# *An Investigation of Deep Learning Approaches in Natural Language Processing (NLP) for Sarcasm Detection in Web-Based Text: Toward the Development of a Real-Time Browser Extension*

*By*
*Evelina Mariova Ivanova*
*Student ID: 2677336*


*Supervisor: Dr Phillip Smith*

*A thesis submitted to the University of Birmingham*

*For the degree of MSc in Data Science*

*School of Computer Science*

*University of Birmingham, Birmingham, UK*

*September 2025*

# *Table of Contents*

# *Honour Code*

*"I certify that this project is my own work. I consulted ChatGPT-4o for guidance on Unity concepts, debugging suggestions, initial project architecture, and minor editorial edits for consistency across scripts and methods. ChatGPT-4o generated images related to the interface of the extension and some of the tables. I independently designed, implemented, and verified all code and design decisions, and personally reviewed and selected all sources and references. The overall report structure and initial draft are my own; ChatGPT-5 was used solely for final edits to improve clarity and consistency. All AI-assisted outputs were critically reviewed and adapted by me to accurately reflect my own work."*

# *Acknowledgements*

*My sincere thanks to my supervisor, Dr Phillip Smith, for his support, patience, and invaluable guidance throughout this project, and to my second supervisor, Dr Rickson C. Mesquita, for his constructive feedback and encouragement.*

*I am also grateful to the staff and lecturers at the University of Birmingham, whose dedication, expertise, and commitment to teaching have made my studies both stimulating and rewarding, and have greatly supported the development of this dissertation.*

*Finally, my deepest thanks go to my family and frinds. Without your constant encouragement and unwavering support, I would not have been able to pursue my studies abroad or make the most of this life-changing opportunity.*

# *Abstract*

*Sarcasm is a challenging aspect of online communication, where meaning diverges from surface form through polarity inversion, pragmatic contrast, or rhetorical cues. Its prevalence complicates sentiment analysis, stance detection, and dialogue systems, making robust detection essential for NLP.*

*This dissertation evaluates whether current models can detect sarcasm reliably across domains and in real time. A balanced dataset combining Reddit, Twitter, and news headlines was used to benchmark classical baselines, deep learning models, and transformer architectures under a unified protocol. Evaluation considered not only accuracy and F1 but also calibration, latency, and model size.*

*Transformers decisively outperformed earlier models: RoBERTa achieved the strongest accuracy, while DistilRoBERTa matched performance with much lower latency and footprint, making it optimal for deployment. Domain analyses confirmed robustness on news and Reddit but highlighted persistent challenges on Twitter. Finally, deployment through the SarcQuest Chrome extension demonstrated both the feasibility and limitations of real-time sarcasm detection in everyday web use.*

# Chapter 1: Introduction

## 1.1 Motivation and Problem Statement

Sarcasm is one of the most pervasive and subtle devices in human communication. Unlike literal language, where meaning typically aligns with lexical form, sarcasm relies on the contrast between the explicit utterance and its intended interpretation. For example, the remark "What a wonderful idea to schedule a meeting at 7 a.m." superficially appears positive, yet conveys frustration or criticism. This divergence between form and intent makes sarcasm particularly challenging for computational models (Camp, 2012).

In digital communication, sarcasm is widespread. Social media platforms such as Reddit and Twitter, as well as online journalism and commentary, contain a high density of ironic, mocking, or satirical content (Khodak et al., 2018; González-Ibáñez et al., 2011; Misra & Arora, 2019). For humans, sarcasm can function as humour, critique, or social bonding, but it also carries the risk of misinterpretation, especially in text-only settings where prosody and body language are absent. For Natural Language Processing (NLP), sarcasm presents a consistent source of error: sentiment analysis systems misclassify sarcastic negativity as positivity, stance detectors confuse irony for support, and dialogue agents risk producing inappropriate responses when sarcasm is misunderstood (Riloff et al., 2013).

Addressing sarcasm is therefore not only a theoretical problem in computational linguistics but also a practical requirement for reliable NLP applications. However, operationalising such systems requires models that are not only accurate but also efficient, calibrated, and robust across heterogeneous text sources.

This dissertation addresses these challenges by developing SarcQuest, a browser extension that highlights sarcastic sentences on live webpages. The work combines multi-domain dataset integration, systematic model benchmarking, and deployment-oriented evaluation to investigate whether modern NLP models can reliably detect sarcasm in real-world conditions.

However, the central objective of this dissertation is not the development of a browser extension per se, but the systematic evaluation of sarcasm detection models. SarcQuest functions as a demonstration platform rather than the research focus: its role is to provide a realistic deployment setting in which insights from model benchmarking can be validated. The scientific contribution therefore lies in analysing accuracy, efficiency, calibration, and robustness across modelling families, with deployment serving as a supporting test of feasibility.

## 1.2 Research Questions

The study is structured around four guiding research questions:

**RQ1 (Modelling):** How do classical baselines, pre-transformer deep learning models, and transformer-based encoders compare for sarcasm detection when evaluated under a consistent experimental framework?

**RQ2 (Domain Robustness):** How well do models generalise across domains such as Reddit, Twitter, and news headlines, and what domain-specific limitations emerge?

**RQ3 (Reliability):** Are model probability estimates well-calibrated, and can calibration methods such as temperature scaling improve their suitability for threshold-based applications?

**RQ4 (Deployment):** Which models achieve the most effective balance between predictive accuracy, efficiency, and size, and can these be operationalised in a real-time browser extension?

## 1.3 Scope of the Study

This dissertation focuses on text-based sarcasm detection. Sarcasm is treated as a binary phenomenon at the sentence or utterance level, with related pragmatic devices such as humour or irony considered only insofar as they overlap with sarcasm in annotated datasets. Multimodal signals such as images, video, or audio are excluded, as the deployment target a browser extension operates primarily on text. Contextual depth is also constrained: although sarcasm often depends on conversational turns or world knowledge, the models here are trained on single-sentence annotations, reflecting the constraints of most available datasets.

## 1.4 Contributions

This dissertation makes four principal contributions to the field of sarcasm detection. First, a large-scale, balanced, and multi-domain corpus of over one million entries is created by integrating publicly available datasets from Reddit, Twitter, and news headlines. Rigorous cleaning and normalisation procedures are applied, including the removal of duplicates, invalid entries, and annotation artefacts, alongside contraction expansion, negation propagation, and punctuation standardisation. This process ensures the quality of the dataset and enables meaningful comparison across domains, while exploratory analysis highlights stylistic and structural differences between platforms.

Second, a unified benchmarking framework is established, within which classical machine learning methods, deep learning architectures, and transformer-based models are implemented and evaluated under consistent experimental conditions. The models range from Logistic Regression baselines to BiLSTM and CNN networks, including BiLSTM with Attention, as well as state-of-the-art transformer models such as BERT, RoBERTa, DistilRoBERTa, and DistilRoBERTa augmented with emotion embeddings. By training and assessing these models under identical protocols, the study ensures fair comparison and provides a clear understanding of the trade-offs between approaches.

Third, the evaluation extends beyond predictive accuracy to include measures of reliability and efficiency, thereby reflecting the practical requirements of real-time deployment. Calibration is assessed using Expected Calibration Error and Brier Score, while inference latency is measured on both CPU and GPU. Model size and memory footprint are also recorded, highlighting the balance between accuracy and efficiency that is necessary for integration into lightweight applications. For further analysis, both domain-specific evaluation and misclassification analysis are conducted, providing insight into how models perform across heterogeneous sources and which linguistic cues most frequently lead to error.

Finally, the dissertation contributes a proof-of-concept operational deployment through the development of *SarcQuest*, a functional Chrome extension. The extension demonstrates the feasibility of sarcasm detection in real-time web browsing by extracting text from webpages, transmitting it to a Hugging Face-hosted inference API, and highlighting sentences predicted as sarcastic. This practical component illustrates both the opportunities and the limitations of deploying sarcasm detection technology in everyday contexts, thereby bridging the gap between theoretical performance and applied usability.

## *1.5 Significance and Potential Impact*

The significance of this work lies in bridging the gap between academic sarcasm detection research and real-world deployment. While prior studies often focus exclusively on predictive accuracy, this dissertation demonstrates that deployment requires equal attention to latency, calibration, and user trust.

The potential impact extends beyond technical advancement:

- **Accessibility:** Supporting neurodivergent readers who struggle with pragmatic inference (e.g., Happé, 1995). Research in developmental psychology has shown that individuals on the autism spectrum, for instance, often experience difficulties in recognising nonliteral intent such as sarcasm or irony, since these rely heavily on shared context and implicit social cues. A tool capable of automatically flagging sarcastic sentences could therefore reduce barriers to online communication, enhancing both comprehension and participation in digital discourse.
- **Language learning:** Assisting English language learners in recognising irony and sarcasm, which are often considered among the most difficult aspects of pragmatic competence (Cheang et al., 2011).
- **Business applications:** Enabling tone-aware monitoring of online discourse, where inappropriate or misunderstood sarcasm can affect customer relationships (Naz et al., 2019).

While these applications are not the primary focus of the dissertation, they illustrate the broader societal and practical value of reliable sarcasm detection systems.

Taken together, these contributions highlight the dual value of this research: advancing the scientific understanding of sarcasm detection across modelling families and domains, while simultaneously demonstrating its practical viability through a real-world deployment scenario.

# Chapter 2: Literature Review

## 2.1 The Nature of Sarcasm

Sarcasm is commonly defined as a rhetorical device in which the literal meaning of an utterance contrasts sharply with its intended meaning, typically to convey ridicule, contempt, or ironic humour (Camp, 2012; Attardo, 2000). While irony and sarcasm are closely related, most linguistic accounts treat sarcasm as a subtype of irony distinguished by its critical or mocking intent (Gibbs, 1986). Irony can include broader phenomena such as understatement or situational incongruity, whereas sarcasm is often more pointed, targeting a specific individual, event, or claim.

For Natural Language Processing (NLP), sarcasm poses significant challenges. Unlike sentiment analysis, where positive or negative polarity tends to be explicit, sarcasm frequently relies on polarity inversion—using positive surface forms to convey negative intent (e.g., "oh great, another meeting at 7 a.m.") (Riloff et al., 2013). Beyond polarity, sarcastic meaning often depends on implicit world knowledge or conversational context: without understanding that a situation is undesirable, models may misinterpret the utterance as genuinely positive (Bamman & Smith, 2015). This pragmatic dependency makes sarcasm particularly difficult for text-only systems that lack access to prosody, facial expression, or shared situational awareness (Kreuz & Roberts, 1995).

Researchers have therefore explored a range of linguistic cues associated with sarcastic expression. Negation is a frequent signal, as sarcastic utterances often flip expectations through constructions such as "not bad at all" (Khodak et al., 2018). Punctuation cues, such as exclamation marks or question marks, quotation marks, and unusual capitalisation, have been shown to correlate with sarcastic intent (González-Ibáñez et al., 2011). Other stylistic markers include positive sentiment words used in incongruous contexts ("yeah right", "awesome job") (Riloff et al., 2013), or overconfident expressions that contrast with reality (Oprea & Magdy, 2020). Although these cues provide useful features, they are often unreliable in isolation and highly domain-dependent.

Early computational approaches to sarcasm detection attempted to operationalise these cues through rule-based or feature-driven systems. For example, Riloff et al. (2013) proposed a pattern-based classifier capturing the contrast between positive sentiment verbs and negative situation descriptors. Similarly, Kreuz and Caucci (2007) examined lexical and pragmatic cues in manually annotated datasets. While these approaches established important baselines, they often suffered from poor generalisation: handcrafted rules could capture sarcasm in narrow contexts but failed when applied to new domains, topics, or linguistic styles (Joshi et al., 2017). This motivated the shift towards data-driven machine learning approaches, beginning with linear classifiers trained on n-gram features and later evolving into deep learning and transformer-based methods.

## 2.2 Datasets in Prior Work

The development of sarcasm detection systems has been closely tied to the availability of annotated datasets. Much of the early work on sarcasm detection relied on single-domain corpora, typically constructed from Reddi or Twitter (González-Ibáñez et al., 2011; Khodak et al., 2018; Misra & Arora, 2019). While these resources have provided valuable benchmarks, their domain-specific nature has also limited generalisability across diverse text types.

### 2.2.1 Reddit

The most widely used Reddit corpus is the Self-Annotated Reddit Corpus (SARC) introduced by Khodak et al. (2018). It contains over one million comments labelled for sarcasm, using the "/s" convention common in online communities. The dataset is valuable because of its scale and coverage across a wide range of topics, from politics to entertainment, and because it preserves conversational context. However, it is also noisy: annotation depends on user adherence to the "/s" marker, which is inconsistently applied, and the informal, discussion-based nature of Reddit makes sarcasm highly context-dependent. Smaller Reddit datasets have also appeared on Kaggle (Danofer, 2017), again leveraging community annotation practices.

### 2.2.2 Twitter

Twitter has been another major source of sarcasm data. González-Ibáñez et al. (2011) produced one of the first corpora, combining n-gram and pragmatic features annotated by crowdworkers. Subsequent datasets used distant supervision via hashtags such as #sarcasm or #not (e.g., Sulis et al., 2016), though these introduce annotation artefacts that models may overfit. More recently, the iSarcasm dataset introduced by Oprea & Magdy (2020) provided high-quality, author-intended sarcasm labels, enabling more robust benchmarking. Twitter datasets capture the brevity and stylistic features of microblogs (emojis, hashtags, creative orthography), but are typically imbalanced, noisy, and highly reliant on conversational or cultural context that is not always accessible in text alone

### 2.2.3 News headlines

A contrasting style of sarcasm appears in the News Headlines Corpus for Sarcasm Detection by Misra and Arora (2019), consisting of 26,709 headlines drawn from The Onion (sarcastic) and HuffPost (non-sarcastic). These headlines are written by professionals in a formal manner, making them short, grammatical, and topical. Sarcasm in this dataset is often achieved through incongruent co-occurring word phrases or a reliance on common knowledge rather than overt markers typically found in social media. This dataset is notably cleaner and less noisy than social media corpora due to its controlled label quality. Its binary source-based labelling method, where all The Onion headlines are considered sarcastic and all HuffPost headlines non-sarcastic, is presented by the authors as a way to obtain very high-quality, controlled labels in a relatively large quantity. However, the formal, editorial style of these headlines differs significantly from conversational or social-media discourse, which limits cross-domain generalisation and suggests the need for techniques like transfer learning for downstream tasks.

### 2.2.4 Limitations of single-domain corpora

While Reddit, Twitter, and news headlines each provide valuable resources, they capture different facets of sarcastic expression: Reddit offers context-rich community discussions, Twitter presents short and informal posts often marked with hashtags or emojis, and headlines exemplify editorial irony. Studies that trained and evaluated models on a single domain have repeatedly shown significant drops in performance when those models are tested on another domain (e.g., Ghosh & Veale, 2016; Bamman & Smith, 2015). This demonstrates that sarcasm is not a uniform phenomenon but one that is highly sensitive to genre and medium. The cues that signal sarcasm vary accordingly—quotation marks and discourse contrast are more prevalent in Reddit threads, hashtags and emojis dominate in Twitter, and topical incongruity characterises news headlines. These differences highlight the challenge of cross-domain generalisation and underscore the need for multi-domain approaches such as the one pursued in this study.

### 2.2.5 Motivation for the present study

These limitations motivate the approach taken here: combining Reddit, Twitter, and news headlines into a single corpus to ensure both scale and stylistic diversity. By merging multiple domains, the dataset used in this dissertation is better aligned with real-world deployment scenarios, such as browser extensions, where text encountered by users is heterogeneous and cannot be assumed to belong to a single source. This design provides a stronger foundation for evaluating model robustness and cross-domain generalisation than single-domain benchmarks alone.

## 2.3 Models for Sarcasm Detection

Research on automatic sarcasm detection has progressed through several methodological stages, beginning with classical machine learning classifiers, moving into pre-transformer deep learning architectures, and culminating in transformer-based language models. Each stage reflects attempts to capture the complex linguistic and pragmatic cues of sarcasm with increasing representational capacity.

### 2.3.1 Classical Machine Learning Approaches

Early computational studies typically represented text with surface-level features such as n-grams, sentiment lexicons, punctuation cues, and pragmatic markers, and then trained supervised classifiers. González-Ibáñez et al. (2011) applied Logistic Regression and SVMs to Twitter data, showing that linear models could capture lexical patterns like “yeah right” or “great job” when combined with sentiment features. Riloff et al. (2013) proposed a pattern-based classifier that explicitly modelled the contrast between positive sentiment words and negative situations, an important linguistic mechanism in sarcasm. Similar approaches integrated handcrafted features such as negation, intensifiers, and exclamation marks to improve robustness (Reyes et al., 2013).

Comprehensive surveys (Joshi et al., 2017) conclude that SVM and Logistic Regression dominated early work, providing competitive baselines that remain widely reported in later studies. These models are computationally efficient and interpretable but struggle with contextual phenomena, since they treat text as a bag of features rather than modelling sequential or discourse-level dependencies.

