do not provide any contextual information to the model, TSC_v2 seems to overlook many sarcastic tweets. In contrast, TSC_v4 shows the most balanced performance across the board, particularly in binary sarcasm classification.

5.2.4 IronyTR vs Turkish Sarcasm Corpus

The creation of Turkish Sarcasm Corpus version 3 was motivated by the need for a resource that could facilitate direct comparisons with external datasets in Turkish Irony Classification tasks. This initiative aligns with our previous discussions about the IronyTR dataset's features. To ensure consistency in terms of the quantity of ironic and non-ironic samples, we randomly selected 300 unique examples from our original collection of 1,025 sarcastic tweets. This collection was initially gathered and annotated through automatic processes. We carefully reviewed each of these 300 sarcastic samples to confirm their effectiveness in representing sarcasm in written text. Additionally, for the non-sarcastic component, we randomly chose 300 samples from our non-sarcastic dataset. These samples were then rigorously evaluated to ensure they did not inadvertently contain any elements of sarcasm.

Unlike the samples in the IronyTR dataset, our samples from both sarcastic and non-sarcastic categories have undergone less preprocessing. We believe that the true effectiveness of a text-based irony classification task is best evaluated by maintaining the text and context as close to real-life scenarios as possible. We demonstrate the differences between our dataset and IronyTR in terms of how the inputs are presented and the extent of preprocessing involved.

Datasets	80-20	70-30	60-40	Average
TSC_v3	75.66	74.81	72.50	74.32
IronyTR	77.33	70.00	77.63	74.98

Table 9: Accuracy scores for models trained on IronyTR and TSC v3

Example (IronyTR):
O kadar hızlı sürdü ki bir an duracağız diye çok korktum!
(He drove so fast that I was so afraid we would stop for a moment!)

Example (TSC v3):
@1parcatuhaftik gece muhafızları galaksimizi kötülüklerden koruyor
(The night guard protects our galaxy from evil)

As seen from examples, IronyTR example is more isolated and straightforward, making it easier to analyze in terms of linguistic structure alone. The TSC v3 example, on the other hand, involves a more complex scenario where understanding the context, the conversation history, and possibly specific cultural or historical references is necessary to grasp the irony. Furthermore, samples from TSC v3 include elements like hashtags and mentions (e.g.@1parcatuhaftik). This complexity makes the TSC v3 example more aligned with real-world text where irony often depends on external contexts and subtleties.

As seen from Table 9, TSC v3 demonstrates consistent performance across various dataset splits (80-20, 70-30, 60-40), with accuracy scores between 72.50 and 75.66 and an average of 74.32, indicating its reliability regardless of dataset division. In contrast, IronyTR shows greater variability in its performance, with accuracy scores ranging from 70.00 to 77.63 and an average of 74.98. Although IronyTR's average accuracy is marginally higher, its fluctuating scores suggest a heightened sensitivity to dataset composition changes.

Datasets	F1-Macro	F1-Binary
TSC_v3	0.52	0.52
IronyTR	0.50	0.44

Table 10: Macro and Binary Scores (80-20 Split)

Train/Test	Precision	Recall	F1-Macro	F1-Weighted	F1-Binary
TSC v3/IronyTR	0.46	0.47	0.44	0.43	0.55
IronyTR/TSC v3	0.64	0.53	0.37	0.33	0.59
TSC v3/TSC v3	0.53	0.53	0.52	0.52	0.52
IronyTR/IronyTR	0.45	0.45	0.45	0.45	0.38

Table 11: Comparison of model performance metrics between TSC v3 and IronyTR datasets

Table 10 presents F1-Macro and F1-Binary scores for two datasets, TSC v3 and IronyTR, each following an 80-20 training-testing split. The scores indicate that the model trained on TSC v3 performs equally well in detecting both classes (sarcastic and non-sarcastic tweets). All variants of the F1-score are also equal to 0.52. An F1-Macro score of 0.52 suggests a balanced performance, as it equally weighs the model's capability to predict both classes. The equal F1-binary score reinforces this balance, indicating good performance in correctly identifying the positive class. On the other hand, for IronyTR, the F1-Macro score of 0.50 shows a balanced performance between classes but slightly lower overall effectiveness compared to TSC v3. The F1-Binary score of 0.44, which is lower than the F1-Macro score, suggests that the model is less effective at correctly identifying the positive class in the IronyTR dataset compared to TSC v3.

As a result, both datasets exhibit relatively balanced performances across classes, as indicated by the F1-Macro scores being around 0.50. TSC v3 shows a slightly better performance in sarcasm detection, with both F1-Macro and F1-Binary scores being higher than those of IronyTR. The differences in performance could be attributed to variations in the dataset characteristics, such as the nature of the sarcasm, and the language used.

Table 11 presents a comparison of various performance metrics, including Precision (macro avg), Recall (macro avg), F1-Macro, F1-Weighted, and F1-Binary, for two test scenarios. The first row shows the performance of TSC v3 model tested on the IronyTR dataset, and the second row details the performance of the IronyTR model on the TSC v3 dataset. This provides insights into the models' effectiveness and generalizability across different datasets. The model appears to perform moderately well on both datasets, with slightly better performance metrics when trained on IronyTR and tested on TSC v3. The scores indicate a reasonable balance between

Datasets	80-20	70-30	60-40	Average
TSC_v1	76.53	75.94	75.17	75.88
TSC_v2	75.24	77.53	75.27	76.01
TSC_v3	85.28	76.48	77.50	79.75
TSC_v4	80.24	79.47	77.11	78.94

Table 12: Accuracy Scores for Model Trained on Datasets with Context Inputs

precision and recall, with some room for improvement in terms of generalizability across different datasets.

Overall, TSC v3 seems more robust, as evidenced by its steady accuracy in different data splits, and marginally outperforms IronyTR in the 80-20 split for sarcasm detection, as shown by its higher F1-binary score. This implies a greater effectiveness in identifying sarcastic utterances. Both models exhibit balanced precision and recall, but their moderate overall scores indicate potential areas for enhancement in sarcasm detection accuracy. TSC v3, with minimal preprocessing, slightly excels in binary classification, possibly due to retaining more contextual information. IronyTR's more extensive preprocessing could be a factor in its slightly inferior performance in binary classification, although it achieves a slightly higher overall accuracy. This suggests that in certain situations, cleaner, more structured data might improve general sarcasm detection but could compromise the detection of contextual subtleties.

5.3 Experiments with Context

In this experiment setup, we trained the BERTurk model the same way on three different parts of four datasets. These parts had different sizes: 80% training and 20% testing, 70% training and 30% testing, and 60% training and 40% testing. The only change from our previous experiment was the addition of context tweets as inputs to our model, along with the target text, which included both sarcastic and non-sarcastic tweets. Our first step was to evaluate the model's accuracy for each dataset split to determine the most effective training-testing ratio for optimizing the model's performance.

5.3.1 Accuracy

According to Table 12, TSC_v1 shows a consistent performance across all splits with a slight drop in the 60-40 split. The highest score is at the 80-20 split (76.53), and

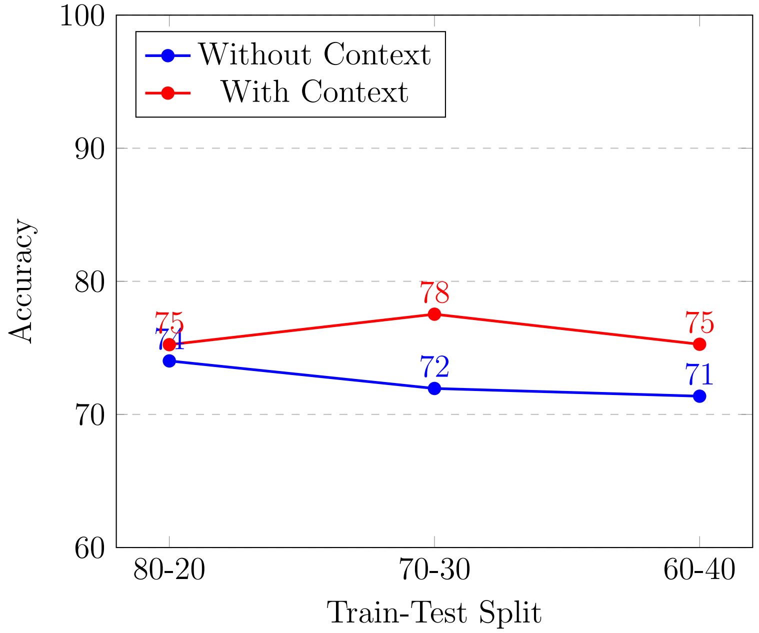

Figure 4: Two TSC v2 models performance between train-test splits

the average accuracy is 75.88. This suggests that the model performs well with this dataset, particularly with more training data.

As shown in Figure 4, TSC_v2 demonstrates a notable increase in accuracy at 70-30 split (77.53), which is the highest across all splits for this dataset. The average accuracy is 76.01, indicating that he model has a fairly consistent performance with TSC_v2, with a peak at a moderate amount of training data.

TSC_v3 dataset shows the highest accuracy score among all datasets in the 80-20 split (85.28). The scores are relatively high in the other splits too, resulting in an average accuracy of 79.75. This high level of performance suggests that the model is particularly well-suited to TSC_v3 with context inputs.

TSC_v4 dataset exhibits strong performance, especially at the 80-20 split (80.24) and 70-30 (79.47). The average accuracy is 78.94, indicating that this version of the dataset also leads to effective model training with context inputs.

The inclusion of context inputs appears to have a positive impact on the model's performance. Particularly, TSC_v3 and TSC_v4 show significantly high accuracies, suggesting context plays a crucial role in understanding and identifying irony in these datasets. The variation in performance across different splits also highlights the importance of selecting an appropriate train-test ratio for optimal model training and evaluation.