### 2.3.2 Deep Learning before Transformers

The introduction of deep learning enabled models to move beyond manually engineered features. Convolutional Neural Networks (CNNs) were among the first to be applied, effective at capturing local n-gram patterns associated with sarcasm (e.g., “sure thing”, “best day ever”) (Poria et al., 2016; Amir et al., 2016). Recurrent Neural Networks (RNNs), particularly Bidirectional LSTMs, improved upon this by modelling sequential dependencies and polarity shifts across longer sentences (Ghosh & Veale, 2016). Attention mechanisms were later added to highlight salient tokens, improving interpretability by showing which words drove sarcastic predictions (Tay et al., 2018).

Although these architectures reduced reliance on manual features, performance gains over linear baselines were often modest. Studies found that deep models still overfit domain-specific markers and remained brittle when sarcasm depended on conversational or world knowledge (Joshi et al., 2017). Nonetheless, they established a bridge between feature-driven classifiers and transformer-based architectures.

### 2.3.3 Transformer-Based Models

The advent of large pre-trained transformers marked a decisive shift in sarcasm detection. BERT (Devlin et al., 2019) introduced bidirectional contextual embeddings through self-attention, significantly outperforming

prior models on Twitter and Reddit sarcasm corpora (Savini & Caragea, 2022). RoBERTa (Liu et al., 2019), trained with larger corpora and improved objectives, further advanced performance and has been reported as one of the strongest baselines in recent shared tasks *(Shu, 2024)*.

For real-time or resource-constrained settings, compressed architectures such as DistilBERT and DistilRoBERTa (Sanh et al., 2019) retain most of the accuracy of their larger counterparts while reducing size and latency. This efficiency has made them attractive choices for deployment-oriented studies. Comparative evaluations show that transformers consistently achieve 7–10 point F1 improvements over classical or pre-transformer neural models (Sinha & Choudhary, 2023), confirming their state-of-the-art status.

### *2.3.4 Extensions and Enhancements*

Beyond single encoders, researchers have explored auxiliary signals to capture pragmatic tone. Emotion and affect features have been integrated into sarcasm models, either through handcrafted features (Reyes et al., 2013) or pretrained emotion encoders such as GoEmotions (Demszky et al., 2020). These approaches reflect findings that sarcasm often co-occurs with affective stance shifts. Contextual extensions have also been proposed: Bamman and Smith (2015) demonstrated that incorporating author history and conversational context improved sarcasm detection on Twitter, while Wallace et al. (2014) highlighted the role of discourse in Reddit. More recently, multimodal sarcasm detection (Cai et al., 2019) has combined text with images (e.g., memes), though this remains outside the scope of text-only studies.

Sarcasm detection has evolved from linear models to transformers, which now provide state-of-the-art cross-domain performance, while advances in emotion modelling, context, and calibration address remaining challenges with implicit sarcasm.

## *2.4 Deployment and Practical Considerations*

While research on sarcasm detection has advanced steadily, comparatively little work has focused on the practical requirements for deploying models in real-world applications. Most prior studies evaluate performance only in terms of accuracy or F1-score, yet deployment in interactive environments—such as browser extensions or conversational agents—requires additional considerations relating to efficiency, calibration, and domain robustness.

### *2.4.1 Latency and efficiency*

User-facing systems demand near-instant responses. A browser extension, for example, must process sentences within a fraction of a second to avoid disrupting the reading experience. Classical models such as Logistic Regression or SVM are trivially fast (microseconds per prediction), but their accuracy is limited (Joshi et al., 2017). By contrast, transformer-based models achieve state-of-the-art performance but incur higher computational cost. Distilled architectures such as DistilBERT and DistilRoBERTa reduce inference time and memory usage by approximately 40–60% while retaining most of the accuracy of their teacher models (Sanh et al., 2019). Such trade-offs between predictive quality and responsiveness are critical in deployment-oriented research, where even minor latency can affect user acceptance.

### *2.4.2 Model size and resource constraints.*

Deployment contexts often impose strict limits on memory and storage. Full-size models like BERT-base can exceed several hundred megabytes, which is impractical for lightweight environments such as web extensions or mobile apps. DistilRoBERTa, in contrast, offers a reduced footprint (~250MB) while maintaining robust performance (Sanh et al., 2019). These efficiency gains have made compressed

transformer models attractive candidates for operational sarcasm detection pipelines (Savini & Caragea, 2022).

### 2.4.2 Domain robustness

Real-world environments expose models to heterogeneous data, yet most systems are trained and evaluated on a single domain. Prior work shows that models trained on Twitter often degrade on Reddit or news, and vice versa (Bamman & Smith, 2015; Ghosh & Veale, 2016). For deployment, this cross-domain fragility limits generalisability: a browser extension cannot assume the user only visits news sites or social media. Addressing domain variation—whether through multi-domain training, domain adaptation, or dynamic thresholding—is therefore essential for operational success.

Sarcasm detection research has begun addressing efficiency, calibration, robustness, and privacy, but real-world evaluations remain scarce. This work contributes by benchmarking models across domains and demonstrating feasibility through the deployment of the SarcQuest browser extension.

## 2.5 Summary and Research Gaps

This chapter reviewed prior work on sarcasm detection across four dimensions: the linguistic background of sarcasm, the datasets commonly used, the modelling approaches explored, and the practical issues surrounding deployment.

Research has established that sarcasm is a complex linguistic phenomenon, often realised through polarity inversion, pragmatic contrast, and stylistic markers such as negation, punctuation, or exaggeration. Existing datasets have been primarily single-domain, with large-scale Reddit corpora, hashtag-annotated Twitter collections, and news headlines each capturing different cues. However, models trained on one genre often degrade when applied to another, highlighting domain sensitivity as a critical obstacle.

Methodological approaches have evolved from classical machine learning baselines (e.g., Logistic Regression, SVMs), through pre-transformer neural networks such as CNNs and BiLSTMs, to transformer-based architectures like BERT and RoBERTa. Empirical studies consistently show that transformers achieve state-of-the-art performance, though often at the cost of efficiency and calibration. Deployment considerations remain underexplored, with few studies systematically addressing inference latency, model size, or probability reliability despite their importance for real-time, user-facing applications.

Taken together, these findings highlight several gaps in the literature:

1. **Multi-domain integration**: Most prior work evaluates sarcasm detection within a single dataset or domain, limiting cross-domain generalisability.
2. **Comprehensive model comparison**: Few studies compare linear, deep neural, and transformer-based approaches under a unified experimental protocol, making it difficult to assess trade-offs fairly.
3. **Calibration and reliability**: High-performing models often produce overconfident probability estimates, restricting their practical reliability in threshold-based applications (Guo et al., 2017).
4. **Real-time deployment**: Existing studies rarely extend to live environments, meaning latency, model footprint, and user interaction remain open challenges.

This dissertation addresses key gaps by building a multi-domain dataset, evaluating models from linear baselines to transformers on accuracy, calibration, efficiency, and robustness, and deploying sarcasm detection in the SarcQuest browser extension to demonstrate both its potential and limitations in real-world use.

# Chapter 3: Data and Preprocessing

## 3.1 Data Sources (Twitter, Reddit, News Sarcasm Dataset)

This study draws on multiple publicly available sarcasm datasets spanning different platforms and styles of online discourse. The decision to combine heterogeneous sources was motivated by two considerations: (i) to provide sufficient scale for training deep learning models, and (ii) to ensure coverage of diverse domains representative of real-world use cases for a browser extension. The three domains included are Reddit, Twitter, and news headlines.

### 3.1.1 Reddit

The largest component of the corpus was sourced from the Reddit Sarcasm Dataset published on Kaggle (Danofer, 2017) [1]. This dataset contains over 1 million labelled Reddit comments, where sarcasm labels were derived from users explicitly marking their own posts with the "/s" convention (Danofer, 2017). Reddit data is valuable because it reflects naturally occurring, community-generated sarcasm across a wide range of topics, from politics, scientific to popular culture (Khodak, 2017). Posts vary in length and style, often embedding sarcasm within longer discourse, which makes the detection task more challenging for some models than in short, standalone sentences (Ghosh & Veale, 2016).

### 3.1.2 News Headlines

The second dataset was the News Headlines Sarcasm Corpus by Misra and Arora (2019), also hosted on Kaggle. It consists of 26,709 headlines drawn from two sources: The Onion (a satirical news outlet) labelled as sarcastic, and HuffPost headlines labelled as non-sarcastic (Misra & Arora, 2019). This dataset introduces a different form of sarcasm: structured, editorially written, and often reliant on topical irony rather than conversational cues (Misra & Arora, 2019). Its shorter length and journalistic style contrast with Reddit's informal discourse, making it a useful testbed for cross-domain generalisation.

### 3.1.3 Twitter

Finally, two separate datasets were used to capture sarcasm in short-form, social-media contexts. The iSarcasmEval dataset (Farha et al., 2020), released for the SemEval-2020 shared task, provides labelled English tweets annotated for sarcasm and irony. The Sarcasm Tweets Dataset on HuggingFace (Nikesh66, 2021) contains additional labelled tweets, offering a more balanced distribution of sarcastic and non-sarcastic examples. Tweets differ from Reddit and news data in both length and linguistic form (González-Ibánez et al., 2011). They frequently employ emojis, hashtags, or exaggerated punctuation, and often encode sarcasm through concise, informal phrasing (Wahyuni, 2025). While smaller and noisier than other domains, Twitter adds crucial stylistic diversity to the dataset and mirrors the type of short, fast-moving text common in real-world social media platforms (Rajadesingan et al., 2015).

## 3.2 Dataset Statistics and Class Distribution

A clear understanding of the dataset is essential for establishing reliable evaluation baselines and ensuring that the findings are relevant for both academic benchmarking and practical deployment. In this section, descriptive statistics are presented for the combined sarcasm dataset, covering overall size, class balance, and domain composition. These characteristics directly influence how models are trained, evaluated, and ultimately deployed within the browser extension.

### 3.2.1 Overall dataset size

The raw merged dataset comprised 1,047,403 rows, with class proportions that were already highly balanced: 526,678 non-sarcastic examples (50.3%) and 520,725 sarcastic examples (49.7%). Such balance provides a clear advantage for modelling, as it is difficult to achieve in practice; sarcastic instances are considerably less frequent in natural text, where non-sarcastic examples typically dominate (Khodak et al., 2017). By balancing the dataset, I reduced the risk of the models simply defaulting to the majority class. This step makes evaluation metrics such as accuracy and F1-score genuinely reflective of model performance rather than being skewed by class imbalance. Following the cleaning pipeline described in *Section 3.3*, the dataset was reduced to 950,694 rows, representing the removal of approximately 9.2% of the data. Importantly, the class balance was preserved during this process: 467,758 non-sarcastic examples (49.2%) and 482,936 sarcastic examples (50.8%) remained. This outcome indicates that noise reduction was achieved without introducing bias towards one class. For model training and evaluation, this means results can be interpreted with confidence that performance is not being distorted by imbalance.

### 3.2.2 Per-source distribution

Although balanced overall, the dataset is not uniform across domains. It integrates three distinct sources of sarcastic and non-sarcastic text: Reddit, Twitter, and news headlines. These differ not only in size but also in style and structure, which has direct implications for model generalisation.

Reddit constitutes the overwhelming majority of the dataset, with 1,010,826 rows before cleaning and 919,475 after. Its distribution was perfectly balanced before cleaning (50% sarcastic, 50% non-sarcastic) and shifted only slightly towards sarcastic content afterwards (51.1% sarcastic vs. 48.9% non-sarcastic). The dominance of Reddit data ensures that deep learning models have sufficient examples to capture nuanced sarcastic patterns.

News headlines are far smaller, with 26,709 rows before cleaning and 26,597 afterwards, a negligible reduction. The class proportions remained consistent at approximately 44% sarcastic vs. 56% non-sarcastic. As headlines are formal, and often contextually anchored in real-world events, they offer a different stylistic challenge for sarcasm detection models.

Twitter represents the smallest and noisiest source. It contained 9,868 rows before cleaning but was reduced to only 4,622 rows afterwards, meaning that over half of the original tweets were removed due to duplication, extremely short length, or noise. The reduction also intensified the class imbalance: while the original distribution was 36.4% sarcastic vs. 63.6% non-sarcastic, the cleaned set contained only 23.5% sarcastic vs. 76.5% non-sarcastic. This skew increases the difficulty of evaluating sarcasm detection models in this domain, as they are more likely to overpredict the majority (non-sarcastic) class.

## 3.3 Data Cleaning and Normalisation

The initial dataset, drawn from Reddit, Twitter, and news headlines, comprised over one million entries. However, the raw text contained substantial noise, duplication, and inconsistencies that, if left unaddressed, would have undermined both model performance and evaluation reliability. A systematic cleaning and normalisation pipeline was therefore implemented. The goal of this process was to standardise textual inputs across domains, remove artefacts unlikely to contribute meaningful signals, and preserve stylistic markers that are central to sarcasm detection.

Duplicate removal. Repeated entries were prevalent, particularly in Reddit and Twitter where popular sarcastic posts or memes circulated widely. Duplicates pose a risk of data leakage across training and test

sets, artificially inflating performance. Removing them ensured that evaluation more accurately reflected generalisation ability rather than memorisation.

Filtering of low-quality data. Several exclusion criteria were applied:

- Empty or invalid texts. A total of 183 entries with missing values or null strings were discarded.
- Very short texts. Posts shorter than four characters (e.g., “ok”, “no”) were excluded, as manual inspection confirmed they lacked sufficient context for sarcasm and risked introducing label noise.
- Excessively long texts. Entries exceeding 500 characters (<0.03% of the dataset) were removed. These were disproportionately noisy, often consisting of copied discussions or repeated tokens, with sarcastic content limited to a small fragment.
- Symbol-only entries. Examples composed solely of hashtags, punctuation, or symbols were eliminated, as they lacked lexical information necessary for classification.

Text normalisation, to improve consistency across domains, several standardisation steps were undertaken:

- URL and emoji removal. Web links and emojis were stripped. While emojis may occasionally encode sarcasm, their sparse occurrence relative to text-based cues and potential to disrupt tokenisation motivated their exclusion.
- Hashtag cleaning. The hash symbol (#) was stripped from all tokens. Explicit markers such as #sarcasm or #irony were removed, as they function as annotation artefacts rather than authentic cues.
- Special characters. Non-standard symbols (e.g., @, $, ^) were removed, while punctuation relevant to sarcasm (e.g., question marks, exclamation marks) was preserved due to their role in rhetorical or exaggerated expression.
- Whitespace and repetition. Multiple spaces were collapsed into single spaces, and character repetitions were limited to two (e.g., “soooo funny” → “soo funny”), preventing rare variants from being treated as unique tokens while retaining expressive emphasis.

Linguistic standardisation. Contractions were expanded (e.g., don’t → do not) to ensure consistency in representing negation. This step was particularly important given the role of polarity shifts in sarcasm. A negation propagation scheme was also applied, in which tokens following a negation (e.g., not happy) were suffixed with _NEG (→ not happy_NEG). This provided models with explicit cues regarding polarity inversion, one of the most common linguistic mechanisms of sarcasm.

Final quality assurance. Only English-language entries with valid alphabetic tokens were retained. No further exclusions were required at this stage. Following these procedures, the dataset was reduced to 950,694 entries. The resulting corpus balanced cleanliness with fidelity, preserving stylistic variation and sarcasm markers while removing noise and inconsistencies. This created a consistent and reliable foundation for the subsequent sarcasm detection experiments.

## *3.4 Handling Class Imbalance*

The combined dataset was nearly perfectly balanced after cleaning, with 49.2% non-sarcastic and 50.8% sarcastic examples. As a result, no explicit rebalancing procedures were required for the majority of experiments. Retaining this natural balance avoided the introduction of synthetic data and preserved the linguistic authenticity of the corpus.

Imbalance became relevant only in the context of domain-level analysis. While Reddit and news headlines remained relatively balanced, the Twitter subset was heavily skewed (76.5% non-sarcastic vs. 23.5%

sarcastic). To mitigate this, domain-specific experiments on Twitter were conducted using an artificially balanced subset: all sarcastic tweets were retained, and an equal number of non-sarcastic tweets were sampled. Each training seed drew on a different set of non-sarcastic examples, reducing sampling bias and improving robustness. For comparability, Reddit and news headline data were also downsampled to equivalent sizes under the same per-seed randomisation protocol.

This adjustment was applied exclusively for domain-level experiments. In this way, the study leveraged the advantages of the overall balanced corpus for general model development, while also accounting for domain-specific skew when evaluating cross-domain performance.

## *3.5 Data Splitting Strategies*

To enable fair and robust evaluation, the dataset was stratified into 70% training, 10% validation, and 20% test sets. Stratification preserved class balance, ensuring that both sarcastic and non-sarcastic examples were proportionally represented across splits. The training set was used to optimise parameters, the validation set for early stopping and hyperparameter tuning, and the test set for unbiased assessment of generalisation. K-fold cross-validation was not employed due to the computational cost of repeatedly fine-tuning large transformer models. Instead, methodological consistency was maintained across all model families, from classical baselines to BiLSTMs and transformers. To further reduce variance from stochastic training, each model was run with multiple random seeds, providing more reliable estimates of performance without incurring the prohibitive cost of full cross-validation.

## *3.6 Exploratory Data Analysis (EDA)*

Following data cleaning and partitioning, exploratory analysis was conducted to examine structural and linguistic properties of the dataset. The aim of this stage was to characterise variation across domains and to identify the cues that models would need to capture in order to achieve robust sarcasm detection, particularly in the context of deployment within a browser extension.

**Text length:** Considerable variability in text length was observed across domains. Reddit posts initially contained extreme outliers, in some cases exceeding 10,000 characters, while Twitter rarely surpassed a few hundred. After filtering, the maximum sequence length across all sources was reduced to 472 characters (≈38 tokens), and the median stabilised at approximately 49 characters (≈9 tokens). The majority of texts fell between 5 and 30 words. Manual inspection of excessively long entries indicated that they were linguistically uninformative, often consisting of repeated tokens (e.g. “Barca! Barca! Barca! …”), copied discussion threads, or extended rants in which sarcastic content appeared only in isolated fragments. To address this, a maximum sequence length of 40 tokens was adopted. This threshold preserved more than 99% of meaningful content while excluding outliers, thereby improving tokenisation efficiency, reducing unnecessary padding in transformer models, and mitigating vanishing gradient risks in BiLSTMs.

**Stylistic markers:** Sarcasm often relies on paralinguistic cues such as punctuation, emojis, and hashtags, and the frequency of these markers varied substantially by domain. On Reddit, quotation marks appeared in nearly half of all posts (≈47%), while exclamation marks were considerably more frequent in sarcastic than non-sarcastic examples (13.8% vs. 5.6%). On Twitter, emojis appeared in over 20% of non-sarcastic tweets but only 10% of sarcastic ones, making them unreliable predictors; accordingly, emojis were removed from the dataset to prevent misleading signals. In contrast, sarcastic tweets exhibited higher proportions of question marks (15.5% vs. 8.3%). News headlines rarely contained explicit markers (<1%), reflecting their editorial constraints. In light of these observations, punctuation such as “!” and “?” was retained, as these provided consistent, domain-general cues for sarcasm detection.

**Lexical patterns:** Frequent unigrams and bigrams further illustrated domain-specific stylistic variation. Sarcastic texts often employed superficially positive words (e.g. “great”, “awesome”, “sure”) in ironic collocations (“oh great”, “yeah right”), whereas non-sarcastic texts favoured factual or descriptive constructions such as “years ago”, “study finds”, or “actually good”. Distinct lexical profiles emerged across domains: Reddit posts emphasised conversational markers (e.g. “yeah”, “just”, “don’t know”), Twitter was characterised by exaggerated positivity (e.g. “genuinely amazing”, “best experience”), and news headlines were dominated by named entities (e.g. “Donald Trump”, “White House”). These findings highlight that sarcasm detection requires not only modelling general ironic constructions but also adapting to domain-specific stylistic contexts.

# Chapter 4: Methodology

## 4.1 Research Design and Rationale

The research design was guided by two objectives: *(i)* to compare a wide range of models for sarcasm detection, and *(ii)* to align evaluation with the requirements of real-world deployment in the SarcQuest browser extension. A diverse set of models was chosen to assess the trade-off between predictive accuracy and inference latency, a critical factor for deployment in resource-constrained, real-time environments.

Three tiers of models were implemented: *(1)* classical baselines (Logistic Regression, Support Vector Machines) to provide interpretable benchmarks; *(2)* deep learning architectures (BiLSTM, CNN, BiLSTM with Attention) to capture sequential and contextual cues; and *(3)* transformer-based models (BERT, RoBERTa, DistilRoBERTa, and a variant with emotional embeddings) to leverage attention mechanisms and large-scale pretraining.

Evaluation emphasised both predictive quality—measured via accuracy and macro-F1 averaged across three random seeds—and deployment feasibility, assessed through inference latency and model size. Additional analyses included misclassification inspection to identify recurring error patterns, and per-domain evaluation (Reddit, Twitter, News) to assess robustness under stylistic variation.

This design ensured comparability across modelling families, balancing state-of-the-art accuracy with efficiency trade-offs and domain generalisation challenges that are critical for practical deployment.

## 4.2 Model Architectures and Training Setup and Hyperparameters

To comprehensively evaluate approaches to sarcasm detection, this study implemented a range of model architectures spanning traditional machine learning, deep neural networks, transformer-based encoders, and an extended dual-encoder variant. This tiered approach allowed for direct comparison of baseline interpretability and efficiency against the representational power of modern deep learning models, while also investigating the added value of emotion-aware embeddings.

### 4.2.1 Classical Machine Learning Models

As a lightweight, interpretable baseline, Logistic Regression (LR) was trained on TF–IDF representations with 1–2-gram features and a cap of 10,000 terms. The TF–IDF vectorizer was fitted only on the training split to avoid leakage and then applied to validation and test sets. Where available, a small set of engineered features—negation_count, exclamations, question_marks—was appended to the sparse TF–IDF matrix; these were standardised and concatenated to preserve sparsity. The LR model used the lbfgs solver with L2 regularisation and max_iter = 2000, which provided stable convergence on the high-dimensional feature space.

To ensure comparability with later models, LR followed the same protocol: a 70/10/20 stratified split with fixed indices for reproducibility; multi-seed runs {0,1,2}; and threshold tuning on the validation set (grid from 0.05 to 0.95) to maximise F1 before scoring on the test set. Results for LR were aggregated across seeds as mean ± standard deviation and stored alongside all artefacts (vectorizer, scaler, model checkpoints, calibration bins, per-seed reports). This configuration establishes a robust, computationally inexpensive reference point against which the more complex neural models can be interpreted.

### 4.2.2 Deep learning approaches

To represent pre-transformer neural methods, three architectures were implemented: Bidirectional LSTMs (BiLSTMs), BiLSTMs with Attention, and Convolutional Neural Networks (CNNs). These were selected because they are widely used in sarcasm detection and capture complementary cues (Sinha & Choudhary, 2023). BiLSTMs model sequential dependencies, attention mechanisms highlight salient tokens, and CNNs capture local n-gram patterns. Together, they provide a robust baseline for comparison with transformer-based models.

All models were initialised with pre-trained GloVe embeddings (100d and 300d), enabling evaluation of whether higher-dimensional vectors offered measurable benefits. Embeddings were frozen during training to preserve general semantic knowledge and reduce overfitting. Inputs were truncated or padded to a maximum of 40 tokens, derived from exploratory analysis to ensure >99% coverage while excluding noisy outliers.

Each BiLSTM used 64 hidden units per direction, balancing modelling capacity with efficiency. Dropout (0.5 after the recurrent layer, 0.3 after the dense layer) was applied to mitigate overfitting, which is particularly important as sarcastic examples are relatively sparse compared to literal ones (Khodak et al., 2018). A dense layer of 32 units preceded the final sigmoid classifier. The attention variant employed an additive mechanism to selectively weight tokens that are especially indicative of sarcasm (e.g., exaggerated modifiers or ironic markers), with the recurrent backbone kept identical to the baseline BiLSTM **(Ghosh & Veale, 2017)**.

CNNs were included as a lightweight alternative. While they lack sequential memory, they excel at detecting fixed n-gram patterns common in sarcastic text, such as “yeah right” or “great job” (Poria et al., 2016). The adopted architecture used parallel filters of size 3, 4, and 5 (128 each), followed by global max pooling, concatenation, and a dense layer of 64 ReLU units with dropout. This design provided an effective trade-off between accuracy and computational efficiency, aligning with deployment requirements such as browser extensions.

For consistency, all models followed a unified training protocol: 70/10/20 train/validation/test split, batch size of 128, maximum of 20 epochs, and early stopping with patience = 1 (restoring best weights). This setup ensured class balance across splits, efficient training, and prevention of unnecessary overfitting.

### 4.2.3 Transformer Models

To complement the recurrent and convolutional baselines, three single-encoder transformer models and one dual-encoder variant were evaluated: BERT (bert-base-uncased), RoBERTa (roberta-base), DistilRoBERTa (distilroberta-base), and a DistilRoBERTa + GoEmotions dual-encoder. The single-encoder models provide a like-for-like comparison across mainstream architectures that generate contextualised token and sequence representations, while the dual-encoder explores whether injecting auxiliary affective signals benefits sarcasm recognition.

#### *Single-encoder models (BERT, RoBERTa, DistilRoBERTa)*

Each model was fine-tuned as a binary classifier with a softmax layer over the [CLS] representation. Tokenisation used the corresponding pretrained tokenizer; sequences were truncated/padded to 128 tokens to cover the post-cleaning length distribution while controlling memory and latency. Training was conducted under a consistent protocol: three fixed epochs (based on earlier experiments where early stopping also converged within this range), the AdamW optimiser with a learning rate of 2e-5, a linear warm-up over 10% of steps, mini-batches of size 16 for training and 32 for evaluation, and three random seeds {0, 1, 2} to account for stochastic variation. These settings reflect standard practice for transformer fine-tuning and were chosen to keep compute comparable across backbones while allowing stable convergence within a fixed

budget. Inference latency was measured as milliseconds per sample on a fixed probe input to reflect deployment constraints.

- BERT establishes a baseline using bidirectional self-attention with WordPiece tokenisation (Devlin et al., 2019).
- RoBERTa retains the same architecture but benefits from training refinements and more data (Liu et al., 2019).
- DistilRoBERTa offers a lighter student model that preserves most language understanding performance with reduced parameters, making it attractive for real-time use (Sanh et al., 2019).

*Dual-encoder: DistilRoBERTa + GoEmotions*

To test whether affective cues help detect ironic polarity shifts, a dual-encoder was built by concatenating the [CLS] embeddings from DistilRoBERTa and a compact GoEmotions encoder trained for multi-label emotion recognition. A linear classifier operates on the concatenated vector. This setup preserves modularity (each encoder processes the same input), keeps memory in check (gradient checkpointing and AMP enabled when available), and allows the sarcasm head to exploit cues correlated with affective stance. GoEmotions provides a taxonomy of fine-grained emotions commonly used to probe sentiment and pragmatic tone (Demszky et al., 2020).

Training used early stopping (patience = 1) with a cap of 4 epochs, AdamW at 2e-5, warm-up 10%, max length 128, and batches 8/16 (train/eval) to avoid OOM when running two encoders. As with single-encoders, results are reported over three seeds.

### *4.2.3 Domain-Specific analysis*

To assess cross-domain robustness under controlled conditions, I ran a dedicated set of balanced, equal-size experiments for Twitter, Reddit, and News. The procedure mirrors the code pipeline and is summarised below.

**Balancing and equalisation:** For each domain and for each of three balanced variants (seeds {0,1,2}), the data were first downsampled to 50/50 class balance (sarcastic vs. non-sarcastic) using a domain-specific random draw. To prevent the largest domain from dominating, domains were then equalised to the same total size, capping each to the minimum balanced count observed across domains for that seed. Within the cap, both classes were downsampled to cap/2. This yields equal-sized, class-balanced datasets per domain and seed.

**Splits:** Each balanced-and-equalised domain dataset was split 70/20/10 into train/validation/test (stratified), producing three comparable test beds per seed.

**Models:** The following single-encoder transformers were fine-tuned independently within each domain: BERT (bert-base-uncased), RoBERTa (roberta-base), and DistilRoBERTa (distilroberta-base).

This selection spans the accuracy–latency spectrum and aligns with the main study's deployment focus. Training setup. Sequences were truncated/padded to 128 tokens. Optimisation used AdamW (lr=2e-5) with linear warm-up (10%), batch sizes 16/32 (train/eval), and exactly 3 epochs. Randomness was controlled via set_all_seeds. All hyperparameters are held fixed across models and domains to isolate domain effects.

*Evaluation Metrics*

The evaluation framework was designed to reflect both classification performance and deployment feasibility, ensuring that results were not only statistically valid but also practically meaningful for real-world integration into the SarcQuest browser extension.

#### *Classification Metrics*

Standard supervised learning metrics were computed, including accuracy, precision, recall, and F1-score. While accuracy provides a global measure of correctness, it can mask systematic errors when class distributions are balanced. Therefore, macro-averaged F1 was emphasised as the primary indicator of classification quality, since it weights sarcastic and non-sarcastic classes equally. In addition, the Area Under the Precision–Recall Curve (AUPRC) was reported, as it provides a more robust view of performance under conditions of class imbalance, which are common in naturalistic domains such as Twitter.

#### *Calibration Metrics*

Beyond classification, reliable probability estimates are crucial in real-world systems where decisions (e.g., highlighting text) depend on model confidence. To this end, two complementary measures were employed: Expected Calibration Error (ECE), quantifying the deviation between predicted confidence and empirical accuracy, and the Brier Score, capturing the mean squared error of predicted probabilities. Post-hoc temperature scaling was also applied as a simple yet effective method to improve probability calibration, yielding confidence scores better aligned with true likelihoods.

#### *Efficiency Metrics*

Since SarcQuest operates as a browser extension, evaluation also included latency (milliseconds per sample) and model footprint (MB). These metrics are vital for maintaining an interactive user experience. Transformer-based models were therefore benchmarked both on GPU and CPU, with latency thresholds defined at <10 ms/sample (GPU) and <50 ms/sample (CPU). Model size was also restricted to ~500 MB to ensure compatibility with lightweight deployment environments such as Hugging Face Spaces.

#### *Error Categorisation*

To understand systematic weaknesses, misclassified examples were categorised according to linguistic markers. Regular expressions were applied to tag the presence of negation, exclamation marks, hyperbolic expressions, and overconfidence markers. This provided qualitative insight into which rhetorical devices were most challenging for models, highlighting failure cases where surface cues conflicted with pragmatic meaning.

### *4.3.5 Statistical Significance Testing*

To verify that observed performance differences were not due to random variation, two statistical tests were applied: paired bootstrap resampling, which compared F1-score distributions across models, and McNemar's test, which directly examined the overlap in misclassification patterns. Together, these ensured that claims of superiority or robustness were statistically valid rather than anecdotal.

## *4.4 Deployment Pipeline (SarcQuest Extension)*

This section outlines the end-to-end methodology used to operationalise sarcasm detection in a real-time browser setting, from inference serving to feedback-driven model updates.

### *4.4.1 Inference service.*

A FastAPI application (hosted on Hugging Face Spaces) exposes three endpoints: /predict (batched inference), /feedback (user-label capture), and /reload (hot model swap). The service initialises a Hugging Face pipeline from a versioned model path and applies a stable label map to normalise heterogeneous classifier outputs. A fixed confidence threshold is enforced server-side to support deterministic decision rules. CORS is enabled to permit browser access.

### 4.4.2 Client orchestration

The extension's background service worker is the single point of interaction with the API. It implements: (i) candidate construction by requesting sentence spans from a content script; (ii) input normalisation (whitespace collapse, truncation to 800 characters); (iii) request control via batching (size 20) and timeouts (15 s); (iv) response normalisation to harmonise label/score fields; and (v) decision logic using a client threshold (0.65) to forward only positive candidates for rendering. The content script is render-only and guarded against reinjection; it underlines sentences and attaches inline controls for feedback without modifying page semantics.

### 4.4.3 Continual learning and model promotion and Operational guarantees

A CPU-only fine-tuning script ingests feedback and constructs a training set with importance weighting: disagreements are up-weighted, and near-uncertain agreements (score in [0.45, 0.65]) are duplicated to target the decision boundary. Tokenisation and training follow standard Hugging Face Trainer defaults, starting from the currently deployed checkpoint. Outputs are saved to timestamped directories to enable explicit versioning. An orchestration script enforces training policy (minimum total/new votes and a cooldown interval), triggers fine-tuning when criteria are met, and then calls /reload to hot-swap the API to the new checkpoint. Older checkpoints are pruned, while rollback remains trivial by reloading any previous directory.

This design separates concerns—UI rendering, orchestration, inference, and learning—so that (a) UI latency is managed by batching and truncation, (b) inference is stateless and easily reloadable, and (c) model evolution is governed by clear promotion criteria grounded in real user feedback.

## 4.5 Implementation Environment

Experiments were conducted across local and cloud-based environments. Data cleaning, exploratory analysis, and training of classical and deep learning baselines were performed in Jupyter Notebook on a local machine, while transformer models were fine-tuned in Google Colab Pro with NVIDIA A100 and L4 GPUs (16–24 GB VRAM). This split was required due to the high computational cost of transformer training. Hardware constraints shaped the experimental design: GPU memory limits required capping sequence length, tuning batch sizes, and enforcing fixed epochs. Time constraints favoured smaller, efficiency-oriented models and parameters over more resource-intensive ones.

The software stack included Python 3.12, scikit-learn (classical models), TensorFlow/Keras (BiLSTM, CNN), and PyTorch with Hugging Face Transformers (for fine-tuning and deployment). Data handling relied on pandas and NumPy. Random seeds were fixed for reproducibility, stratified splits were consistent across experiments, and version control was maintained through GitLab. Outputs such as checkpoints, predictions, calibration tables, and evaluation reports were stored in structured directories.