5.3.2 Precision and Recall

Table 13 provides binary precision and recall scores for the same datasets as in Table 7, but this time the models have been trained with additional context of preceding tweet. This additional context is expected to influence the model's ability to understand and classify sarcasm more accurately.

Datasets	**Precision**	**Recall**
TSC_v1 (context)	0.53	0.61
TSC_v2 (context)	0.40	0.41
TSC_v3 (context)	0.51	0.53
TSC_v4 (context)	0.68	0.60

Table 13: Binary Precision and Recall scores (80-20 Split)

For TSC v1, the precision is 0.53 and recall is 0.61. Compared to the previous scenario without context, the recall has improved slightly, suggesting that with context the same model is now better at identifying sarcastic tweets.

By including the context TSC v2 dataset, both precision and recall have seen a minor improvement. However, this suggest the context has provided some benefit, but the model still struggles with this dataset, possibly due to its imbalance in sarcastic and non-sarcastic tweets.

While precision for TSC v3 dataset has slightly improved with added context, the recall score has decreased from 0.65 (without context) to 0.53 (with context) indicating that the model, when given additional context, is now missing more of the sarcastic tweets it previously was able to detect.

Interestingly, while the precision for TSC v4 remains high, the recall decreased compared to the scenario without context. This could suggest that the additional context might be causing the model to become more conservative in labeling tweets as sarcastic, potentially due to the nature of the context or its interaction with target tweets.

The decline in performance scores for the model may be attributed to the complexity or irrelevance of additional context, potentially causing the model to miss more sarcastic tweets. If the context is noisy, irrelevant, or overly complex, it could overwhelm or confuse the model. Additionally, if the context doesn't align well with the specific features of sarcastic tweets in certain datasets, like TSC v3, it might not effectively detect sarcasm.

Consequently, while context is often expected to enhance sarcasm detection, it does not always improve model performance uniformly. The impact of context inclusion varies across different datasets. In some instances (like versions 1 and 2), it slightly improves recall and precision, whereas in others (like versions 3 and 4), it may lead to a decline in one of the metrics. Such variations could be due to how well the context integrates with the target tweets in each dataset and the unique characteristics of each dataset.

5.3.3 Evaluation on F1 Scores

The analysis of the F1 scores for the models trained on different versions of the Twitter Sarcasm Corpus (TSC). Each model was trained using distinct versions of datasets, maintaining a consistent division of 80% for training and 20% for testing. In the second experimental setup, we included context as an element of the training input, alongside the target tweets, to determine if they were sarcastic.

The evaluation metrics used include F1-Macro and F1-Binary scores. Each of these metrics provides a different perspective on the performance of the models, and their interpretation is pivotal for understanding the efficacy of sarcasm detection in social media text.

Datasets	**F1-Macro**	**F1-Binary**
TSC_v1 (context)	0.48	0.56
TSC_v2 (context)	0.52	0.41
TSC_v3 (context)	0.55	0.51
TSC_v4 (context)	0.46	0.63

Table 14: Macro and Binary F1 scores (80-20 Split)

As seen in Table 14 the model trained on the TSC v1 dataset exhibits moderate performance with F1-Macro score of 0.48 indicating moderate performance across the classes. A score under 0.5 suggests some imbalance in the model's ability to detect sarcasm equally across different classes. The F1 scores, ranging from 0.48 (Macro) to 0.56 (Binary), further affirm this observation, indicating a moderate level of effectiveness in sarcasm detection.

The model trained on the TSC v2 dataset incorporating target and context tweets demonstrates an improved performance with an F1-Macro of 0.52 indicating a more balanced performance across different classes compared to compared to TSC v1. This enhancement suggests a better alignment in identifying sarcastic versus non-sarcastic tweets. However, the variability in F1 scores, with 0.41 (Binary) and 0.52 (Macro), points to inconsistencies in the model's performance across different classes which may result from the imbalanced distribution of classes within this dataset.

The model trained on TSC v3 dataset shows the highest F1-Macro scores among the datasets at 0.55. This version seems to manage the nuances of sarcasm detection more effectively, reflecting a more reliable and consistent performance in sarcasm detection. The F1-Binary scores, consistently hovering around 0.51 showing an improvement over TSC v2, reflect a well-balanced model with a robust ability to differentiate between sarcastic and non-sarcastic content.

The BERT model trained on the TSC v4 dataset shows moderate performance in identifying sarcasm. It demonstrates a reasonable ability to detect sarcasm (F1-Binary), but its overall balance between F1-Binary and F1-Macro is not very high. Similarly to TSC v1 a drop in the F1-Macro scores can be observed indicating poorer. The highest F1-Binary score of 0.63, indicates a strong performance in detecting the positive class. This version is the best at identifying sarcasm but might be at the cost of misclassifying the negative class more often as suggested by its lower F1-macro score.

In essence, a higher F1-Binary score in some versions (such as TSC v4) does not necessarily correlate with a better overall balance in class performance, as indicated by the F1-Macro score. The discrepancies between the F1-Macro and F1-Binary scores suggest a trade-off between achieving a balanced performance across classes and accurately detecting sarcasm. We hypothesize that these discrepancies between the two F1 scores might also be impacted by imbalanced datasets, for example in

TSC v2 and TSC v4, where significant class imbalances exist (see Figure 2). In such cases, models might develop a bias towards the majority class. Conversely, more balanced datasets, like TSC v1 and TSC v3, are less likely to introduce class bias into the models, leading to a more accurate understanding of the model's ability to distinguish between sarcastic and non-sarcastic content. These results highlight the challenges of sarcasm detection in social media, where context plays a crucial role. However, its incorporation into the model must be carefully managed to ensure both balanced and accurate performance.

5.4 The Effect of Context

In this section, we analyze the outcomes of our experiments to understand how effective our models are at detecting sarcasm in social media. Our primary aim was to evaluate the influence of conversational context in this task. As depicted in Figure 5, there is a marked improvement in the accuracy of our transformer model when trained on datasets with and without contextual tweets. The enhancement in detecting sarcastic utterances in the test data is evident for each version of our TSC datasets.

Utilizing the TSC v1 dataset as training material without contextual information, our model attained an accuracy of 0.71 in recognizing sarcastic utterances in the test data. However, when the same training process was enriched with contextual tweets, the model's accuracy rose to 0.76, a gain of 0.05. This suggests that the inclusion of contextual information significantly aids the model's comprehension of sarcasm.

When we trained the model using the TSC v2 dataset without contextual tweets, it achieved an accuracy of 0.72. With the addition of context, the accuracy improved to 0.76, an increase of 0.04, further indicating the value of contextual data.

The TSC v3 dataset, as previously mentioned, appears to be the most effective in enhancing our model's performance. This dataset also led to the most significant improvement when contextual information was provided. The initial accuracy score was 0.74, which increased to 0.80 with the addition of context, marking the highest improvement of 0.06 observed in our study.

While the TSC v4 dataset yielded better results than those trained on TSC v1 and TSC v2, it showed the least improvement between the versions with and without context. The model achieved an accuracy of 0.74 on this training data, which

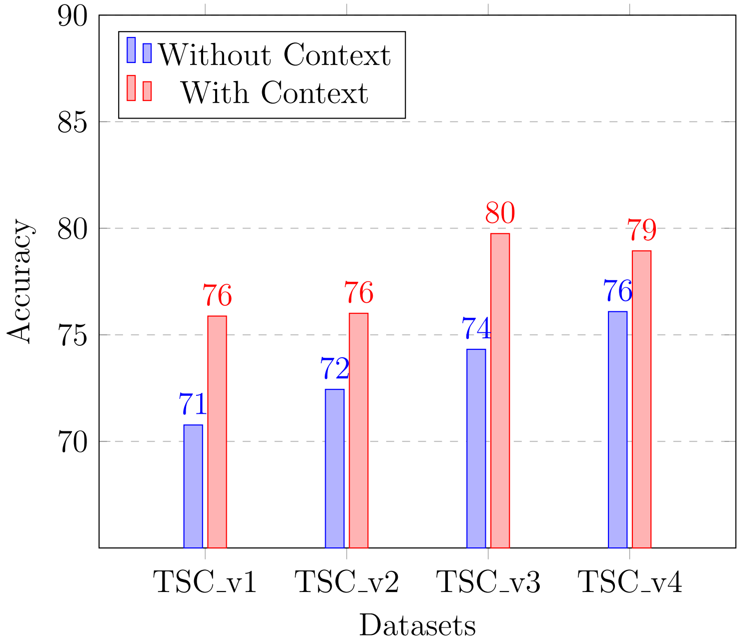

Figure 5: Performance comparison of two models across different training datasets

increased to 0.79 with the inclusion of contextual information.

The accuracy metric can sometimes be misleading in datasets with uneven class distributions, where one class significantly outnumbers another. For example, in a dataset predominantly consisting of sarcastic instances, a model might achieve high accuracy by effectively identifying this majority class. However, this does not necessarily imply improved proficiency in recognizing the minority class. This discrepancy is evident when comparing datasets with varying class balances. For instance, TSC v4 dataset, has a notably unbalanced distribution of sarcastic to non-sarcastic samples. This imbalance is mirrored in its relatively high accuracy scores, ranging from 0.76 to 0.79. In contrast, more balanced datasets like TSC v1, with accuracy scores between 0.71 and 0.76, demonstrate a different scenario. When models are assessed on TSC v4, they often yield lower precision, recall, and F1-macro scores, as indicated in Table 15, highlighting the impact of class distribution on various performance metrics.

The discrepancy between accuracy and F1 scores may also stem from the quality of context tweets in the dataset. Contextual information is vital for discerning the intent behind a tweet, particularly for sarcasm detection. However, these context

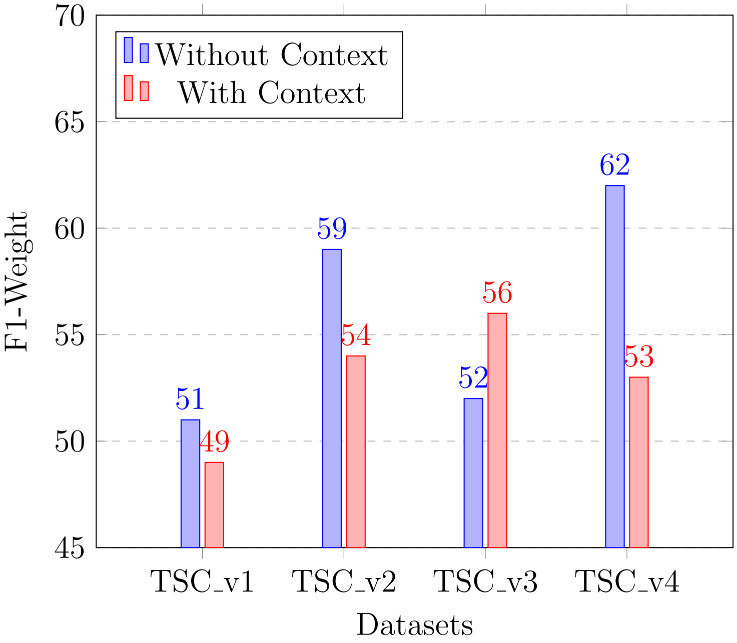

Figure 6: Comparison of weighted F1-scores of two models on each training dataset

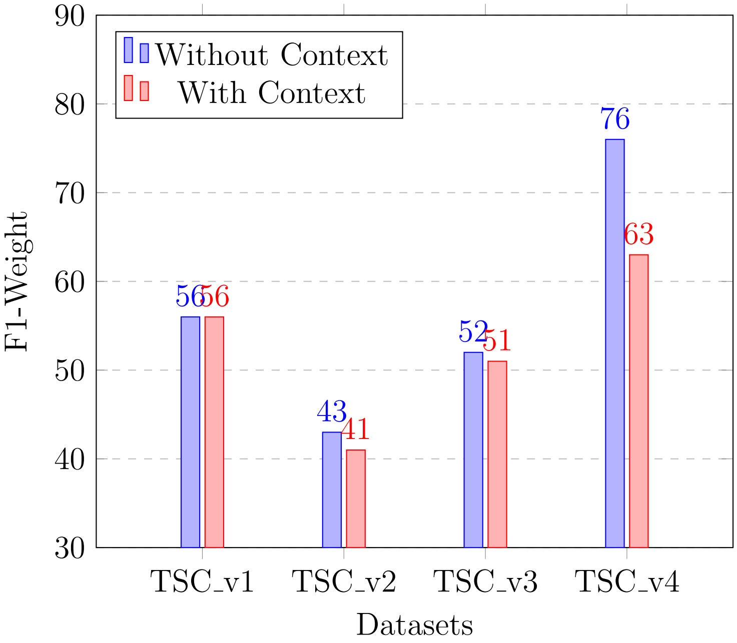

Figure 7: Comparison of binary F1-Scores of two models on each training dataset

Datasets	**Precision**	**Recall**	**F1-Macro**	**F1-Weighted**	**F1-binary**
TSC_v1 (context)	0.48	0.49	0.48	0.49	0.56
TSC_v2 (context)	0.52	0.52	0.52	0.54	0.41
TSC_v3 (context)	0.56	0.56	0.55	0.56	0.51
TSC_v4 (context)	0.47	0.47	0.46	0.53	0.63

Table 15: Detailed classification report scores of models with context (80-20 Split)

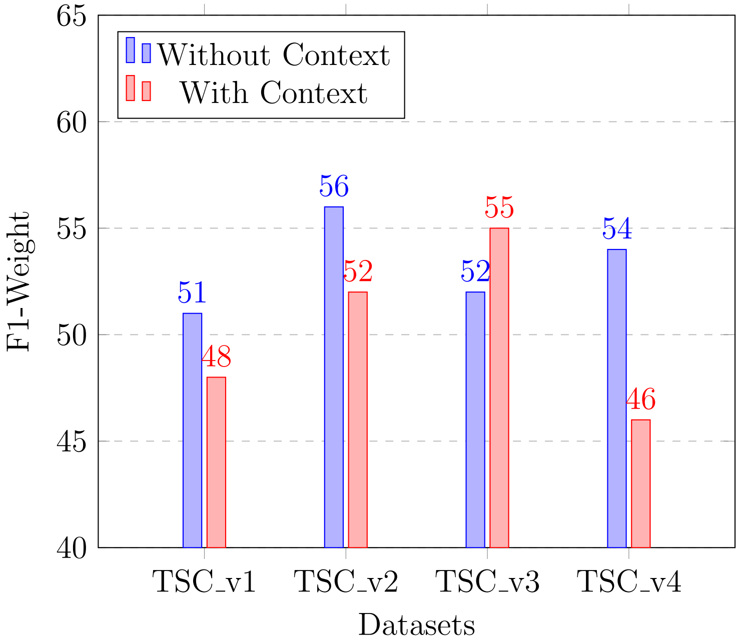

Figure 8: Comparison of macro F1-Scores of two models on each training dataset

tweets, automatically collected via the Twitter API, lack additional annotations to assess whether they add clarity or instead introduce noise and confusion. The relevance and clarity of the context are crucial; ambiguous or weakly indicative contexts may not only fail to aid the model but could also add misleading information. This aspect of dataset quality is a crucial area for further research. Future studies should focus on carefully examining and improving these context tweets before they are used in models, thereby enhancing the effectiveness of sarcasm detection. This approach could explain why models might perform well in certain scenarios, reflected in higher accuracy, yet show inconsistent performance across the board, as indicated by varying F1 scores

Datasets	Accuracy	Precision	Recall	F1-Macro	F1-Weighted	F1-binary
ESC_v1	0.79	0.51	0.51	0.51	0.51	0.48
ESC_v2	0.75	0.52	0.52	0.52	0.55	0.40
ESC_v3	0.85	0.51	0.51	0.51	0.51	0.50
ESC_v4	0.80	0.55	0.55	0.55	0.60	0.70

Table 16: Classification Report Scores without context (English Dataset)

For datasets labeled as having unique tweets[7] (IronyTR, TSC v3, TSC v4) there is no repetition of tweet and context combinations. This could lead to a more diverse range of examples for the model to learn from, potentially impacting the model's ability to generalize. In contrast, non-unique datasets (TSC v1, TSC v2) might contain repetitions in tweet and context combinations, which could influence the model's learning in different ways, possibly leading it to overfit specific patterns that are not representative of the overall task.

The size and composition of the datasets can greatly influence model performance. Larger datasets typically provide more information for the model to learn from, which can lead to better generalization. However, the quality of the data is also crucial. The variation in dataset sizes (e.g., TSC v2 has a larger number of non-sarcastic tweets compared to sarcastic ones) might contribute to different learning dynamics for the model.

Sarcasm detection is inherently challenging due to its reliance on contextual cues and often subtle indicators of tone or intent. This complexity means that even small changes in dataset composition or the inclusion of context can have significant impacts on performance metrics.

5.5 Experiments on English Dataset

In this section, we discuss the results of applying BERT model to the English dataset. This English dataset was gathered and annotated by using the method described in Shmueli et al. (2020). To maintain consistency, both the Turkish and English datasets feature an equal distribution of sarcastic and non-sarcastic samples. For instance,

[7] Unique Tweets indicates that every entry in both the 'text' and 'context tweets' columns is distinct. This means no two rows in a dataset will have the same combination of tweet text and contextual tweets.

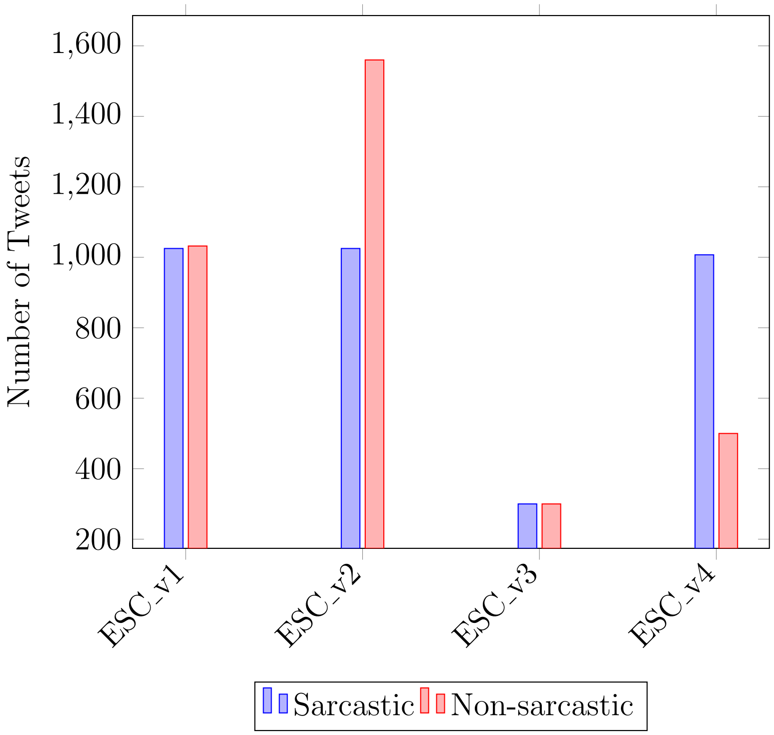

Figure 9: Sarcastic and non-sarcastic tweet distribution in English datasets

the first version of the Turkish Sarcasm Corpus (TSC v1) contains 1032 non-sarcastic and 1025 sarcastic samples, and this distribution is reflected in the English datasets (see Figure 2 and Figure 9 for comparison). The primary objective of this analysis is to assess the impact of conversational context on the task's performance. Similar to the Turkish dataset, the accuracy of the English models also improved when training included additional contextual information from the previous tweet, as illustrated in Figure 10.