For deployment, fine-tuned models were exported in Hugging Face-compatible format (pytorch_model.bin + tokenizer files) and served via a FastAPI backend on Hugging Face Spaces. This API integrated directly with the SarcQuest Chrome Extension, implemented in HTML, CSS, and JavaScript, which performed client-side text extraction, API communication, and highlighting of sarcastic sentences.

# Chapter 5: Results

## 5.1 Overview

This chapter presents the evaluation of sarcasm detection models under a controlled protocol, comparing deep learning and transformer architectures across Reddit, Twitter, and news data using balanced splits and multiple seeds. Performance was assessed through classification metrics, calibration, efficiency, and model size, reflecting both accuracy and deployment feasibility. Domain analyses and misclassification breakdowns (e.g., negation, exclamations, hyperbole) revealed strengths and weaknesses, while qualitative examples highlighted challenges with implicit sarcasm.

## 5.2 Model Effectiveness and Reliability Assessment

The predictive performance of the evaluated models was assessed using the mean and standard deviation of F1-score, accuracy, precision, and recall across three experimental seeds. These four metrics were selected to capture complementary aspects of classification performance: accuracy reflects overall correctness, precision measures reliability of positive predictions, recall evaluates the ability to detect sarcastic instances, and F1 balances the trade-off between precision and recall. *Table 5.1* summarises the results for baseline, deep learning, and transformer models.

*Table 5.1: Summary of Model Performance Across Baselines, Deep Learning, and Transformer Architectures*

**Model Performance Summary (3-seed means)**

| | Model | Category | F1 | Accuracy | Precision | Recall | ECE | Brier | Latency (ms/sample) | Size (MB) |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Logistic Regression (TF-IDF) | Baseline | 70% | 70% | 72% | 68% | 0.20 | 0.19 | 0.03 | 0 |
| 1 | BiLSTM + Attention (100d) | DL | 70% | 71% | 74% | 66% | 0.22 | 0.19 | 0.18 | 0 |
| 2 | BiLSTM + Attention (300d) | DL | 71% | 71% | 74% | 68% | 0.23 | 0.19 | 0.17 | 0 |
| 3 | BiLSTM + GloVe (100d) | DL | 70% | 71% | 74% | 67% | 0.22 | 0.19 | 0.14 | 0 |
| 4 | BiLSTM + GloVe (300d) | DL | 71% | 71% | 74% | 68% | 0.23 | 0.19 | 0.16 | 0 |
| 5 | CNN + GloVe (100d) | DL | 69% | 70% | 72% | 66% | 0.18 | 0.20 | 0.07 | 0 |
| 6 | CNN + GloVe (300d) | DL | 69% | 70% | 72% | 67% | 0.18 | 0.19 | 0.19 | 0 |
| 7 | BERT (base) | Transformer | 77% | 76% | 76% | 78% | 0.35 | 0.17 | 7.03 | 419 |
| 8 | DistilRoBERTa | Transformer | 77% | 77% | 77% | 77% | 0.32 | 0.16 | 4.02 | 318 |
| 9 | DistilRoBERTa + Emotion | Transformer | 76% | 76% | 76% | 77% | 0.30 | 0.17 | 4.21 | 1138 |
| 10 | RoBERTa (base) | Transformer | 78% | 77% | 77% | 78% | 0.33 | 0.16 | 7.38 | 480 |

*This table reports macro-F1, accuracy, precision, recall, Expected Calibration Error (ECE), Brier score, average inference latency, and model size for all evaluated models. Results are averaged across three random seeds. Warm (coral) shading indicates stronger performance; cool (blue) shading indicates weaker performance. For ECE, Brier, and latency, lower values are better. Best results per column are outlined.*

### 5.2.1 Comparative Analysis of Predictive Metrics

Table 5.2 aggregate pairwise model comparisons across deep learning and transformer baseline s.

#### *Baseline Performance*

The logistic regression classifier with TF–IDF features served as a simple reference point. It achieved an F1-score of 70% (±0.0), with precision at 72% and recall at 67%. These results demonstrate balanced but limited predictive power, confirming that linear baselines can capture surface lexical cues but struggle with the deeper semantic and contextual signals of sarcasm.

### *Deep Learning Architectures*

Deep learning models provided incremental improvements over the baseline. BiLSTM with GloVe embeddings achieved F1-scores of 70–71%, with the 300-dimensional embedding outperforming the 100-dimensional variant. Standard deviations were small (≈±0.6), indicating consistent results across seeds. BiLSTM with attention reached similar performance, achieving an average F1-score of approximately 71% with a standard deviation of ±1.1%. This indicates that while the model was reasonably consistent across runs, the inclusion of the attention mechanism did not substantially enhance discrimination in this configuration. The CNN models performed slightly worse, with F1-scores averaging around 69% and a standard deviation of ±0.8%, aligning them more closely with the logistic regression baseline and confirming their weaker ability to capture the nuances of sarcastic expression. As shown in Table 5.2, pairwise significance testing confirmed that BiLSTM and CNN variants differed only marginally, with most ΔF1 values below 2%, indicating that no deep learning approach delivered a decisive improvement over the others**.** Overall, the deep learning models demonstrated similar results as baseline models but failed to close the gap to the transformer models.

### *Transformer Models*

Transformer-based architectures consistently achieved the highest performance and demonstrated low variance across experimental seeds, underlining their robustness. BERT (base) achieved an average F1-score of 77% with a standard deviation of ±0.1. This very small deviation suggests that BERT's performance was highly stable, with minimal sensitivity to random initialisation. Precision and recall were balanced at 76% and 78% respectively, indicating that the model maintained consistency in detecting sarcasm without sacrificing reliability.

DistilRoBERTa reached a mean F1-score of 77% with an exceptionally low standard deviation of ±0.0, reflecting virtually identical outcomes across all seeds. This stability, combined with its reduced size and faster inference time, makes DistilRoBERTa particularly attractive for deployment in real-time systems, as it offers both efficiency and reproducibility. RoBERTa (base) produced the strongest overall results, with a mean F1-score of 78% and a standard deviation of ±0.1. The model also achieved the highest recall at 78%, confirming its superiority in reliably identifying sarcastic instances. The low deviation indicates that this improvement is not incidental but consistently replicable, further reinforcing RoBERTa's robustness as the best-performing model in the suite.

The extended DistilRoBERTa configuration with emotion embeddings achieved an F1-score of 76% with a slightly higher standard deviation of ±0.2. Although this result is marginally lower than the other transformer models in terms of overall accuracy, the higher recall of 77% demonstrates its value in prioritising sensitivity over precision. The slightly greater variability across runs suggests that incorporating emotion features introduces additional complexity, which may cause the model to be more sensitive to shorter texts. However, the gain in recall highlights its practical utility in user-facing scenarios, where failing to detect sarcasm (false negatives) is often more detrimental than occasional false positives.

Beyond descriptive averages, pairwise statistical testing confirmed that these differences were systematic rather than incidental. Across three seeds, RoBERTa significantly outperformed both DistilRoBERTa (ΔF1=+0.88 points; Stouffer-combined bootstrap p<1e-6) and BERT (ΔF1=+1.07 points; p<1e-6). Even the small advantage of DistilRoBERTa over BERT (+0.19 F1 points) was statistically detectable (p≈1e-4), although the effect size was negligible in practice. McNemar's exact test on discordant predictions further showed that when models disagreed, RoBERTa was more often correct, confirming that its advantage reflected consistent error reduction rather than chance.

The ranking is robust: RoBERTa > DistilRoBERTa > BERT. RoBERTa is best for raw accuracy. DistilRoBERTa is nearly as good, but far faster and lighter, making it the most practical option for real-time use. BERT consistently trailed both models and offered no tangible advantage.

*Comparative Analysis*

Across all metrics, transformer models substantially outperformed both logistic regression and deep learning baselines, with absolute F1 improvements of 7–8 percentage points over the baseline. Precision and recall remained balanced across categories, but transformers consistently provided the strongest recall, which is critical for sarcasm detection tasks.

These results establish transformer architectures, particularly RoBERTa and DistilRoBERTa, as the most reliable models for sarcasm detection, combining high predictive performance with computational efficiency. *Section 5.2.2* extends this analysis by examining the calibration of model confidence scores.

## *5.2.2 Calibration*

Beyond classification accuracy, it is important to assess whether models produce well-calibrated probability estimates. In the context of sarcasm detection, calibration reflects the degree to which predicted confidence values align with the actual likelihood of correctness. For example, a perfectly calibrated model that outputs a probability of 0.80 for a given sentence being sarcastic should be correct approximately 80% of the time. Poor calibration can undermine user trust in a deployed system, particularly in real-time applications such as browser extensions, where threshold-based highlighting depends on reliable probability estimates.

Calibration was evaluated using Expected Calibration Error (ECE) and the Brier score. ECE measures the average discrepancy between predicted probabilities and observed accuracies across discrete bins, with lower values indicating better calibration. The Brier score similarly quantifies the mean squared difference between predicted probabilities and true labels, combining aspects of both calibration and refinement.

The logistic regression baseline achieved an ECE of 0.20 and a Brier score of 0.19, suggesting moderately well-calibrated probabilities despite its limited discriminative capacity. The deep learning models demonstrated mixed results: BiLSTM variants produced ECE values of approximately 0.22 with Brier scores near 0.19, while CNN models achieved slightly lower ECE scores (≈0.18) but higher Brier scores (≈0.20). These outcomes indicate that, although CNNs provided marginally better alignment between predicted confidence and observed accuracy, they did not translate this into stronger predictive performance.

Transformer models showed a more complex pattern. BERT (base) achieved a Brier score of 0.17 but suffered from relatively poor calibration with an ECE of 0.35, indicating that while its raw predictions were strong, its confidence estimates tended to be misaligned with true probabilities. DistilRoBERTa achieved slightly better calibration, with an ECE of 0.32 and a Brier score of 0.16, while RoBERTa (base) exhibited similar trends, producing strong accuracy but with overconfident predictions (ECE = 0.33). The addition of emotion embeddings in DistilRoBERTa improved calibration modestly, reducing ECE from 0.30 (raw) to 0.25 after applying temperature scaling, and lowering the Brier score from 0.166 to 0.162. This highlights the utility of post-hoc calibration techniques for refining confidence outputs, even in already high-performing transformer architectures.

Overall, these findings emphasise that high predictive accuracy does not necessarily entail good calibration. Transformer models, while achieving the strongest F1-scores, consistently demonstrated a tendency towards overconfidence. In practical terms, this means that while they excel at identifying sarcasm, their reported probabilities should not be interpreted at face value without calibration. This has direct implications for deployment in the SarcQuest browser extension, where decisions such as highlighting sentences above an

80% sarcasm threshold rely on trustworthy probability estimates. By applying post-hoc calibration techniques such as temperature scaling, the models' probability outputs can be better aligned with observed accuracies, increasing the reliability and usability of the system in real-world settings.

### 5.2.3 Latency

Inference latency was assessed in order to evaluate the feasibility of deploying the models in real-time environments. Latency was measured as the average time taken to classify a single sentence, reported in milliseconds per sample. The results show a clear trade-off between model complexity and response time. The logistic regression baseline was by far the fastest, requiring only 0.03 ms per sample, followed by the CNN models at approximately 0.07–0.19 ms, reflecting their lightweight architecture. BiLSTM variants were slower, ranging from 0.14 to 0.18 ms per sample, with training times also considerably longer due to sequential processing overheads.

The transformer models exhibited higher latency, with BERT and RoBERTa requiring around 7 ms per sample, while DistilRoBERTa reduced this to approximately 4 ms, representing a substantial efficiency gain without compromising performance. The DistilRoBERTa + Emotion configuration added minimal overhead, maintaining latency near 4.2 ms. Although transformers are slower than traditional models, their latency remains within acceptable limits for browser-based deployment, where feedback within a fraction of a second is sufficient for a seamless user experience.

## 5.3 Missclassification and Domain Analysys

### 5.3.1 Missclasification Analysis

While aggregate accuracy and F1-scores provide an overall estimate of model performance, they mask systematic weaknesses in handling specific linguistic phenomena that are central to sarcasm. Sarcasm frequently relies on cues such as negation, exclamatory emphasis, hyperbolic exaggeration, or overconfident tone, all of which can invert or distort literal sentiment (Khodak et al., 2018). A focused error analysis across these categories therefore offers valuable insight into how different architectures capture (or fail to capture) the mechanisms of sarcastic expression. . *Table 5.3* summarises the results of the Missclasification Analysis and *Figure 5.2* shows some missclasification examples the particular example is from Logisitic Regression

*Table 5.3: Misclassification Summary Across Baselines, Deep Learning, and Transformer Architectures*

**Misclassification Summary (3-seed totals)**

| Model | Category | Total Errors | Negation | Exclamation | Hyperbole | Overconfidence |
|---|---|---|---|---|---|---|
| **0** Logistic Regression (TF-IDF) | Baseline | 169,830 | 37,083 (21.8%) | 13,560 (8.0%) | 31,689 (18.7%) | 3,207 (1.9%) |
| **1** BiLSTM + Attention (100d) | DL | 165,432 | 36,428 (22.0%) | 15,129 (9.1%) | 31,013 (18.7%) | 4,781 (2.9%) |
| **2** BiLSTM + Attention (300d) | DL | 163,247 | 35,793 (21.9%) | 14,720 (9.0%) | 30,450 (18.7%) | 5,881 (3.6%) |
| **3** BiLSTM + GloVe (100d) | DL | 165,358 | 36,257 (21.9%) | 14,984 (9.1%) | 31,042 (18.8%) | 5,379 (3.3%) |
| **4** BiLSTM + GloVe (300d) | DL | 162,947 | 35,729 (21.9%) | 14,784 (9.1%) | 30,637 (18.8%) | 5,860 (3.6%) |
| **5** CNN + GloVe (100d) | DL | 173,581 | 38,324 (22.1%) | 15,420 (8.9%) | 32,496 (18.7%) | 2,708 (1.6%) |
| **6** CNN + GloVe (300d) | DL | 170,553 | 37,493 (22.0%) | 15,176 (8.9%) | 31,756 (18.6%) | 3,533 (2.1%) |
| **7** BERT (base) | Transformer | 135,993 | 29,169 (21.4%) | 10,129 (7.4%) | 6,742 (5.0%) | 1,312 (1.0%) |
| **8** DistilRoBERTa | Transformer | 133,124 | 28,665 (21.5%) | 10,016 (7.5%) | 6,574 (4.9%) | 1,346 (1.0%) |
| **9** RoBERTa (base) | Transformer | 129,036 | 27,586 (21.4%) | 9,793 (7.6%) | 6,309 (4.9%) | 1,275 (1.0%) |

*This table reports total misclassifications across three random seeds, broken down by linguistic phenomenon: negation, exclamatory markers, hyperbole, and overconfidence. Warm (coral) shading indicates higher error counts; cool (blue) shading indicates lower error counts. Best results per column are outlined. Percentages represent the proportion of each error type relative to total errors for that model.*

*Classic Models (BiLSTM, CNN, Logistic Regression)*

The traditional baselines produced high misclassification rates, with total errors exceeding 160k–173k across seeds. For instance, CNN+GloVe(100d) yielded 173,581 total misclassifications, of which 38,324 (22.1%) involved negation and 32,496 (18.7%) hyperbole. This indicates that nearly two-fifths of all CNN errors can be attributed to the two phenomena most closely tied to sarcasm, suggesting a structural limitation in the model's ability to exploit contextual inversion.

BiLSTM variants reduced errors modestly (≈162–165k total), but still exhibited negation error rates above 21% and hyperbole around 18–19% of their total errors. Logistic Regression with TF-IDF, despite being the simplest model, was competitive in some respects, producing 169,830 errors, with 37,083 negation (21.8%) and 31,689 hyperbole (18.6%). However, it underperformed on exclamatory cues, misclassifying 13,560 instances (8.0%), which highlights its limited sensitivity to stylistic punctuation.

Overall, the baselines demonstrate that while linear and recurrent models capture word-level sentiment shifts, they are unable to consistently model the interaction between lexical and contextual cues that drive sarcasm.

*Transformer Models (BERT, RoBERTa, DistilRoBERTa)*

Transformer architectures substantially reduced misclassification counts across all categories.

- BERT produced 135,993 total errors, with 29,169 (21.5%) due to negation and 10,129 (7.4%) to exclamatory markers.
- DistilRoBERTa achieved 133,124 errors, with 28,665 (21.5%) negation and 6,574 (4.9%) hyperbole misclassifications.
- RoBERTa was the strongest performer, with 129,036 total errors—a ~21% reduction compared to CNN+GloVe(100d)—and lower proportions of both negation (21.4%) and exclamatory errors (7.6%).

A striking finding is the reduction in overconfidence misclassifications, which fell from 2,700–5,800 in BiLSTM/CNN to only ~1,200–1,300 in transformers (≈60% fewer). This indicates that transformer models are not only more accurate, but also better calibrated in their probability estimates an essential property for deployment in real-time tools such as browser extensions, where misclassification accompanied by high confidence could severely undermine user trust.