When the English Sarcasm Corpus version 1 (ESC v1) was used to train our model without incorporating contextual data, it achieved 79% accuracy rate in identifying sarcastic expressions in the test set, ad detailed in Table 16. However, integrating contextual tweets into the training process improved the model's accuracy to 81% (Table 17), an increase of 0.02. This improvement indicates that including context enhances the model's ability to discern sarcasm in English datasets as well.

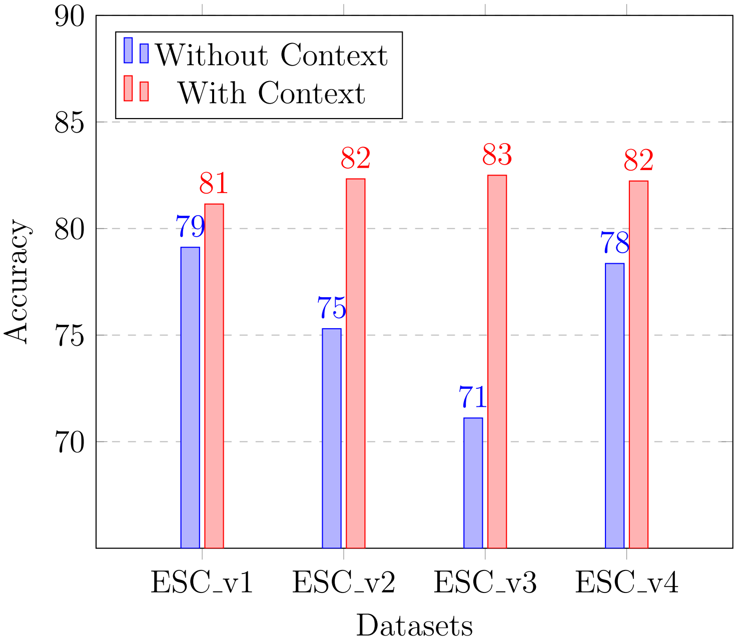

Figure 10: Comparison of English training dataset accuracy scores between two models

Datasets	Accuracy	Precision	Recall	F1-Macro	F1-Weighted	F1-binary
ESC_v1 +context	0.81	0.51	0.51	0.51	0.51	0.52
ESC_v2 +context	0.82	0.52	0.52	0.52	0.53	0.46
ESC_v3 +context	0.83	0.50	0.50	0.50	0.51	0.56
ESC_v4 +context	0.82	0.47	0.47	0.47	0.51	0.60

Table 17: Classification Report Scores with context (English Dataset)

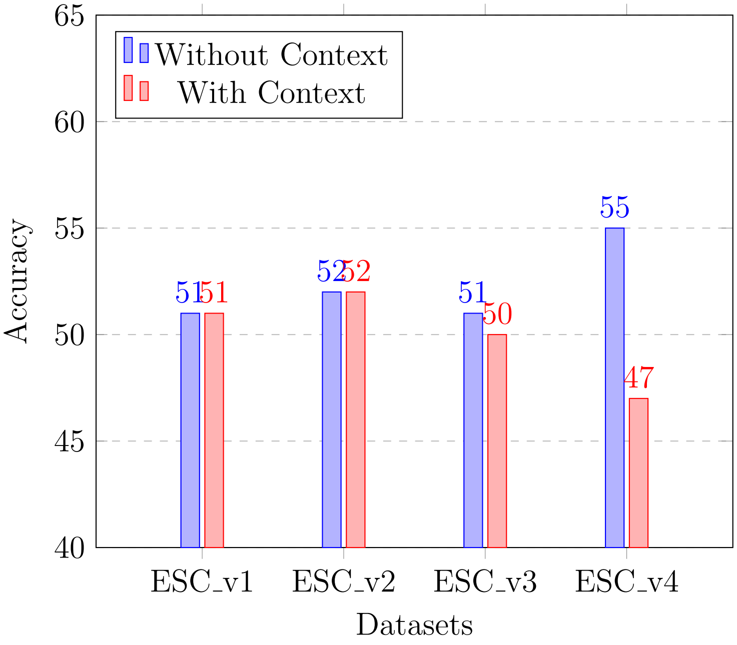

Figure 11: Comparison of macro F1 scores of two models on each training dataset (English)

When we trained the model using the ESC v2 dataset without contextual tweets, it achieved an accuracy of 0.75. With the addition of context, the accuracy improved to 0.82, an increase of 0.07, further indicating the value of contextual data.

Like its Turkish counterpart, the English Sarcasm Corpus version 3 (ESC v3), with its balanced distribution of 300 sarcastic and 300 non-sarcastic samples, significantly enhanced our model's accuracy. The inclusion of contextual information in this dataset led to a remarkable improvement, with accuracy jumping by 0.12. Initially, the model scored 0.71 in accuracy without context, the lowest across the datasets. Adding context boosted this score to 0.83, representing the most significant advancement in our research.

ESC v4 dataset showed one of the two least improvements between the versions with and without context. The model achieved an accuracy of 0.78 on this training data, which increased to 0.82 (+0.04) with the inclusion of contextual information.

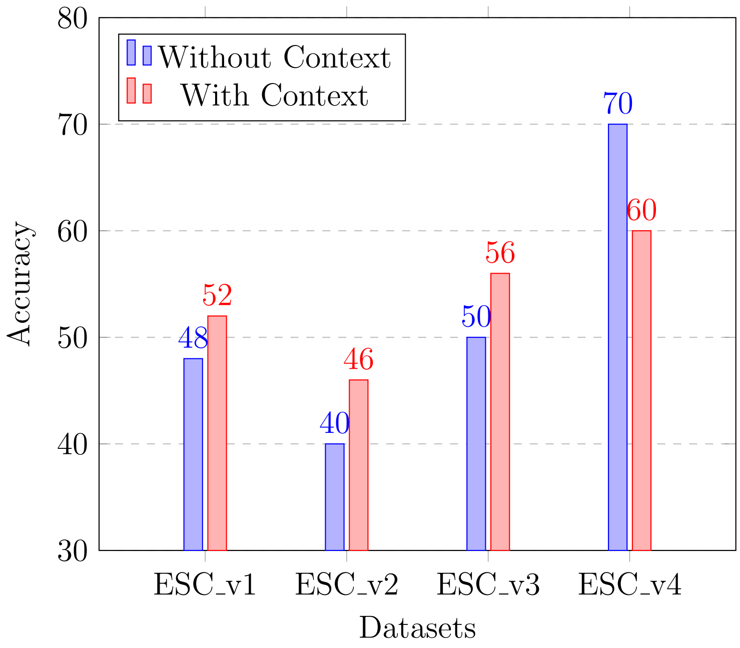

Figure 12: Comparison of binary F1 scores of two models on each training dataset (English)

As we mentioned earlier, the accuracy scores for evaluating model performance can be deceptive, especially considering the size of the data and uneven class distributions in datasets. To obtain a more comprehensive assessment, it is beneficial to employ additional metrics like F1-Macro, and F1-Binary, among others.

In Figure 11, we can see that the F1-Macro scores for each model are not consistently going up. Contrary to expectations, incorporating context into training does not significantly enhance the F1-Macro scores. For instance, while the ESC v3 model shows a notable 12% improvement in accuracy scores when trained with context as opposed to without, this trend is not reflected in its F1-Macro scores. Specifically, the model scores 0.51 in non-contextual training and slightly decreases to 0.50 when context is added, indicating a marginal decline of 0.01. This pattern of decline is also observable in models trained on other versions.

When assessing a transformer model on binary classification, its performance is measured by F1-binary scores. Figure 12 highlights the impact of including context in training datasets on the binary performance of each model. Unlike the F1-Macro

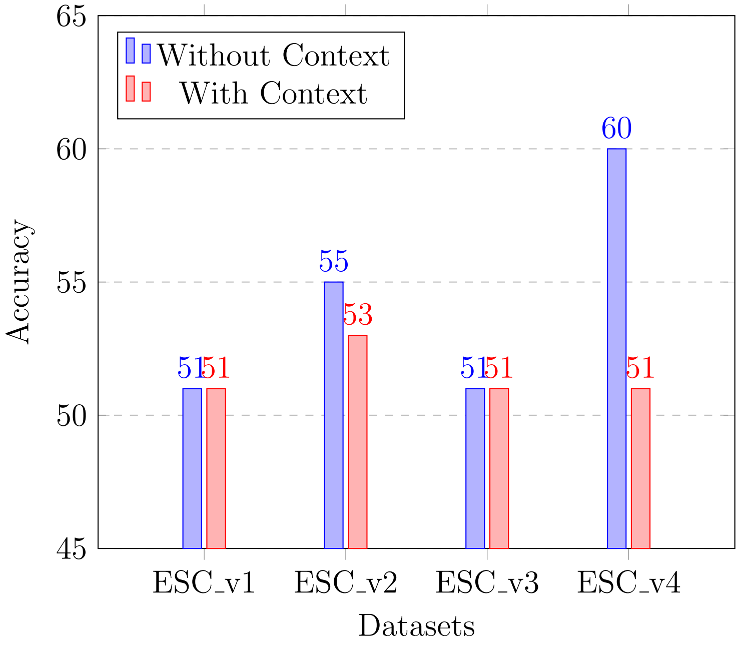

Figure 13: Comparison of weighted F1 scores of two models on each training dataset (English)

Model ID	**Batch Size**	**Learning Rate**	**Dataset**	**Precision**	**Recall**	**Accuracy**	**F1-macro**
xlm-roberta-base	32	2	SPIRS	0.792	0.787	0.787	0.786
xlm-roberta-base	32	5	SPIRS	0.784	0.784	0.784	0.784
xlm-roberta-base	16	2	SPIRS	0.788	0.784	0.784	0.783
xlm-roberta-large	32	2	SPIRS	0.796	0.786	0.786	0.783
bert-base-multilingual-cased	16	2	SPIRS	0.783	0.782	0.782	0.781

Table 18: Detailed performance comparison of the first five NLP Models

scores, the first three English dataset versions show that including the context tweet as input enhances binary scores. However, a deviation occurs with a model trained on the ESC v4 dataset (has one of the unbalanced distributions within our dataset versions), where there is a decline in performance from 0.70 to 0.60, a drop of 0.10. This contrast becomes more intriguing when considering our Turkish sarcasm corpus, as detailed in Figure 8. It shows no improvement in binary scores, suggesting potential differences in dataset quality and representation of sarcastic and non-sarcastic expressions in a language. This implies that the Turkish sarcasm corpus (TSC) may need further refinement. However, further research is required to delve into the reasons behind these variations and their implications.

In this study, we expanded our experiments to include the complete SPIRS dataset, as initially compiled using Reactive Supervision Shmueli et al. (2020). This dataset comprises 30,000 tweets, evenly divided between sarcastic and non-sarcastic examples, each paired with relevant contextual tweets. We tested 25 distinct model configurations, varying elements such as the model identifier, batch size, and learning rate. Each configuration was evaluated using metrics similar to those discussed in previous sections. These comprehensive tests were designed to assess the performance of various BERT models on a dataset larger than those used in our prior experiments. The most successful configuration was found to be the XLM-RoBERTa-base model, with a batch size of 32 and a learning rate of 2e-5. This model achieved an F1-macro score of 0.786, surpassing the scores attained with smaller training datasets (see Table 18). While the size of the training data is a critical factor in model performance, our findings also highlight the importance of fine-tuning other aspects, such as hyperparameters, to optimize model effectiveness.

6 Conclusion & Future Work

In conclusion, this study presents a significant stride in understanding the role of context in detecting sarcasm and irony on social media platforms, particularly Twitter. Utilizing the Turkish Sarcasm Corpus (TSC), generated through the innovative Reactive Supervision technique, we have explored the effects of contextual information on the performance of pre-trained transformer models, fine-tuned for binary classification task. Our findings indicate that incorporating additional conversational context as input markedly enhances the accuracy of these models in identifying ironic utterances within tweets.

Our initial experimental setup, which involved training models without contextual tweet inputs, provided a baseline against which we measured the impact of added conversational context. The most notable improvement was observed in the TSC_v3 model, which demonstrated higher accuracy when extra context was included. However, it is worth noting that other key metrics like F1-macro, F1-weighted, and F1-binary did not exhibit a consistent pattern of enhancement. This inconsistency suggests the need for further investigation into the variables influencing these results.

Looking ahead, we aim to refine the Reactive Supervision methodology. Initially, this approach leveraged the full capabilities of the Twitter API, which became unavailable following its commercialization in 2023. Enhancing our scripts to more effectively extract ironical instances from the previously collected Turkish Twitter data will likely improve both the quality and quantity of our datasets. Given the current limitations in dataset size, adopting a 10-fold cross validation method might prove more efficacious for model training and testing than our current approach. Such a strategy would allow for further model tuning and training on a more representative dataset, potentially augmenting the sarcasm detection capabilities of our models.

Our experiments utilizing a full English dataset comprising 15,000 sarcastic and 15,000 non-sarcastic samples have demonstrated superior performance in metrics such as F1 scores, precision, and recall compared to the four versions of our proposed smaller versions of the dataset. This outcome underscores the significance of dataset size in enhancing a transformer model's capability to accurately identify sarcasm automatically. Future research may explore methods of data augmentation for the Turkish Sarcasm Corpus as well. By expanding the number of samples in both sarcastic and non-sarcastic categories to approximate the volume found in the complete SPIRS English dataset, we anticipate that models trained on this augmented Turkish dataset could achieve comparable F1 scores to those trained on the full English dataset.

In reviewing the reliability of our Turkish Sarcasm Corpus data, particularly the inter-annotator agreement scores obtained during the manual annotation process for the non-sarcastic component of our datasets, we acknowledge the need for further improvement. Currently, our annotation process involves three annotators. In future work, to enhance the reliability and robustness of our dataset, we propose expanding this annotation team to include additional annotators. Our objective is to achieve a Fleiss' kappa score of at least 0.80, which, according to Viera et al. (2005), falls into the category of "almost perfect agreement." This expansion in the annotation process is expected to significantly elevate the quality and consistency of the data within the Turkish Sarcasm Corpus, thereby providing a more solid foundation for future sarcasm detection research.

As we consider future directions for our research, a noteworthy area of exploration involves a deeper engagement with more conversational tweets. Our data collection approach successfully identified tweets engaged in a dialogue, such as oblivious tweets that fail to recognize sarcasm in a preceding tweet. In our study, these tweets were not incorporated into the model training. Therefore, examining the impact of such rich contextual information on the efficacy of sarcasm detection presents an opportunity for further research.

Additionally, the incorporation of user modeling techniques might offer a promising avenue for enhancement. By analyzing specific characteristics of users known for their use of sarcasm, as identified in our dataset, we may achieve a more profound comprehension of the contextual nuances and the inherent markers of sarcasm in their expressions. This approach could significantly refine our model's ability to discern sarcasm with greater precision.

In terms of model selection, our current research utilized the BERT model trained on Turkish data. Moving forward, it would be intriguing to ascertain the outcomes of employing a multilingual BERT model, particularly one trained on an English dataset, to identify sarcasm in Turkish tweets. This approach could shed light on the cross-linguistic applicability and versatility of the model, offering valuable insights into its performance across different linguistic contexts.

This research contributes a methodological framework that not only advances our current understanding of sarcasm detection in social media but also sets the stage for more nuanced and effective studies in the future.

7 Acknowledgements

I would like to express my sincere appreciation to my supervisor, Dr. Cağrı Çöltekin, for his guidance, insight, support, encouragement, and positive attitude throughout this study. I am grateful for the opportunity to work with him.

I would like to extend my sincere gratitude to my second supervisor, Professor Detmar Meurers, for his invaluable contributions to my thesis. His efforts in facilitating interactions with knowledgeable colleagues have significantly enhanced my understanding.

I extend our heartfelt gratitude to Boaz Shmueli for his invaluable contribution. His generosity in promptly providing us with the complete SPIRS English dataset has significantly enriched our research. We deeply appreciate his readiness to assist when we reached out for support.

I also extend my deep gratitude to Fırat Çekinel for his invaluable support in providing the results for the models trained SPIRS English dataset. Without his assistance, running these models would have been impossible due to our limited computational resources.

I am profoundly thankful to my family for their endless patience and encouragement. Their support has been invaluable.

Lastly, I would like to thank all my friends for their help, support, and significant contributions during the data annotation process.

8 Appendix

8.1 Appendix A: More details on data collection process

In our data collection process, we have set up a coding environment with three key components, each designed to handle different aspects of our methodology. This comprehensive setup ensures efficient management and analysis of the provided Twitter data.

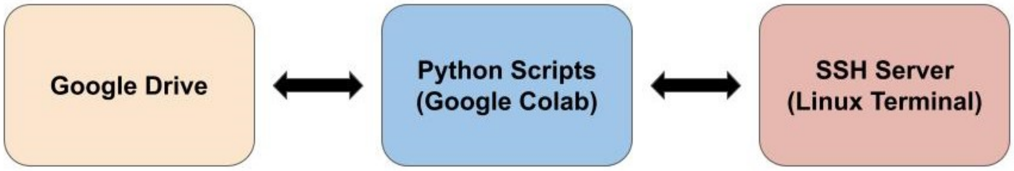

Figure 14: The structure of coding environment for data collection

To begin with, we utilize Google Drive as a storage solution for our smaller files in the data collection process. These files may need processing or may have already undergone preprocessing using Python scripts. Google Drive provides a convenient and accessible platform for storing and organizing these smaller datasets, enabling integration with our data analysis and manipulation workflows.

By leveraging Python 3.10 on Google Colab, we take advantage of the platform's development environment, which offers smooth integration capabilities with Google Drive Service. This decision gives us a versatile and efficient way to manage our data processing tasks while making use of the collaborative features within Google's ecosystem.

To retrieve potential cue tweets containing the query phrase "ironi yap" which is equivalent to the original query "being sarcastic" presented in the paper Shmueli et al. (2020), we utilized the following code snippet in the linux terminal of our server:

```
zcat big_twitter_file.gz | grep -i "Ironi yap" | gzip -c > cue_tweets.gz
```

This command utilizes the "grep" tool, which is an utility for searching patterns within text files. By executing this command in the terminal of our server, we effectively search through the large Twitter dataset for tweets containing a specified query phrase. The results are then saved into the "cue_tweets.gz" file for each month processed for further analysis and processing.

After executing the code snippet to retrieve potential cue tweets containing the query phrase "ironi yap" from the large Twitter dataset, we proceed to upload the resulting file, which typically ranges from 500 KB to 10 MB in size, to our personal Google Drive environment. This step is crucial as it facilitates further processing using our Python scripts on Google Colab, which are linked to our Google Drive storage.