*Importance of These Findings*

This error analysis demonstrates that misclassification is not uniformly distributed across linguistic cues. While transformers improve overall robustness, negation remains persistently difficult, accounting for over 21% of all errors in every model. This suggests that the semantic inversion inherent in sarcasm remains an open challenge even for state-of-the-art contextual models. Moreover, the reduction of hyperbole and exclamatory errors in RoBERTa (to under 5–7% of total errors) suggests that self-attention mechanisms better capture stylistic and rhetorical devices. This supports the claim that contextual embeddings enable models to process sarcasm not only as sentiment polarity but as a broader discourse phenomenon.

From a deployment perspective, this analysis underscores the practical reliability of transformer models. A system that systematically overpredicts sarcasm in negated sentences or exclamations risks alienating users. By contrast, the comparatively lower overconfidence of RoBERTa and DistilRoBERTa provides evidence that such models are both accurate and trustworthy in real-time environmen

### 5.3.2 Domain-Level Performance Analysis

Domain-level evaluation was conducted for the three transformer architectures BERT, RoBERTa, and DistilRoBERTa in order to assess how performance varied across different text sources.These models were chosen as they consistently outperformed the baselines and classical deep learning models on aggregate metrics, making them the most informative candidates for detailed domain analysis. Importantly, this analysis reflects per-domain performance when all models were trained on the overall combined dataset, rather than on domain-specific subsets. A complementary evaluation where each model is trained separately on individual domains is presented in *Section 5.4*. Refer to *Table 5.4* for the full results for Domain Analysis.

*Table 5.4: Domain-Level Performance of Transformer Models on News Headlines, Reddit, and Twitter.*

| Domain | BERT F1 | BERT Acc | BERT ECE | BERT Brier | RoBERTa F1 | RoBERTa Acc | RoBERTa ECE | RoBERTa Brier | DistilRoBERTa F1 | DistilRoBERTa Acc | DistilRoBERTa ECE | DistilRoBERTa Brier |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| News Headline | 0.861 | 0.877 | 0.5 | 0.096 | 0.895 | 0.909 | 0.504 | 0.069 | 0.882 | 0.897 | 0.492 | 0.077 |
| Reddit | 0.766 | 0.758 | 0.34 | 0.174 | 0.777 | 0.77 | 0.329 | 0.162 | 0.768 | 0.763 | 0.313 | 0.165 |
| Twitter | 0.432 | 0.722 | 0.64 | 0.204 | 0.468 | 0.741 | 0.619 | 0.183 | 0.457 | 0.737 | 0.596 | 0.181 |

*This table reports F1-score, accuracy, Expected Calibration Error (ECE), and Brier score for three transformer architectures (BERT, RoBERTa, DistilRoBERTa) evaluated on each domain when trained on the full data set. Results are averaged across three seeds. Warm (red) shading indicates poorer performance; cool (blue) shading indicates stronger performance.*

The results reveal a consistent pattern across all three transformers. Performance was strongest on news headlines, where RoBERTa achieved an F1 of 0.895, DistilRoBERTa 0.882, and BERT 0.861. Accuracy was similarly high (0.877–0.909), and both calibration error and Brier score were relatively low, indicating reliable predictions. This outcome aligns with expectations, as headlines tend to be more grammatically regular and less context-dependent, which makes sarcasm easier to detect.

On Reddit, performance declined modestly, with F1 scores ranging from 0.766 (BERT) to 0.777 (RoBERTa). This is still strong performance given the greater lexical variety and pragmatic complexity of Reddit comments compared to headlines. The results suggest that transformers generalise reasonably well to semi-structured, user-generated text but encounter more nuanced forms of sarcasm that reduce accuracy and calibration.

The most challenging source was Twitter, where all models underperformed. RoBERTa achieved the highest F1 (0.468), followed closely by DistilRoBERTa (0.457) and BERT (0.432). While accuracy remained moderate (≈0.72–0.74), calibration error was highest in this domain, showing that models were often miscalibrated in their confidence on short, noisy texts. These findings align with expectations, as Twitter posts are short, often contain creative orthography, and rely heavily on cultural or conversational context that is not captured in text alone. It is also important to note that the Twitter portion of the dataset was skewed (≈30% sarcastic, 70% non-sarcastic), and this imbalance was not rebalanced at the domain level. This skew likely contributed to the comparatively low F1-scores, as models were biased towards the majority non-sarcastic class.

Overall, the domain-level analysis confirms that transformer architectures can detect sarcasm reliably in structured and semi-structured text, but their performance diminishes considerably on short, imbalanced, and noisy social media data such as Twitter.

## 5.4 Per-Domain Training Analysis

While *Section 5.3* analysed domain-level performance by evaluating globally trained transformer models across news, Reddit, and Twitter subsets, this section takes a different approach: each source was trained and

evaluated in isolation. This ensures that models were fully optimised for in-domain distribution, balanced to 50/50 sarcastic and non-sarcastic, and equalised to the same number of examples across domains. The motivation for this design was to investigate whether domain-specific training provides advantages compared to cross-domain generalisation. Refer to *Table 5.5* for the full results for the Per-Domain Training Analysis

### *5.4.1 Baselines (Logistic Regression)*

The baseline Logistic Regression model showed substantial variation across domains. Performance was strongest on news headlines (F1 = 0.768, Acc = 0.760, Prec = 0.743, Rec = 0.795, Brier = 0.180), reflecting the relative regularity and shorter length of headlines. On Reddit, F1 dropped to 0.646, with reduced accuracy (0.648) and weaker calibration (Brier = 0.227). Twitter was again the most challenging, with F1 = 0.598 and Brier = 0.238, despite balancing and equalisation. These results confirm that even when trained in-domain, Twitter sarcasm remains the most difficult to model due to noisy, short, and context-heavy nature of the text. Variance across seeds was negligible, indicating stable but consistently poor performance on short and noisy Twitter texts.

### *5.4.2 Classical Deep Learning (BiLSTM, CNN)*

The classic DL models outperformed Logistic Regression overall but still followed the same difficulty hierarchy. On news headlines, the best-performing model was BiLSTM with attention and 300d embeddings (F1 = 0.760), with most architectures clustering between 0.73–0.76 F1. On Reddit, F1 dropped into the 0.55–0.63 range depending on the model, with CNNs showing relatively competitive performance despite their simpler structure. On Twitter, CNNs surprisingly performed best (CNN_GloVe100 F1 = 0.620), slightly outperforming BiLSTMs, suggesting that local n-gram style features may be more useful than long-range dependencies in extremely short texts. Across all DL models, calibration was weaker on Reddit and Twitter, with higher Brier scores and error counts, showing models were often overconfident in wrong predictions. Importantly, results on Twitter showed noticeably higher variance across seeds compared to news and Reddit, suggesting that DL models were unstable on small, imbalanced, and noisy datasets. By contrast, on news and Reddit, standard deviations were not significant, with performance consistent across runs.

### *5.4.2 Comparison with Transformers*

Then compared with the transformer results in *Section 5.3*, an important contrast emerges. Transformers trained globally and tested across sources achieved higher overall F1 and better calibration than both baselines and DL trained per-source. For example, RoBERTa achieved F1 = 0.868 on News and 0.777 on Reddit without domain-specific fine-tuning, while the best DL models reached only ~0.76 and ~0.63 respectively when trained exclusively in-domain. On Twitter, transformers also underperformed (RoBERTa F1 = 0.643), but still matched or slightly outperformed the best DL model (CNN_GloVe100 F1 = 0.620). Standard deviations were consistently low for transformers, indicating stable performance even on Twitter, in contrast to the variability observed in DL models.

### *5.4.3 Interpretation*

Domain-specific training provides limited benefit, as transformers generalise better across sources due to large-scale pre-training and contextual sensitivity. Baseline and deep learning models remain fragile under domain shifts, needing balanced in-domain data yet still failing to match transformer performance. Twitter poses persistent challenges for all models due to brevity, external context, and class imbalance. Overall, the analysis confirms transformers' robustness and the continued difficulty of detecting sarcasm in noisy social media text.

# Chapter 6: Discussion

## 6.1 Introduction

The results presented in Chapter 5 provided a comprehensive evaluation of sarcasm detection models, comparing their performance across domains, error types, and deployment constraints. This chapter moves beyond reporting metrics to consider what those findings mean in a broader context. It interprets the results in relation to the research aims, situates them within prior work, and draws out both theoretical and practical implications. In doing so, the chapter highlights not only the strengths of transformer-based approaches, but also the limitations that persist when sarcasm is deployed as a real-time detection task in everyday web environments.

## 6.2 Interpretation of Results

The results collectively indicate that effective sarcasm detection requires contextual modelling beyond surface lexical cues. Classical baselines (e.g. Logistic Regressionwith TF–IDF) established a useful point of reference, achieving respectable accuracy and balanced precision–recall, but their performance plateaued because they treat text as largely unordered features. These models capture explicit patterns (e.g., “yeah right”, “great job”) yet struggle when sarcasm is realised through polarity inversion, pragmatic contrast, or dispersed rhetorical markers. Pre-transformer neural models (BiLSTM, BiLSTM+Attention, CNN) offered only modest gains over baselines. Although recurrent encoders can model sequential dependencies and attention can emphasise salient tokens, the improvements remained incremental, implying that sequence memory alone is insufficient to capture the multi-cue, context-sensitive nature of sarcastic language.

Transformer architectures produced a decisive step change. BERT, RoBERTa, and DistilRoBERTa consistently outperformed all other families on macro-F1 with very low variance across seeds, underscoring their stability. The self-attention mechanism yields contextualised token representations that integrate heterogeneous signals—negation scope, hyperbolic modifiers, and discourse-level contrasts—within a single pass. Among these models, RoBERTa achieved the highest aggregate F1, but the performance gap to DistilRoBERTa was small. This near-parity is critical in an applied setting: while RoBERTa’s marginal advantage matters for leaderboard comparisons, DistilRoBERTa offered substantially lower latency and model footprint with effectively equivalent utility for real-time usage. The findings therefore distinguish between “best accuracy” and “best for deployment,” validating the choice of DistilRoBERTa as the operational model in SarcQuest.

Performance differences across domains reveal how text source and style condition detectability. Models were strongest on news headlines, where grammatical regularity and topic-driven irony provide clear cues within short, self-contained spans. Reddit comments, though more varied and dialogue-like, remained sufficiently rich for transformers to generalise, with only a modest decline in F1. Twitter proved the most challenging: utterances are short and noisy, pragmatic intent often depends on prior conversational turns or world knowledge, and (in the cleaned set) the removal of emojis/hashtags reduces overt paralinguistic signals. Residual class imbalance in Twitter further biases decision boundaries. This hierarchy—News > Reddit > Twitter—aligns with expectations that structure and length facilitate modelling, whereas brevity and implicit context hinder it. The implication for deployment is straightforward: SarcQuest can be more assertive on news-style prose and more conservative on short, informal microtexts.

Calibration analysis adds an important layer beyond accuracy. Raw transformer confidence scores tended to be overconfident (elevated ECE), meaning probability outputs did not perfectly track empirical correctness. Post-hoc temperature scaling improved alignment, and the emotion-augmented DistilRoBERTa variant modestly tightened calibration further. Because SarcQuest renders decisions via a probability threshold, calibrated scores are not merely an academic nicety; they directly influence user trust and perceived reliability. A threshold that is tuned on calibrated probabilities reduces both frustrating false positives (unwarranted highlighting) and missed positives at the decision boundary.

Efficiency metrics reframe what constitutes a "good" model for this use case. While classical models are microsecond-fast, their predictive ceiling limits practical utility. Full-size transformers are accurate but heavier; in contrast, DistilRoBERTa achieved millisecond-scale inference with a markedly smaller footprint while preserving accuracy within approximately one F1 point of the best model. This establishes a pragmatic Pareto frontier—maximising combined accuracy, latency, and memory efficiency—on which DistilRoBERTa is optimal for interactive browser deployment.

The error typology clarifies residual weaknesses. Across models, misclassifications disproportionately involved negation and polarity inversion, which remain difficult even with contextual attention, suggesting explicit handling of negation scope or limited conversational context would be beneficial. By comparison, errors related to hyperbole and exclamatory punctuation were notably reduced in transformers relative to baselines, indicating improved sensitivity to stylistic rhetoric. Importantly, transformers also exhibited fewer high-confidence errors, a property that enhances trust in user-facing systems where erroneous but emphatic highlights are particularly damaging to credibility.

In synthesis, the evidence supports three conclusions relevant to the research aims. First, transformer encoders are necessary to achieve robust sarcasm detection across heterogeneous web text; earlier baselines and pre-transformer neural models provide value as references but do not approach the same level of contextual competence. Second, when constraints of real-time deployment are considered, DistilRoBERTa offers the most favourable accuracy–latency–size trade-off and is therefore the correct operational choice for SarcQuest. Third, domain sensitivity and calibration matter in practice: performance is highest on structured editorial prose and most fragile on short, context-dependent microtexts; probability outputs require calibration to support reliable thresholding. These interpretations both explain the observed results and delineate the contours of remaining work—namely, better modelling of negation and discourse context, calibrated decision policies, and targeted adaptation for microtext—necessary to further improve the reliability of sarcasm detection in real-world use.

## *6.3 Comparison with Prior Work*

The results of this study broadly align with previous research on sarcasm detection, while also extending prior work in several important ways.

### *6.3.1 Performance trends across model families.*

Consistent with earlier surveys, linear models such as Logistic Regression provided useful baselines but were limited in their ability to capture pragmatic inversion and context-sensitive cues (Joshi et al., 2017). Pre-transformer deep learning architectures, including CNNs and BiLSTMs, achieved only modest improvements, as also observed by Poria et al. (2016) and Ghosh and Veale (2016). The performance gap between these architectures and transformer-based models was substantial, reflecting the wider consensus that contextualised embeddings and self-attention provide a decisive advantage for sarcasm detection, with BERT and RoBERTa consistently outperforming earlier models (Savini & Caragea, 2022).

### 6.3.2 Transformer variants and efficiency trade-offs.

Within the transformer family, RoBERTa delivered the strongest predictive performance, while DistilRoBERTa achieved nearly identical F1-scores with substantially lower latency and smaller model size. Similar findings have been reported in other domains, where RoBERTa's refined pretraining objectives yield consistent gains over BERT (Liu et al., 2019), and distilled variants preserve most of this performance while reducing computational demands (Sanh et al., 2019). This study therefore confirms prior observations that distilled transformers strike an effective balance between accuracy and efficiency, and demonstrates that this balance holds in sarcasm detection, where DistilRoBERTa achieved near state-of-the-art performance while remaining lightweight enough for real-time deployment.

### 6.3.3 Domain sensitivity and context requirements.

The domain-level analysis revealed a consistent hierarchy models performed best on structured, professionally written news headlines, less well on Reddit comments, and worst on short, noisy tweets. This pattern mirrors findings from earlier research. Bamman and Smith (2015) showed that incorporating conversational context improves sarcasm detection on Twitter, while Wallace et al. (2014) demonstrated that both human annotators and machine learning models struggle when sarcastic intent cannot be inferred from isolated text. Performance on Twitter in this study (RoBERTa F1 = 0.468) is lower than the top systems in iSarcasm= (≈0.60–0.70 F1; Farha et al., 2022) and below context-aware transformers (≈0.79 F1; Dong et al., 2020). This discrepancy is explained by the absence of conversational history, exclusion of overt markers such as hashtags and emojis, and class imbalance in the cleaned Twitter subset. By contrast, performance on news headlines (RoBERTa F1 = 0.895) is similar with headline models, which report accuracies of 92–95% (Ardilla et al., 2024).

### 6.3.4 Label quality and annotation artefacts.

Differences in label quality also account for variation with prior benchmarks. Twitter datasets based on hashtags often inflate performance by introducing superficial cues (Oprea & Magdy, 2020). By removing such artefacts, the present study produced a more challenging but realistic evaluation setting. Similarly, Misra and Arora's (2019) headlines corpus benefits from clear labels and structured text, which partly explains the consistently high performance across studies in this domain.

### 6.3.5 Error patterns and linguistic mechanisms.

Misclassification analysis confirmed that negation and polarity inversion remain the most challenging phenomena, consistent with earlier linguistic accounts of sarcasm (Riloff et al., 2013). Errors linked to hyperbole and exclamatory markers were substantially reduced in transformer architectures, suggesting that self-attention mechanisms improve sensitivity to rhetorical devices compared to earlier feature-driven or sequential models.

### 6.3.6 Calibration and deployment reliability.

A distinctive contribution of this study lies in the inclusion of calibration metrics. While most sarcasm detection research reports only accuracy or F1, this study evaluated Expected Calibration Error and Brier scores, finding that transformer models were systematically overconfident—a well-established issue in deep learning (Guo et al., 2017). Post-hoc temperature scaling improved reliability, and an emotion-augmented variant of DistilRoBERTa reduced calibration error further. These findings highlight the practical importance of probability calibration for threshold-based decisions in real-time applications such as browser extensions.