By uploading the file to Google Drive, we effectively integrate it into our data processing pipeline, marking the initial phase of the adopted version of the original 4-step pipeline we discussed earlier. This integration streamlines the flow of data and enables us to leverage the powerful computational capabilities of Google Colab for subsequent analysis and processing tasks.

However, this transition also presents challenges, notably the unavailability of the Twitter API. The absence of access to the Twitter API poses limitations on our ability to freely collect cue tweets by conducting query searches through the API, as we had initially intended. As discussed previously, the changes in Twitter's monetization strategy have restricted our access to the API, necessitating alternative approaches for data collection and analysis.

By compiling a list of tweet IDs from the cue tweet dataset, we can subsequently search for tweets with these IDs within the large Twitter files located on the remote SSH server. This process enables us to identify and gather the corresponding oblivious tweets for further analysis and examination. To accomplish this task, we employ a similar command as used in the initial phase for cue tweets:

```
zcat twitter_file.gz | grep -F -f oblivious_tweet_IDS.txt
| gzip -c oblivious_tweets.gz
```

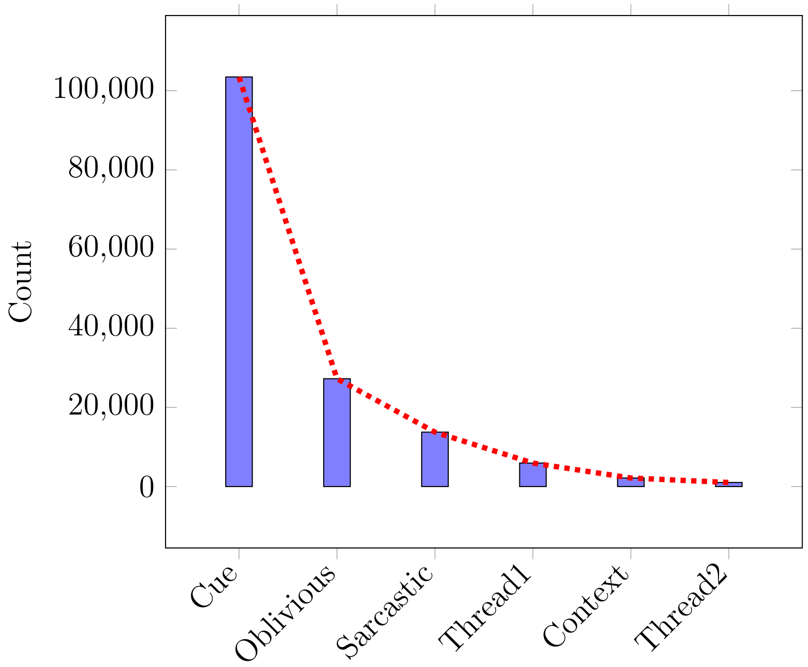

Figure 15: Bar graph showing the counts of different tweet types; Illustrates the data loss between each step in data collection pipeline

Similar procedures are implemented until we reach the tweet that precedes our target text. By the completion of this process, we have successfully gathered 1025 Twitter conversation threads comprising of a cue, an oblivious response, a sarcastic remark (target tweet), and a preceding tweet (context tweet).

8.2 Appendix B: Other Figure and Tables

In this section, we include additional figures that were not incorporated into the main body of our thesis. These supplementary visuals aim to enhance understanding and offer alternate perspectives on the findings from our research.

Figure 15 displays a bar graph depicting the frequency of various tweet types gathered during our data collection phase. This illustration enhances the understanding of the data loss detailed in Section 4.3. 'Thread1' categorizes conversational threads that comprise cue tweets and those with oblivious and sarcastic tweets. 'Context' refers to tweets that appeared prior to the sarcastic tweet. 'Thread2' encompasses the count of conversation threads where all tweet types, ranging from cue tweets to

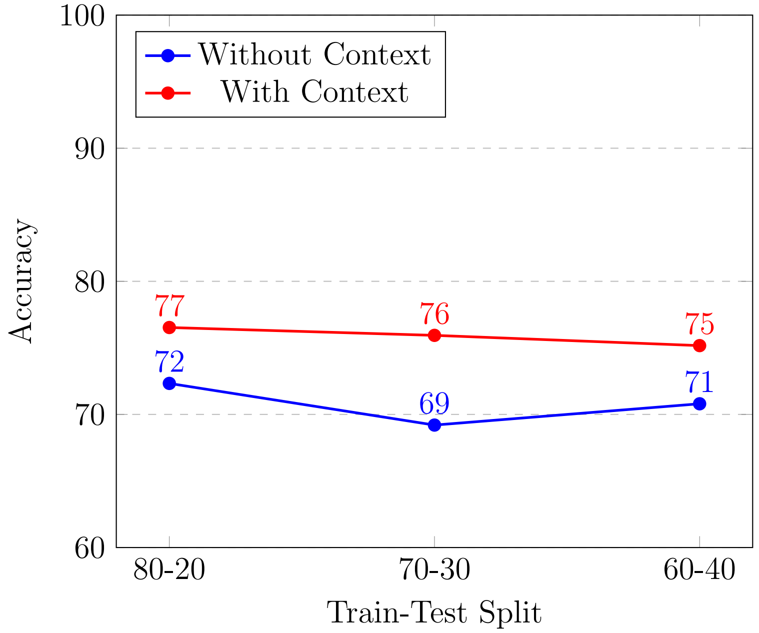

Figure 16: TSC_v1 performance between train-test splits

the aforementioned ’context’ tweets, are included.

Figure 16 displays the accuracy scores for various training and testing split ratios using our Turkish Sarcasm Corpus version 1 data. As detailed in our primary analysis, a split of 80-20 appears to be the most effective for this dataset. This line graph is provided as an alternative representation to the bar graph previously shown.

Similarly to Figure 16, Figure 17 showcases a line graph, offering an alternative to the bar graph format previously used to represent data scores from the third version of the Turkish Sarcasm Corpus.

Figures 18, 19, 20, and 21 offer alternative visualizations, highlighting the variance in model scores trained on data with and without context. These line graphs illustrate the trend of accuracy improvement in the models, as detailed in Section 5. However, they also indicate the lack of consistent performance improvements in the F1-scores across the four different dataset versions.