Overall, the findings converge with prior work in demonstrating the superiority of transformer-based encoders and the central role of domain and context in sarcasm detection. They diverge by providing a systematic evaluation across multiple domains under a unified protocol, incorporating calibration metrics into the assessment, and validating efficiency–accuracy trade-offs in a real-time deployment setting. These contributions extend the literature by showing not only theoretical advances in sarcasm detection but also the operational feasibility of deploying such models in everyday web environments.

## *6.4 Practical Implications*

The deployment of the SarcQuest browser extension demonstrates that real-time sarcasm detection is technically feasible with current NLP models, provided that careful trade-offs are made between accuracy, latency, and memory efficiency. DistilRoBERTa was selected as the operational model because it offered a reduction of approximately 60% in size and inference time compared to RoBERTa and BERT, while incurring only marginal decreases in predictive performance. This decision reflects a broader finding of this study: in applied settings, the most suitable model is not necessarily the one with the highest offline F1 score, but the one that achieves a stable balance between predictive quality and responsiveness in resource-constrained environments.

A small-scale evaluation was performed directly through the extension. Six websites were scanned (two news articles, two Reddit threads, and two webpages explaining sarcasm), and the highlighted outputs were then manually checked. This allowed the number of correctly and incorrectly flagged instances to be counted, providing an indicative measure of how the model behaved under real browsing conditions.

The results revealed clear domain-specific variation, mirroring the trends observed in the controlled experiments of Chapter 5. On news articles, predictions were generally stable, with literal reporting producing no false positives. However, one case of reported speech (“When we have the children in every day the results are just better”) was incorrectly flagged as sarcastic, illustrating how quotations can confuse stance attribution (see *Figure 6.3*). On explanatory webpages, detection was moderate: out of 20 labelled sarcastic sentences in YourDictionary, only 7 were recognised and none were misclassified as sarcastic, while on another page about irony and sarcasm only 2 of 5 sarcastic examples were highlighted and none of the ironic ones were detected (*Figures 6.1–6.2*). This indicates that while explicit cues were sometimes captured, examples phrased outside the training distribution or overlapping with irony proved more difficult. On Reddit, performance dropped most sharply. Several short, literal utterances (“I’ve never heard of this before”, “Yes, that’s crucial”, etc.) were misclassified as sarcastic, illustrating the challenge of handling short, decontextualised remarks in informal discourse (*Figure 6.4*).

Analysing these outcomes highlights several important implications. First, the deployment confirms the same domain hierarchy observed in Chapter 5—news > explicit examples > informal dialogue—but with direct consequences for end-user experience. Misclassification of neutral quotations shows that sarcasm detection cannot always disambiguate between speaker content and intent, raising the risk of misinterpretation when context is limited. Similarly, the errors on short Reddit utterances underline a persistent challenge: without surrounding conversational turns, even a strong model defaults to false positives based on surface cues. These findings suggest that domain-general training alone is insufficient and that domain-specific fine-tuning or calibration may be required for robust real-world performance. Furthermore, extending the extension to read surrounding sentences and incorporate local conversational context would likely improve reliability, as sarcasm often relies on contrast or polarity shifts across multiple utterances rather than within a single sentence. This capability would reduce false positives in short texts and help capture sarcastic intent that emerges only in relation to neighbouring discourse.

The current build of the extension operates as stateless inference, meaning that each prediction is independent and the system cannot adapt based on past errors or user interaction. While the architecture already includes endpoints for feedback capture and automatic retraining, these features were disabled during evaluation to keep the deployment simple and privacy-safe. This represents a setback, as recurrent boundary errors (such as misclassifying short utterances or ambiguous quotations) cannot yet be corrected. However, the foundation for these mechanisms is already implemented, and enabling them is planned as part of future work, where user feedback will allow the system to iteratively refine its predictions and improve reliability across domains.

Beyond technical feasibility, real-time sarcasm detection has potential benefits in applied contexts. Browser-based assistance could support neurodivergent readers who may find pragmatic cues difficult to interpret, aid English language learners in navigating informal or ironic online text, and provide businesses with tools to moderate tone in customer-facing communication. While not the primary focus of this study, these applications highlight the broader accessibility and usability value of operational sarcasm detection.

## *6.5 Limitations*

Although this study demonstrates the feasibility of real-time sarcasm detection, several limitations constrain the generalisability of the findings.

Training data and contextual signals. The models were trained on large but text-only datasets, with sarcasm often labelled at the single-sentence level. As a result, they lacked access to conversational continuity or pragmatic context. This restricted their ability to capture sarcasm that emerges only in relation to surrounding discourse or world knowledge.

**Feature constraints:** While the system architecture supports mechanisms for user-feedback capture and automatic retraining, these features were disabled in the evaluated build. Consequently, the extension functions as stateless inference and cannot adapt its behaviour based on user corrections.

**Context and pragmatics:** Sarcasm expressed in very short utterances or through indirect rhetorical devices (e.g., quotations, irony, or scare-quotes) remained difficult to classify reliably. This is consistent with the observed misclassifications on Reddit, where the absence of broader conversational context increased the risk of false positives.

**Domain variation:** Performance was not uniform across sources. Predictions were most stable on news articles, moderate on explanatory webpages, and least reliable on Reddit. Because a single global threshold was applied, the system was unable to adjust to stylistic differences across domains.

**Evaluation scope:** The extension was tested on a limited number of webpages. While this provided indicative evidence of feasibility and highlighted domain-specific behaviour, a larger and more diverse evaluation would be required to fully assess robustness.

## *6.6 Future Work*

Future research will extend this work in several complementary directions.

**Richer training with context**: Incorporating datasets that include surrounding sentences or conversational turns allow models to better capture pragmatic contrast and polarity shifts. This is will to reduce false positives in short texts and improve performance on context-dependent sarcasm.

**Adaptive learning:** The feedback and auto-training mechanisms already present in the API can be enabled in future iterations. Allowing users to provide corrective labels in an opt-in and privacy-preserving manner would permit the system to gradually refine its decision boundaries and improve reliability over time.

**Domain adaptation:** Introducing domain-sensitive thresholds or lightweight adaptation strategies would help to stabilise predictions across heterogeneous text sources. For example, stricter decision policies could be applied on short, informal texts, while more permissive thresholds could be maintained for structured prose.

**Calibration:** Post-hoc calibration methods such as temperature scaling should be incorporated into the deployment pipeline to ensure that probability scores more accurately reflect empirical correctness. This will support a reliable thresholding and reduce misclassifications.

**Expanded evaluation:** Larger-scale testing, including a broader range of websites and user studies on perceived accuracy and trust, will provide stronger evidence of the extension's practical value. Future evaluations could also examine related pragmatic phenomena, such as irony or humour, to explore whether the same pipeline can be extended to adjacent tasks.

# Chapter 7: Conclusion

This dissertation investigates whether current NLP models can reliably detect sarcasm across diverse online domains and whether such models can be adapted for real-time deployment. Sarcasm remains a challenge for computational systems, as its meaning diverges from surface form and often depends on polarity inversion, pragmatic contrast, or contextual knowledge. These complexities cause systematic errors in NLP tasks such as sentiment analysis, stance detection, and dialogue systems. Rather than focusing solely on developing a production-ready tool, the study adopted a dual emphasis: conducting a systematic comparative evaluation of sarcasm detection models under controlled conditions, and validating these findings in a realistic deployment setting through the SarcQuest browser extension.

The results demonstrated a clear hierarchy of approaches. Classical baselines such as Logistic Regression provided interpretable but limited benchmarks, while pre-transformer deep learning architectures achieved only modest improvements. Transformer-based architectures delivered a decisive step forward, with BERT, RoBERTa, and DistilRoBERTa outperforming all earlier models. RoBERTa achieved the highest predictive scores, but DistilRoBERTa matched its performance within one F1 point while offering substantial gains in efficiency and memory footprint. This balance established DistilRoBERTa as the most suitable model for deployment in real-time applications.

Domain-level analysis highlighted the importance of text source and style. Performance was strongest on structured news headlines, moderate on Reddit, and weakest on Twitter, where brevity, noise, and reliance on external context continue to pose significant challenges. Calibration analysis revealed that transformer models, while highly accurate, were prone to overconfidence, necessitating post-hoc techniques such as temperature scaling to produce trustworthy probability estimates for threshold-based applications. Efficiency testing further confirmed that DistilRoBERTa achieves the most favourable trade-off between accuracy, latency, and model size, enabling its integration into SarcQuest as a proof-of-concept for real-time sarcasm detection.

The contributions of this dissertation are fourfold: the integration of a large, balanced multi-domain dataset; unified benchmarking of models spanning baselines, deep learning, and transformers; evaluation of calibration, latency, and size alongside predictive accuracy; and operational deployment through SarcQuest, demonstrating both feasibility and limitations in live environments. At the same time, several constraints remain. Models were trained on single-sentence annotations without conversational context, cross-domain fragility persisted and calibration techniques were limited to simple scaling. Deployment evaluation was small and did not yet incorporate feedback analysis.

In conclusion, this study shows that transformer architectures, and particularly distilled variants such as DistilRoBERTa, are essential for achieving robust sarcasm detection suitable for real-world use. While current models can deliver reliable performance in structured and semi-structured domains, challenges remain in handling negation, implicit context, and noisy microtexts. By systematically benchmarking models, addressing deployment-oriented factors, and operationalising detection in SarcQuest, this dissertation advances both theoretical understanding and practical feasibility. The central lesson is that success in sarcasm detection lies not only in maximising accuracy but also in ensuring reliability, efficiency, and robustness under the conditions of everyday web interaction.

# References


1. Ardilla, Z. N., Sari, T. I., Hayatin, N., & Fatichah, C. (2024). Sarcasm detection on news headline using multilayer bidirectional-LSTM with GloVe embeddings. AIP Conference Proceedings, 2927(1), 060035. https://doi.org/10.1063/5.0192254
2. Attardo, S. (2000). Irony as relevant inappropriateness. *Journal of Pragmatics, 32*(6), 793–826. https://doi.org/10.1016/S0378-2166(99)00070-3
3. Bamman, D., & Smith, N. A. (2015). Contextualized sarcasm detection on Twitter. In *Proceedings of the Ninth International AAAI Conference on Web and Social Media (ICWSM)* (pp. 574–577). AAAI. https://ojs.aaai.org/index.php/ICWSM/article/view/14655
4. Cai, H., Cai, J., & Wan, X. (2019). Multi-modal sarcasm detection in Twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2506–2515). Association for Computational Linguistics. https://aclanthology.org/P19-1239
5. Camp, E. (2012). Sarcasm, pretense, and the semantics/pragmatics distinction. *Noûs, 46*(4), 587–634. https://www.jstor.org/stable/41682690
6. Cheang, H. S., & Pell, M. D. (2011). Recognizing sarcasm without language: A cross-linguistic study of English and Cantonese. *Pragmatics & Cognition, 19*(2), 203–223. https://doi.org/10.1075/pc.19.2.02che
7. Danofer, D. (2017). *Sarcasm detection on Reddit* [Dataset]. Kaggle. https://www.kaggle.com/datasets/danofer/sarcasm
8. Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., & Ravi, S. (2020). GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4040–4054). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.372
9. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.48550/arXiv.1810.04805
10. Dong, H., Xu, H., & Xu, K. (2020). Transformer with context for sarcasm detection in social media. *arXiv Preprint*, arXiv:2005.11424. https://doi.org/10.48550/arXiv.2005.11424
11. Farha, I. A., Magdy, W., & Mubarak, H. (2020). iSarcasmEval: A shared task on intended sarcasm detection in English and Arabic. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)* (pp. 76–86). Association for Computational Linguistics. https://aclanthology.org/2020.semeval-1.111/
12. Gibbs, R. W. (1986). On the psycholinguistics of sarcasm. *Journal of Experimental Psychology: General, 115*(1), 3–15. https://doi.org/10.1037/0096-3445.115.1.3
13. Ghosh, A., & Veale, T. (2016). Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)* (pp. 161–169). Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-0425

14. Ghosh, A., Fabbri, A. R., & Muresan, S. (2017). The role of conversation context for sarcasm detection in online interactions. *arXiv Preprint*, arXiv:1707.06226. https://arxiv.org/abs/1707.06226
15. González-Ibáñez, R., Muresan, S., & Wacholder, N. (2011). Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Short Papers)* (pp. 581–586). Association for Computational Linguistics. https://aclanthology.org/P11-2102/
16. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)* (pp. 1321–1330). PMLR. https://proceedings.mlr.press/v70/guo17a.html
17. Happé, F. G. E. (1995). Understanding minds and metaphors: Insights from the study of figurative language in autism. *Metaphor and Symbolic Activity, 10*(4), 275–295. https://doi.org/10.1207/s15327868ms1004_3
18. Joshi, A., Sharma, V., & Bhattacharyya, P. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys, 50*(5), Article 73. https://doi.org/10.1145/3124420
19. Khodak, M., Saunshi, N., & Vodrahalli, K. (2018). A large self-annotated corpus for sarcasm. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)* (pp. 641–647). European Language Resources Association. https://aclanthology.org/L18-1102
20. Kreuz, R. J., & Caucci, G. M. (2007). Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on Computational Approaches to Figurative Language* (pp. 1–9). Association for Computational Linguistics. https://aclanthology.org/W07-0101/
21. Kreuz, R. J., & Roberts, R. M. (1995). Two cues for verbal irony: Hyperbole and the ironic tone of voice. *Metaphor and Symbolic Activity, 10*(1), 21–31. https://doi.org/10.1207/s15327868ms1001_3
22. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv Preprint*, arXiv:1907.11692. https://doi.org/10.48550/arXiv.1907.11692
23. Misra, R., & Arora, A. (2019). *News headlines dataset for sarcasm detection* [Dataset]. Kaggle. https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection
24. Naz, F., Kamran, M., Mehmood, W., Khan, W., Alkatheiri, M. S., Alghamdi, A. S., & Alshdadi, A. A. (2019). Automatic identification of sarcasm in tweets and customer reviews. *Journal of Intelligent & Fuzzy Systems, 37*(5), 6815–6828. https://doi.org/10.3233/JIFS-190596
25. Nikesh66. (2021). *Sarcasm tweets dataset* [Dataset]. Hugging Face. https://huggingface.co/datasets/nikesh66/Sarcasm-dataset
26. Oprea, S., & Magdy, W. (2020). iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1279–1289). Association for Computational Linguistics. https://aclanthology.org/2020.acl-main.118/
27. Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. In *COLING 2016: Technical Papers* (pp. 1601–1612). Association for Computational Linguistics. https://aclanthology.org/C16-1151/
28. Rajadesingan, A., Zafarani, R., & Liu, H. (2015). Sarcasm detection on Twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web*

*Search and Data Mining (WSDM '15)* (pp. 97–106). ACM. https://doi.org/10.1145/2684822.2685316

29. Reyes, A., Rosso, P., & Veale, T. (2013). A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation, 47*(1), 239–268. https://doi.org/10.1007/s10579-012-9199-1
30. Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of EMNLP 2013* (pp. 704–714). Association for Computational Linguistics. https://aclanthology.org/D13-1066
31. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT: A distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv Preprint*, arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108
32. Savini, E., & Caragea, C. (2022). Intermediate-task transfer learning with BERT for sarcasm detection. *Mathematics, 10*(5), 844. https://doi.org/10.3390/math10050844
33. Sinha, S., & Choudhary, M. (2023). Sarcasm detection using deep learning approaches: A review. *International Journal of Recent Technology and Engineering, 11*(6), 50–58. https://doi.org/10.35940/ijrte.F7476.0311623
34. Shu, A. (2024). BERT and RoBERTa for sarcasm detection: Optimizing performance through advanced fine-tuning. *ResearchGate*. https://www.researchgate.net/publication/386137701
35. Sulis, E., et al. (2016). Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not. *Knowledge-Based Systems, 108*, 132–143. https://doi.org/10.1016/j.knosys.2016.05.035
36. Tay, Y., Tuan, L. A., & Hui, S. C. (2018). Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 1010–1020). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1093
37. Wallace, B. C., Choe, D. K., Charniak, E., & Jensen, D. (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 (Short Papers)* (pp. 512–516). Association for Computational Linguistics. https://aclanthology.org/P14-2084
38. Wahyuni, E. D. (2025). Deep learning multimodal sarcasm detection in social media comments: The role of memes and emojis. *Journal of Artificial Intelligence and Technology, 16*(1), 45–55. https://ojs.istp-press.com/jait/article/view/699