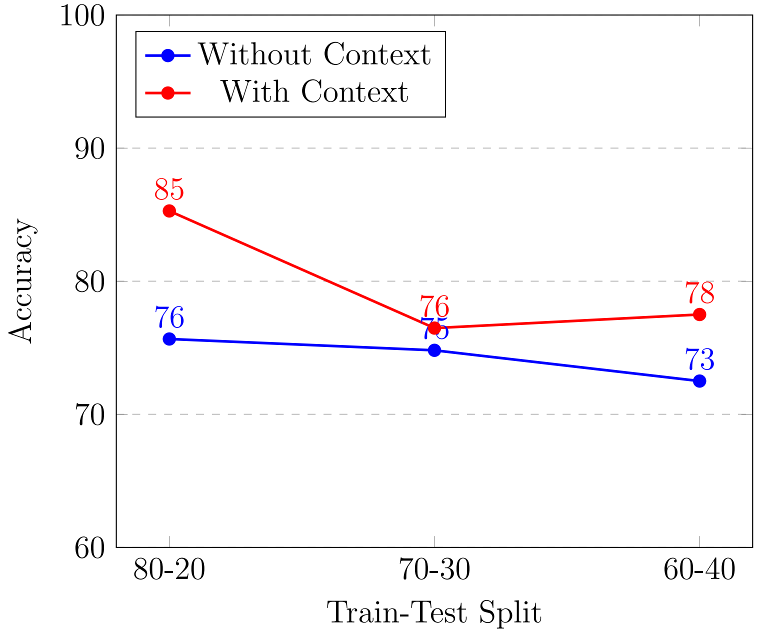

Figure 17: TSC_v3 performance between train-test splits

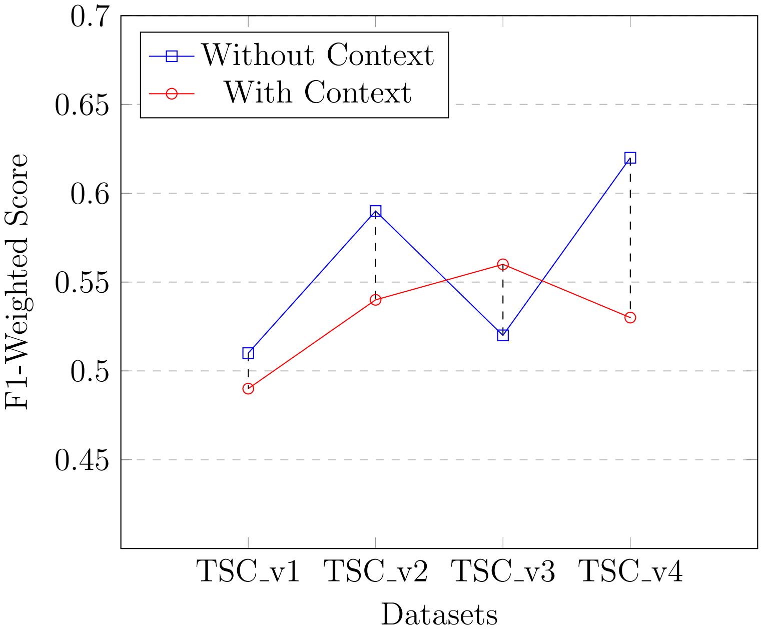

Figure 18: Comparison of weighted F1 scores of models on each training dataset

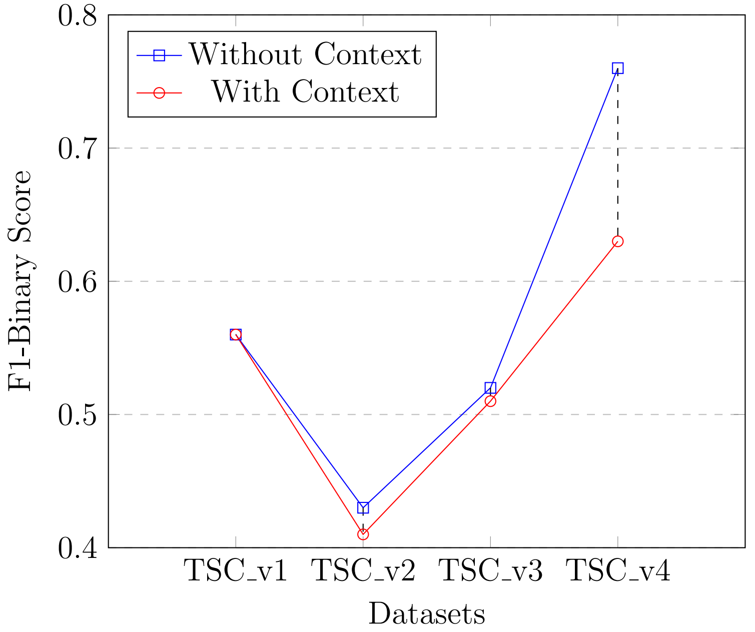

Figure 19: Comparison of binary F1 scores of models on each training dataset

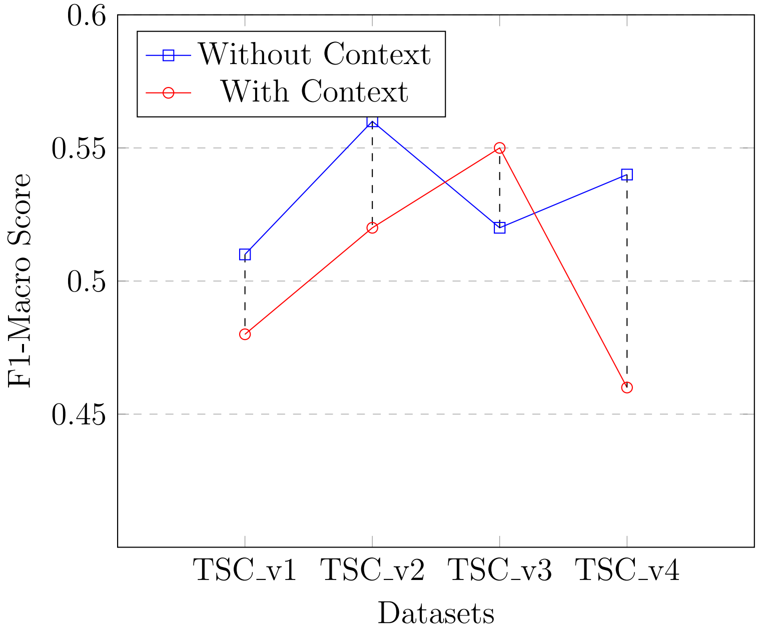

Figure 20: Comparison of macro F1 scores on each version of training datasets

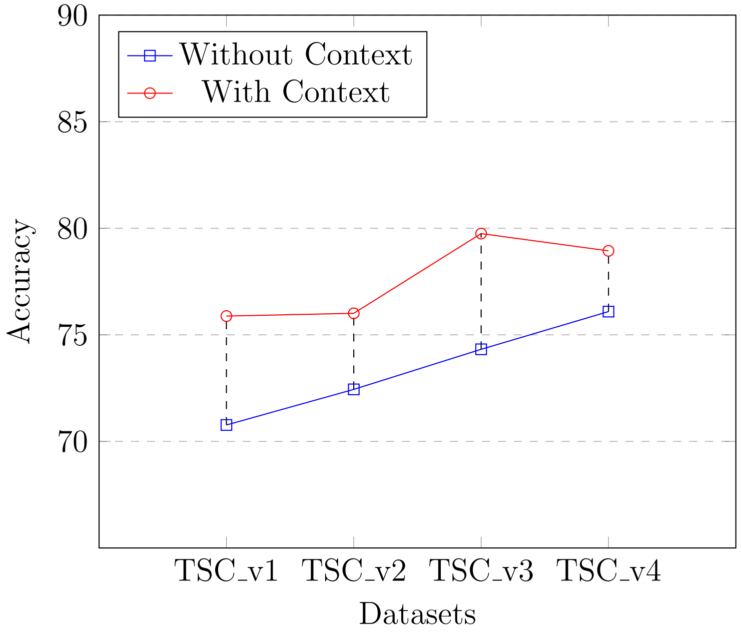

Figure 21: Comparison of Accuracy Scores of two models on each version of training datasets

8.3 Appendix C: Full Classification Report Tables

This section supplements the primary analysis presented in Section 5 by offering a more detailed evaluation. It includes extensive data and tables that cover metrics such as Macro Precision, Macro Recall, and F1-Weighted scores. An analysis example for Table 19 is included to clarify the significance of these metrics in relation to our classification task. We present Table 21 without additional interpretation, allowing future readers of the thesis to draw their own conclusions based on the analysis provided here.

Table 19 shows the performance of various datasets in a classification task, focusing on different metrics (Precision and Recall are macro average) used to evaluate how well a model performs in classifying sarcasm.

Precision measures the proportion of correctly identified sarcastic tweets out of all tweets labeled as sarcastic. High precision means fewer non-sarcastic tweets are mistakenly tagged as sarcastic. TSC_v2 leads in precision (0.57), indicating it is particularly adept at correctly pinpointing sarcastic tweets and avoiding misclassi-

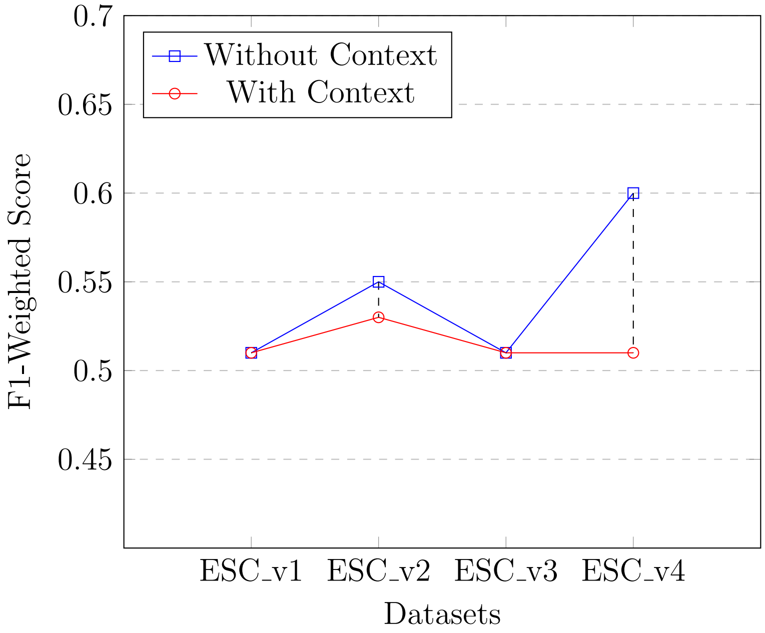

Figure 22: Comparison of weighted F1 scores of models on each English training Dataset (different version)

Datasets	**Precision**	**Recall**	**F1-Macro**	**F1-Weighted**	**F1-binary**
TSC_v1	0.51	0.51	0.51	0.51	0.56
TSC_v2	0.57	0.56	0.56	0.59	0.43
TSC_v3	0.53	0.53	0.52	0.52	0.52
TSC_v4	0.55	0.54	0.54	0.62	0.76
IronyTR	0.50	0.50	0.50	0.50	0.44

Table 19: Detailed Classification Report Scores for datasets (80-20 Split)

fication. However, models like IronyTR and TSC_v3, despite their overall higher accuracy, lag slightly behind in precision with scores of 0.50 and 0.53, respectively.

Recall is the percentage of actual sarcastic tweets correctly identified by the model. In this case, recall scores are fairly consistent across all models, suggesting each is similarly effective at recognizing most sarcastic tweets.

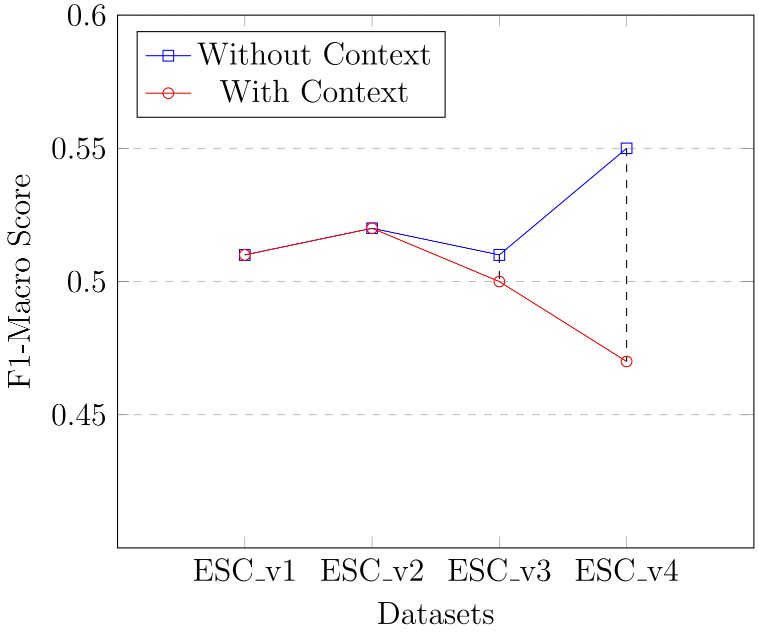

Figure 23: Comparison of macro F1 scores on each version of English training datasets (different version)

The F1-Score combines Precision and Recall, considering both false positives and negatives. We differentiate between F1-Macro, which averages the metrics for each label without considering label imbalance, and F1-Weighted, which does the same but adjusts for the number of true instances for each label. TSC_v4 excels in F1-Weighted score (0.62), suggesting it is the most balanced model overall, especially when considering class imbalances. TSC_v2, while having higher precision, falls behind in this measure with a score of 0.59.

For the binary task of classifying tweets as sarcastic or not, the F1-Binary score is crucial. Higher scores here mean better performance. TSC v4 is notable again with the highest F1-Binary score (0.76), affirming its effectiveness in this specific task. Surprisingly, TSC v2, despite its precision score, has the lowest score (0.43) in this category, possibly indicating it's overly cautious in predicting sarcasm, leading to many missed sarcastic tweets.

Datasets	Precision	Recall	F1-Macro	F1-Weighted	F1-binary
TSC_v1 (context)	0.48	0.49	0.48	0.49	0.56
TSC_v2 (context)	0.52	0.52	0.52	0.54	0.41
TSC_v3 (context)	0.56	0.56	0.55	0.56	0.51
TSC_v4 (context)	0.47	0.47	0.46	0.53	0.63

Table 20: Detailed classification report scores of models with context (80-20 Split)

Datasets	Precision	Recall	F1-Macro	F1-Weighted	F1-binary
TSC_v3	0.53	0.53	0.52	0.52	0.52
IronyTR	0.50	0.50	0.50	0.50	0.44

Table 21: Detailed classification report scores of TSC v3 and IronyTR datasets (80-20 Split)

Table 20 presents the full version of Table 14 with metric likes precision, recall, and F1-Weighted included. Table 21 presents a more detailed version than what is shown in Table 10. Metrics like Precision, Recall, and F1-Weighted is included for further analysis.

References

Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. 2016. Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452.

Nastaran Babanejad, Heidar Davoudi, Aijun An, and Manos Papagelis. 2020. Affective and contextual embedding for sarcasm detection. In *Proceedings of the 28th international conference on computational linguistics*, pages 225–243.

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

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2014. Italian irony detection in twitter: a first approach. *Italian irony detection in Twitter: a first approach*, pages 28–32.

Alexandru-Costin Băroiu and Ștefan Trăușan-Matu. 2022. Automatic sarcasm detection: Systematic literature review. *Information*, 13(8):399.

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

Mondher Bouazizi and Tomoaki Ohtsuki. 2015. Sarcasm detection in twitter:" all your products are incredibly amazing!!!"-are they really? In *2015 IEEE global communications conference (GLOBECOM)*, pages 1–6. IEEE.

Elisabeth Camp. 2012. Sarcasm, pretense, and the semantics/pragmatics distinction. *Noûs*, 46(4):587–634.

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

Rebecca Clift. 1999. Irony in conversation. *Language in society*, 28(4):523–553.

Herbert L Colston. 2000. On necessary conditions for verbal irony comprehension. *Pragmatics & Cognition*, 8(2):277–324.

Anne Cutler. 1974. On saying what you mean without meaning what you say.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010a. Enhanced sentiment learning using twitter hashtags and smileys. In *Coling 2010: Posters*, pages 241–249.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

S Dixon. 2022. Twitter: number of worldwide users 2019-2024. *Statista. URL: https://www. statista. com/statistics/303681/twitter-users-worldwide [accessed 2023-03-17]*.

Yu Du, Tong Li, Muhammad Salman Pathan, Hailay Kidu Teklehaimanot, and Zhen Yang. 2022. An effective sarcasm detection approach based on sentimental context and individual expression habits. *Cognitive Computation*, 14(1):78–90.

Jodi Eisterhold, Salvatore Attardo, and Diana Boxer. 2006. Reactions to irony in discourse: evidence for the least disruption principle. *Journal of Pragmatics*, 38(8):1239–1256.

Elisabetta Fersini, Federico Alberto Pozzi, and Enza Messina. 2015. Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers. In *2015 IEEE international conference on data science and advanced analytics (DSAA)*, pages 1–8. IEEE.

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

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. 2015. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 470–478.

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

Aniruddha Ghosh and Tony Veale. 2017. 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.

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

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

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

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

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Umang Gupta, Ankush Chatterjee, Radhakrishnan Srikanth, and Puneet Agrawal. 2017. A sentiment-and-semantics-based approach for emotion detection in textual conversations. *arXiv preprint arXiv:1707.06996*.

John Haiman. 1998. *Talk is cheap: Sarcasm, alienation, and the evolution of language*. Oxford University Press.

Yanfen Hao and Tony Veale. 2010. An ironic fist in a velvet glove: Creative misrepresentation in the construction of ironic similes. *Minds and Machines*, 20:635–650.

Stacey L Ivanko and Penny M Pexman. 2003. Context incongruity and irony processing. *Discourse processes*, 35(3):241–279.

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

Shengyi Jiang, Chuwei Chen, Nankai Lin, Zhuolin Chen, and Jinyi Chen. 2021. Irony detection in the portuguese language using bert. *Proceedings http://ceur-ws. org ISSN*, 1613:0073.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5):1–22.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark Carman. 2016a. Harnessing sequence labeling for sarcasm detection in dialogue from tv series 'friends'. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 146–155.

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

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

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

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

Jennifer Ling and Roman Klinger. 2016. An empirical, quantitative analysis of the differences between sarcasm and irony. In *The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29–June 2, 2016, Revised Selected Papers 13*, pages 203–216. Springer.

Peng Liu, Wei Chen, Gaoyan Ou, Tengjiao Wang, Dongqing Yang, and Kai Lei. 2014. Sarcasm detection in social media based on imbalanced classification. In *Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings 15*, pages 459–471. Springer.

S Lukin and M Walker. 2013. Really. In *Well. Apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. Language and Social Media Workshop, NAACL*, page 30.

Edwin Lunando and Ayu Purwarianti. 2013. Indonesian social media sentiment analysis with sarcasm detection. In *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 195–198. IEEE.

Aaron Maladry, Els Lefever, Cynthia Van Hee, and Veronique Hoste. 2022. https://doi.org/10.18653/v1/2022.wassa-1.16 Irony detection for Dutch: a venture into the implicit. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 172–181, Dublin, Ireland. Association for Computational Linguistics.

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

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Abdullah Y Muaad, Hanumanthappa Jayappa Davanagere, JV Bibal Benifa, Amerah Alabrah, Mufeed Ahmed Naji Saif, D Pushpa, Mugahed A Al-Antari, and Taha M Alfakih. 2022. Artificial intelligence-based approach for misogyny and sarcasm detection from arabic texts. *Computational Intelligence and Neuroscience*, 2022.

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

Silviu Vlad Oprea and Walid Magdy. 2020. The effect of sociocultural variables on sarcasm communication online. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–22.

Asli Umay Ozturk, Yesim Cemek, and Pinar Karagoz. 2021. Ironytr: Irony detection in turkish informal texts. *International Journal of Intelligent Information Technologies (IJIIT)*, 17(4):1–18.

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

Yang Qiao, Liqiang Jing, Xuemeng Song, Xiaolin Chen, Lei Zhu, and Liqiang Nie. 2023. Mutual-enhanced incongruity learning network for multi-modal sarcasm detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9507–9515.

Rachel Rakov and Andrew Rosenberg. 2013. " sure, i did the right thing": a system for sarcasm detection in speech. In *Interspeech*, pages 842–846.

Md Saifullah Razali, Alfian Abdul Halin, Lei Ye, Shyamala Doraisamy, and Noris Mohd Norowi. 2021. Sarcasm detection using deep learning with contextual features. *IEEE Access*, 9:68609–68618.

Lu Ren, Bo Xu, Hongfei Lin, Xikai Liu, and Liang Yang. 2020. Sarcasm detection with sentiment semantics enhanced multi-level memory network. *Neurocomputing*, 401:320–326.

Antonio Reyes and Paolo Rosso. 2012. Making objective decisions from subjective data: Detecting irony in customer reviews. *Decision support systems*, 53(4):754–760.

Antonio Reyes and Paolo Rosso. 2014. On the difficulty of automatically detecting irony: beyond a simple case of negation. *Knowledge and Information Systems*, 40:595–614.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47:239–268.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. 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.

Stefan Schweter. 2020. https://doi.org/10.5281/zenodo.3770924 Berturk - bert models for turkish.

Simona G Shamay-Tsoory, Rachel Tomer, and Judith Aharon-Peretz. 2005. The neuroanatomical basis of understanding sarcasm and its relationship to social cognition. *Neuropsychology*, 19(3):288.

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

Stephen Skalicky and Scott Crossley. 2018. Linguistic features of sarcasm and metaphor production quality. In *Proceedings of the Workshop on Figurative Language Processing*, pages 7–16.

Dan Sperber and Deirdre Wilson. 1981. Irony and the use-mention distinction. *Philosophy*, 3:143–184.

Hande Taşlıoğlu. 2014. Irony detection on turkish microblog texts. Master's thesis, Middle East Technical University.

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

Cynthia Van Hee. 2017. *Can machines sense irony?: exploring automatic irony detection on social media*. Ph.D. thesis, Ghent University.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018a. Exploring the fine-grained analysis and automatic detection of irony on twitter. *Language Resources and Evaluation*, 52(3):707–731.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018b. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 39–50.

Anthony J Viera, Joanne M Garrett, et al. 2005. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363.

Byron C Wallace, Eugene Charniak, et al. 2015. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1035–1044.

Byron C Wallace, Laura Kertz, Eugene Charniak, et al. 2014. 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.

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