# Appendices

## Tables of Content

### Tables



### Figures



### Supplementary Documentation

# Tables

*Table 5.2. Pairwise Model Comparisons with Statistical Significance Tests*

Pairwise Model Comparioson

| Model A | Model B | F1A | F1B | ΔF1 | Bootstrap *p* | McNemar *p* |
|---|---|---|---|---|---|---|
| RoBERTa (base) | DistilRoBERTa | **0.7786** | 0.7698 | 0.0088 | <0.001 | 1.63E-61 |
| RoBERTa (base) | BERT (base) | **0.7786** | 0.7679 | 0.0107 | <0.001 | 1.52E-135 |
| DistilRoBERTa | BERT (base) | **0.7698** | 0.7679 | 0.0019 | 0.0001. | 4.60E-24 |
| BiLSTM (300d) | CNN (100d) | **0.7072** | 0.6893 | 0.0179 | <0.001 | 6.95E-244 |
| BiLSTM+Attention (300d) | CNN (100d) | **0.7072** | 0.6893 | 0.0179 | <0.001 | 6.94E-227 |
| BiLSTM (300d) | CNN (300d) | **0.7072** | 0.6939 | 0.0133 | <0.001 | 3.39E-135 |
| BiLSTM+Attention (300d) | CNN (300d) | **0.7072** | 0.6939 | 0.0133 | <0.001 | 1.99E-123 |
| BiLSTM (100d) | CNN (100d) | **0.7013** | 0.6893 | 0.012 | <0.001 | 4.94E-159 |
| BiLSTM+Attention (100d) | CNN (100d) | **0.6994** | 0.6893 | 0.0101 | <0.001 | 5.34E-150 |
| BiLSTM (300d) | BiLSTM+Attention (100d) | **0.7072** | 0.6994 | 0.0078 | <0.001 | 5.82E-18 |
| CNN (100d) | CNN (300d) | 0.6893 | **0.6939** | -0.0046 | <0.001 | 4.39E-26 |
| BiLSTM (100d) | BiLSTM+Attention (300d) | 0.7013 | **0.7072** | -0.0059 | <0.001 | 1.75E-13 |
| BiLSTM (100d) | BiLSTM (300d) | 0.7013 | **0.7072** | -0.0059 | <0.001 | 3.47E-18 |
| BiLSTM+Attention (100d) | BiLSTM+Attention (300d) | 0.6994 | **0.7072** | -0.0078 | <0.001 | 1.10E-14 |

*This table summarises aggregate pairwise model comparisons across deep learning and transformer baselines. For each pair, the mean F1 scores, ΔF1 differences, bootstrap p-values, and McNemar significance tests are reported. Shading highlights higher values (red for F1A, blue for F1B), allowing differences in model performance to be visually compared.*

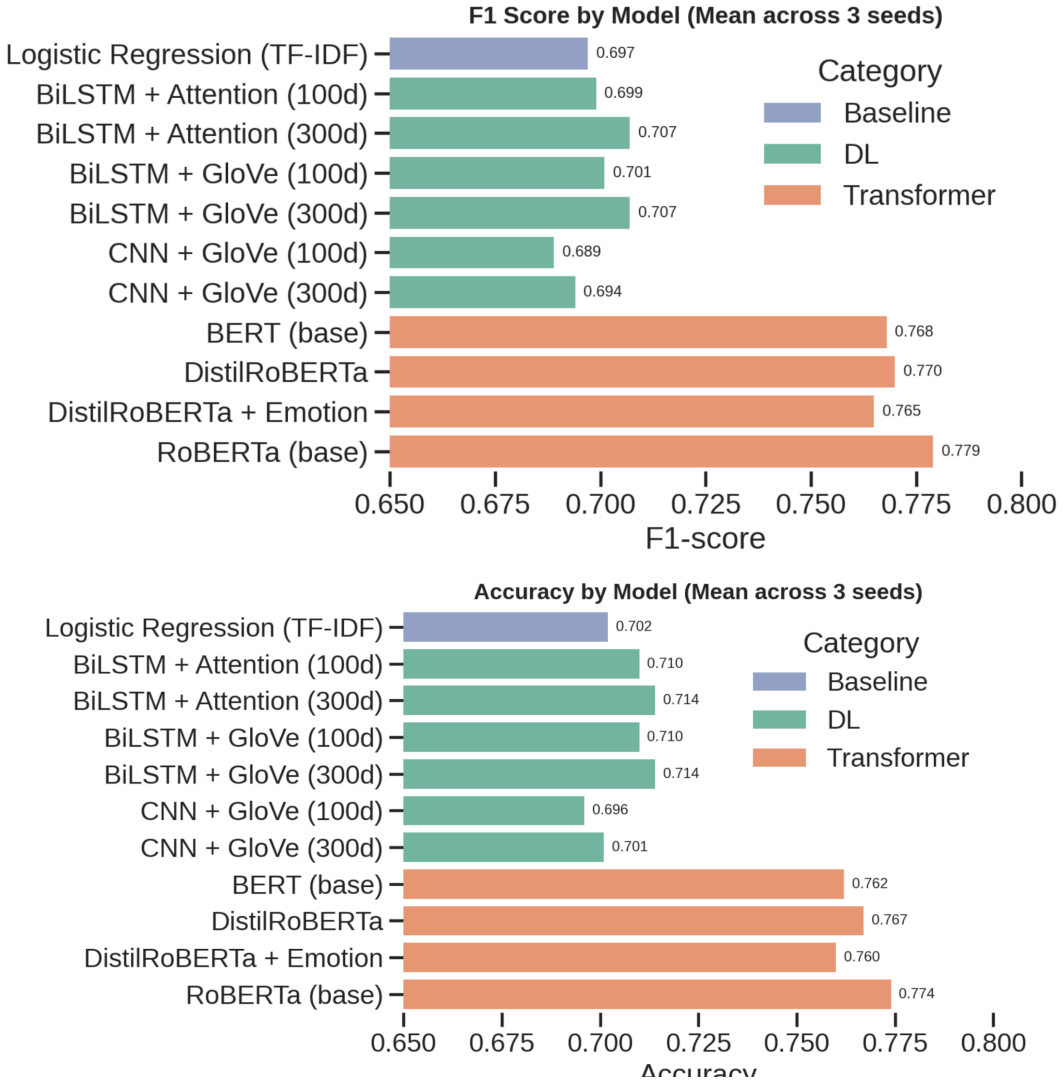
*Table 5.5: Domain-Specific Model Performance with Variance Across Seeds.*

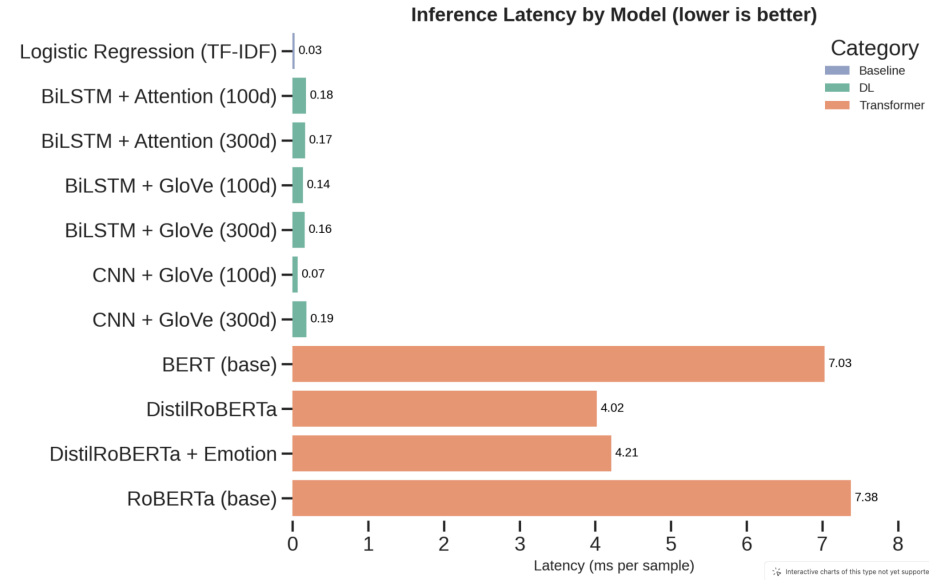
| Model | Domain | F1 (±std) | Acc (±std) | Prec | Rec | AUPRC | Brier | ECE | Variance note |
|---|---|---|---|---|---|---|---|---|---|
| LogReg_TFIDF | News | 0.768 ±0.013 | 0.760 ±0.013 | 0.74 | 0.795 | 0.586 | 0.18 | 0.23 | Low, stable |
| LogReg_TFIDF | Reddit | 0.612 ±0.018 | 0.604 ±0.020 | 0.6 | 0.631 | 0.455 | 0.23 | 0.21 | Low variance |
| LogReg_TFIDF | Twitter | 0.431 ±0.026 | 0.701 ±0.021 | 0.61 | 0.315 | 0.32 | 0.27 | 0.3 | Moderate variance |
| BiLSTM_Attn_300 | News | 0.760 ±0.035 | 0.761 ±0.037 | 0.77 | 0.757 | 0.828 | 0.17 | 0.32 | Std low |
| BiLSTM_Attn_300 | Reddit | 0.553 ±0.096 | 0.588 ±0.040 | 0.6 | 0.535 | 0.649 | 0.24 | 0.11 | Moderate variance |
| BiLSTM_Attn_300 | Twitter | 0.535 ±0.082 | 0.549 ±0.047 | 0.57 | 0.544 | 0.594 | 0.25 | 0.1 | High variance |
| CNN_GloVe100 | News | 0.646 ±0.102 | 0.652 ±0.095 | 0.7 | 0.662 | 0.71 | 0.21 | 0.21 | Std moderate |
| CNN_GloVe100 | Reddit | 0.548 ±0.122 | 0.530 ±0.059 | 0.53 | 0.606 | 0.574 | 0.25 | 0.06 | High variance |
| CNN_GloVe100 | Twitter | 0.620 ±0.044 | 0.536 ±0.053 | 0.53 | 0.769 | 0.59 | 0.25 | 0.08 | High variance |
| RoBERTa | News | 0.895 ±0.002 | 0.909 ±0.002 | 0.88 | 0.901 | 0.882 | 0.07 | 0.5 | Very low variance |
| RoBERTa | Reddit | 0.777 ±0.001 | 0.770 ±0.001 | 0.75 | 0.785 | 0.83 | 0.16 | 0.33 | Very low variance |
| RoBERTa | Twitter | 0.468 ±0.015 | 0.741 ±0.018 | 0.68 | 0.388 | 0.411 | 0.18 | 0.62 | Moderate variance |
| DistilRoBERTa | News | 0.882 ±0.002 | 0.897 ±0.002 | 0.87 | 0.883 | 0.871 | 0.08 | 0.49 | Very low variance |
| DistilRoBERTa | Reddit | 0.768 ±0.000 | 0.763 ±0.000 | 0.74 | 0.762 | 0.811 | 0.17 | 0.31 | Very low variance |
| DistilRoBERTa | Twitter | 0.457 ±0.017 | 0.737 ±0.019 | 0.68 | 0.377 | 0.402 | 0.18 | 0.6 | Moderate variance |
| BERT | News | 0.861 ±0.004 | 0.877 ±0.003 | 0.85 | 0.87 | 0.855 | 0.1 | 0.5 | Low variance |
| BERT | Reddit | 0.766 ±0.001 | 0.758 ±0.001 | 0.73 | 0.761 | 0.801 | 0.17 | 0.34 | Very low variance |
| BERT | Twitter | 0.432 ±0.005 | 0.722 ±0.006 | 0.67 | 0.345 | 0.389 | 0.2 | 0.64 | Low variance |

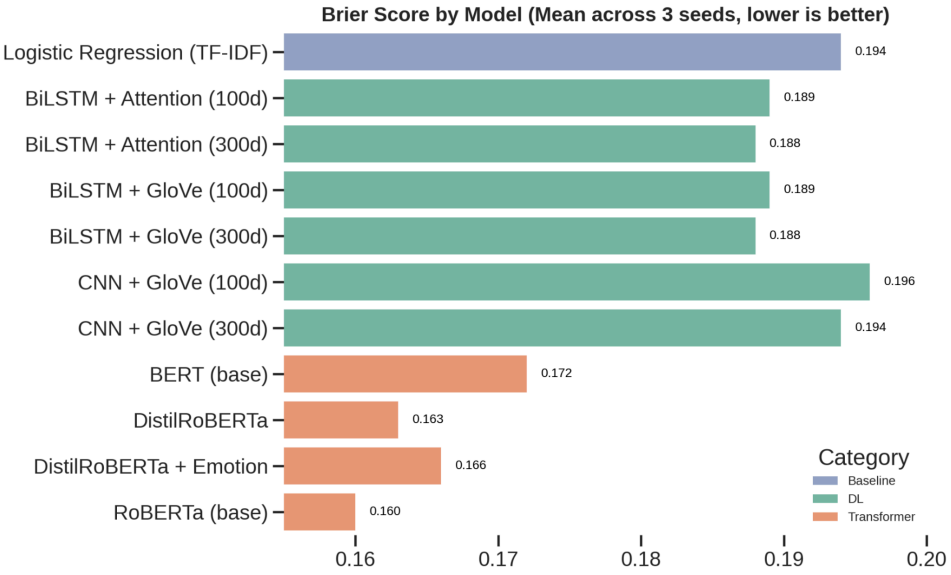
*This table reports domain-specific results for classical, deep learning, and transformer models trained and evaluated separately on News, Reddit, and Twitter. Metrics include F1-score, accuracy, precision, recall, AUPRC, Brier score, and Expected Calibration Error (ECE), with mean ± standard deviation across three seeds. Variance notes indicate model stability (e.g., low vs. high variance). Warm (red) shading indicates poorer performance; cool (blue) shading indicates stronger performance. Best results per column are highlighted.*

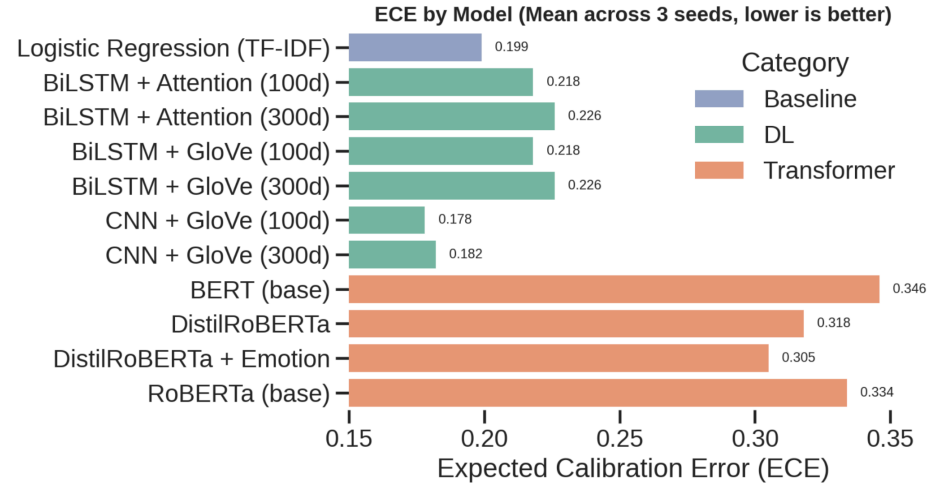
# Figures

*Figure 5.1. 5 Graphs of Model Performance Across Baselines, Deep Learning, and Transformer Architectures*





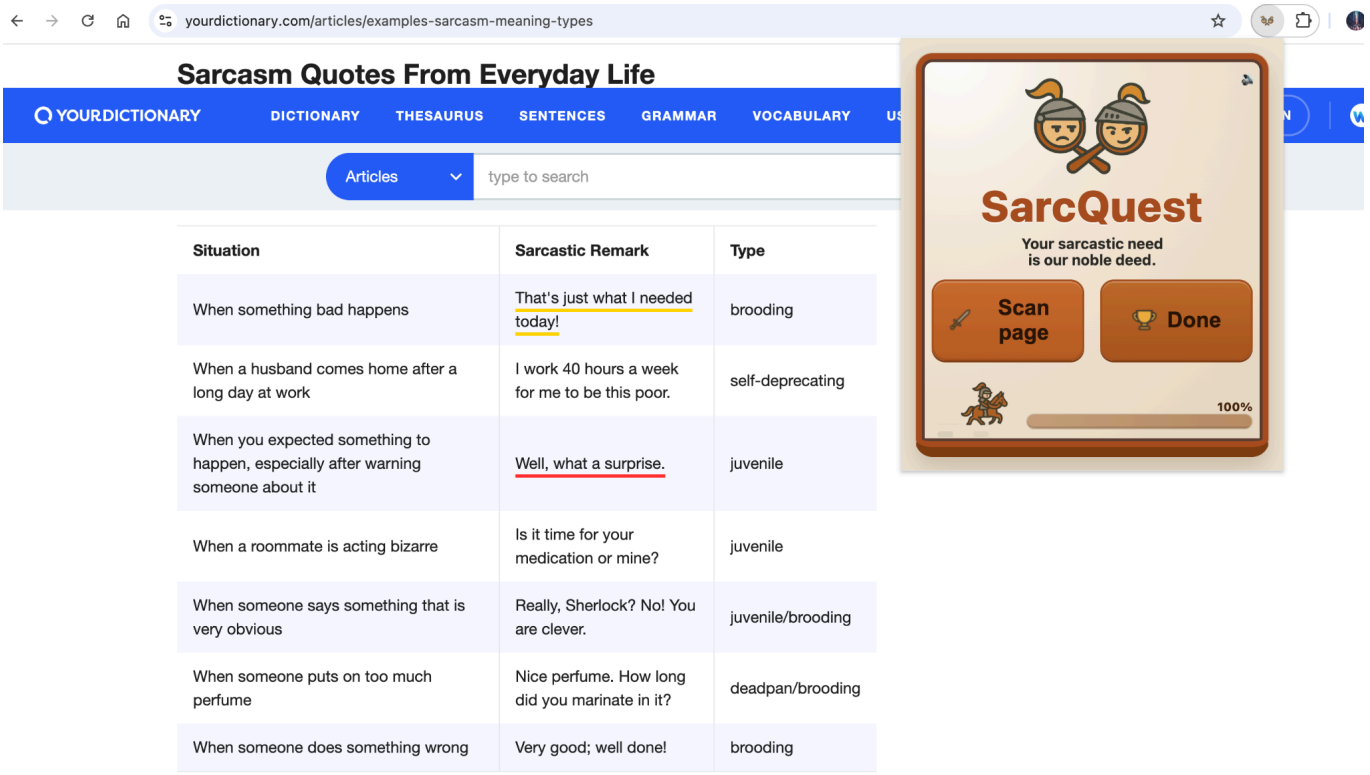

*Mean results across three random seeds are shown for F1-score, accuracy, Brier score, Expected Calibration Error (ECE), and inference latency. Transformers consistently outperform classical and deep learning models on F1 and accuracy, while exhibiting higher latency. This is just for visual representation. The full per-model values are reported in Table 5.1:*

*Figure 5.2. Examples of Model Misclassifications in Sarcasm Detection*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Lol eye strain is a myth people | 1 | 0.2546195827085462 | 0 | False | False | False | False |
| i suppose you need a decent base fund for that right? | 0 | 0.5376577353470633 | 1 | False | False | False | False |
| We here in Central Maine hate you too! | 1 | 0.49061527967997925 | 0 | False | True | False | False |
| In the officers defense they do not do much firearms training | 1 | 0.4430940512918399 | 0 | True | False | False | False |
| A true sociopath | 1 | 0.30309889765178794 | 0 | False | False | False | False |
| Ya and with recount and skada meters Ingame | 1 | 0.249127605303211 | 0 | False | False | False | False |
| Ill probably just keep my McKinnon as a backup but I Am really looking into getting him | 0 | 0.5106170837507277 | 1 | False | False | False | False |
| ZUCK DELETE THIS PLZ CTR | 1 | 0.30460798794203914 | 0 | False | False | False | False |
| Then why are you letting the government take away your rights now? | 0 | 0.6759497965901369 | 1 | False | False | False | False |
| Let Us not all be so quick to judge | 1 | 0.45785268259641515 | 0 | True | False | False | False |
| movie deemed acceptable for mom and dad | 1 | 0.3453589316281007 | 0 | False | False | False | False |
| i only know of the passphrase and that is 57 perfect score | 0 | 0.6434819513012545 | 1 | False | False | False | False |
| Get it together game designers! | 0 | 0.5949523296177138 | 1 | False | True | False | False |
| Some people appreciate quality | 0 | 0.5655810164992868 | 1 | False | False | False | False |
| They might find out your secret address that no one knows | 1 | 0.3680876830034822 | 0 | True | False | True | False |
| Now you are just making stuff up | 1 | 0.3916984479280798 | 0 | False | False | False | False |
| NaVi confirmed tier 4 team | 0 | 0.526946003401348 | 1 | False | False | False | False |
| Jackie Chan? | 1 | 0.2899001589578795 | 0 | False | False | False | False |
| Is anime so bereft that it needs to steal from light novels and manga? | 1 | 0.46671047384020137 | 0 | False | False | False | False |
| great work man these posts are always great to see! | 0 | 0.7626271552686027 | 1 | False | True | True | False |
| I do not think smoking crack will improve your situation | 1 | 0.39486306500858714 | 0 | True | False | False | False |

*This table presents illustrative cases where the model predictions diverged from the gold labels. Each row shows the input text, ground-truth label, predicted probability, predicted class, and associated error flags. Misclassifications often occur in sentences with ambiguous tone, subtle rhetorical devices, or reliance on external context, highlighting the challenge of distinguishing sarcasm from literal statements*

*Figure 6.1: Example of the SarcQuest Browser Extension Highlighting Sarcastic Sentences on YourDictionary*

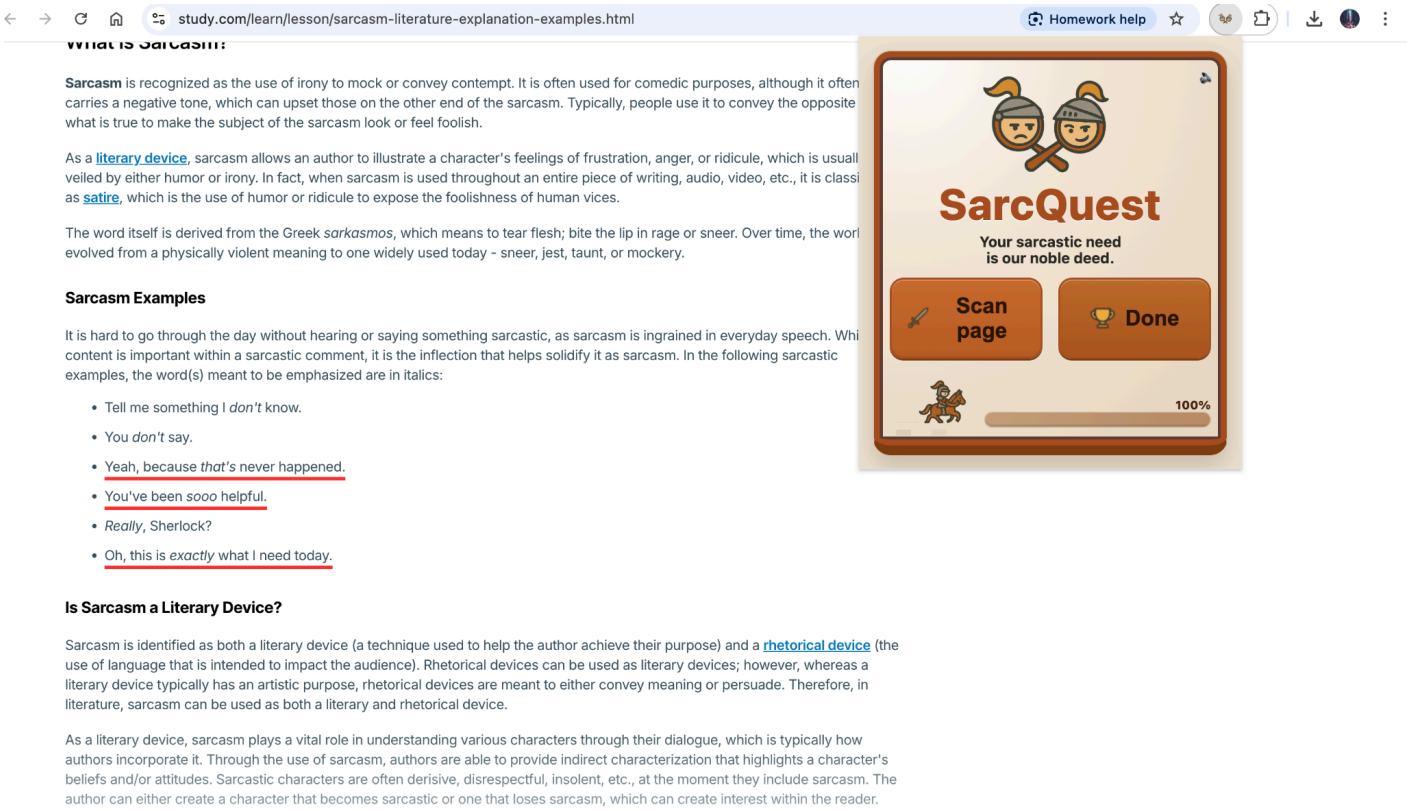



| Situation | Sarcastic Remark | Type |
|---|---|---|
| When something bad happens | That's just what I needed today! | brooding |
| When a husband comes home after a long day at work | I work 40 hours a week for me to be this poor. | self-deprecating |
| When you expected something to happen, especially after warning someone about it | Well, what a surprise. | juvenile |
| When a roommate is acting bizarre | Is it time for your medication or mine? | juvenile |
| When someone says something that is very obvious | Really, Sherlock? No! You are clever. | juvenile/brooding |
| When someone puts on too much perfume | Nice perfume. How long did you marinate in it? | deadpan/brooding |
| When someone does something wrong | Very good; well done! | brooding |

*This screenshot illustrates the SarcQuest Chrome extension in use on a webpage containing sarcastic examples. Sarcastic text ('Well, what a surprise.') is underlined in red by the extension, while the control panel allows the user to scan the page, review highlighted results, or complete the task. The status bar at the bottom reflects scanning progress.*

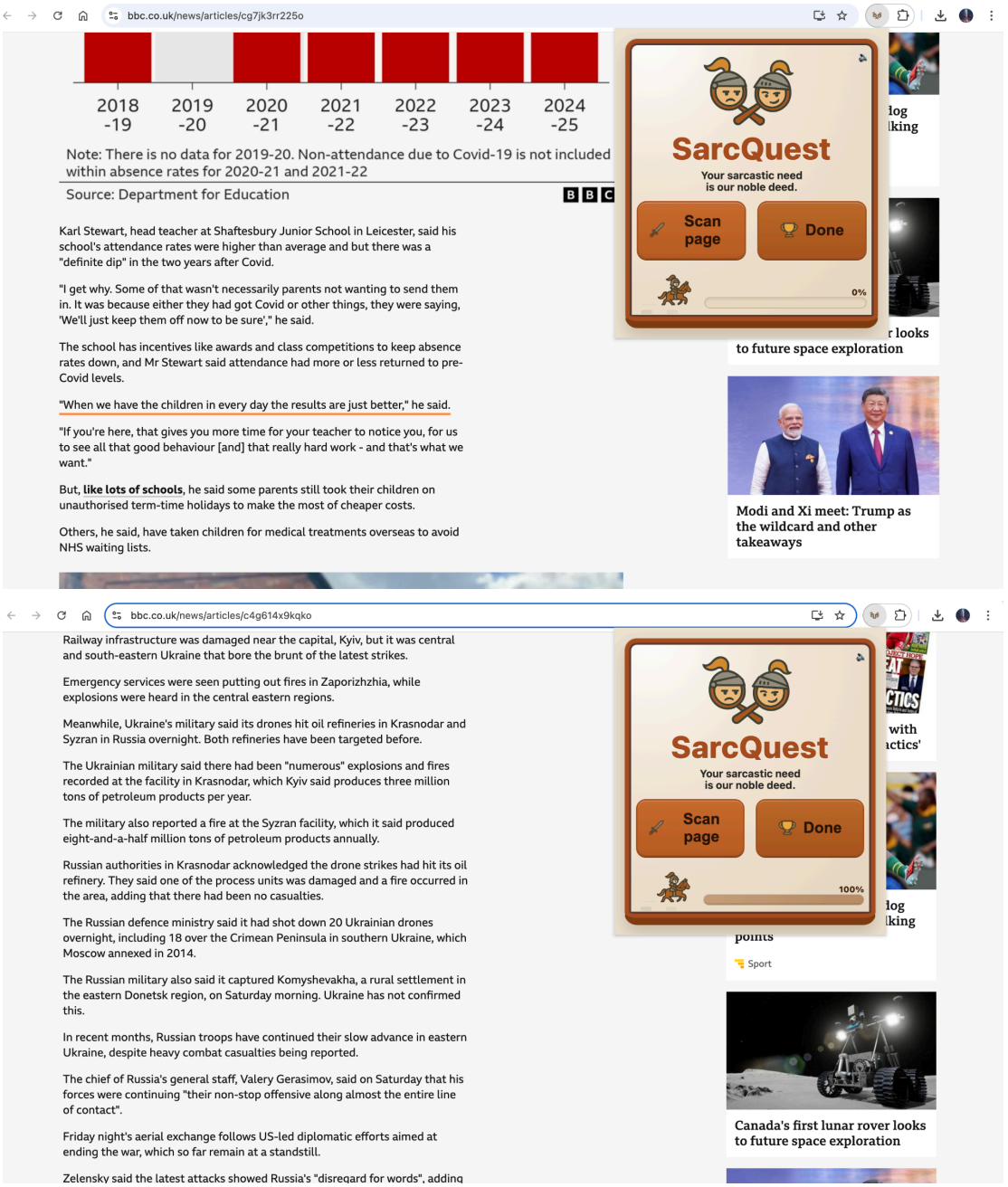
*Source: https://www.yourdictionary.com/articles/examples-sarcasm-meaning-types*

*Figure 6.2: Example of the SarcQuest Browser Extension Highlighting Sarcastic Sentences on Study.com*




*This screenshot shows the SarcQuest Chrome extension applied to a webpage containing sarcasm examples. Phrases such as 'Yeah, because that's never happened' and 'You've been sooo helpful' are underlined in red by the extension to indicate sarcasm. The extension control panel on the right provides options to scan the page, complete the task, and displays scanning progress. Source: https://study.com/learn/lesson/sarcasm-literature-explanation-examples.html*

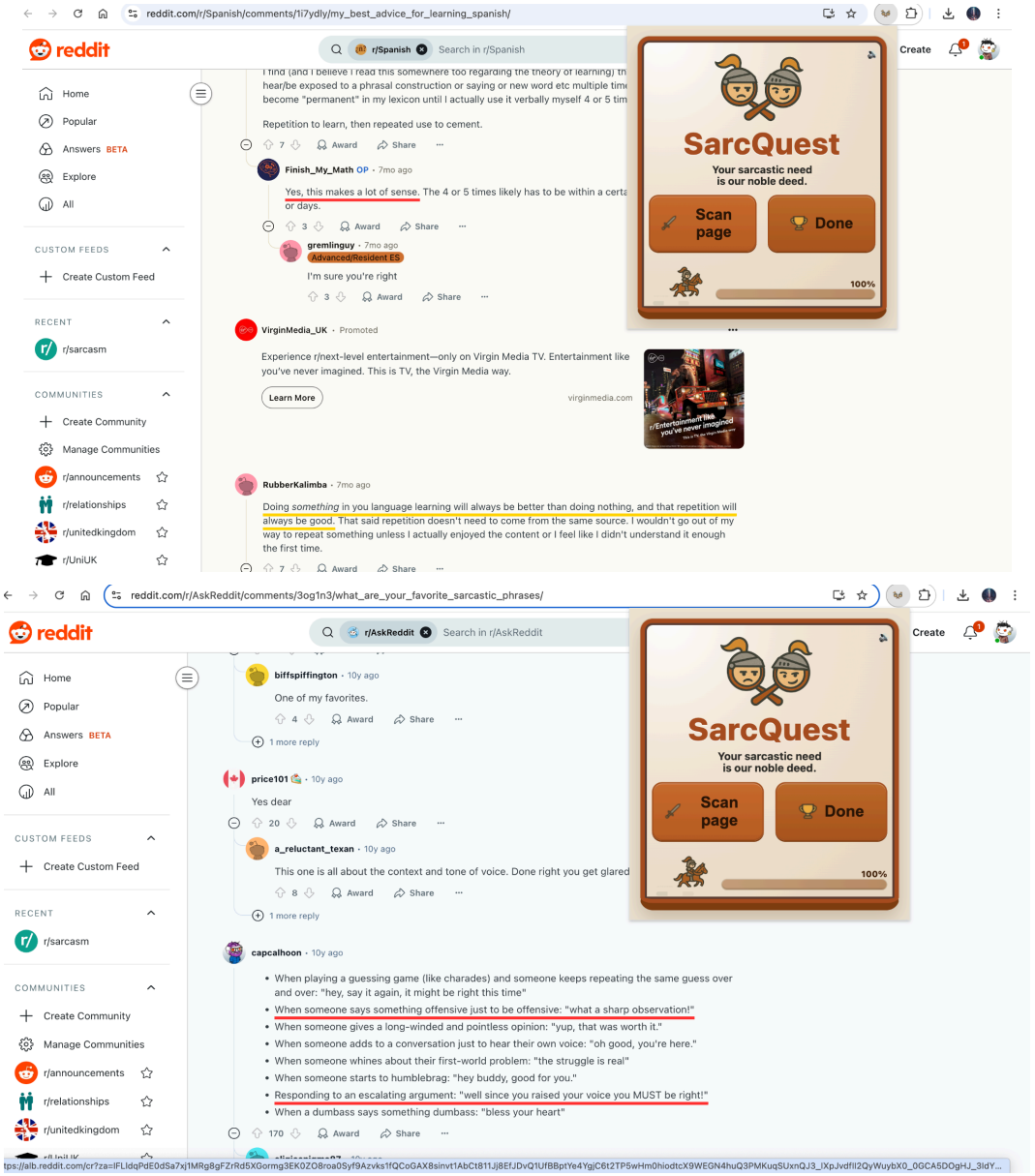
*Figure 6.3: Two Examples of the SarcQuest Browser Extension Highlighting Sarcastic Sentences on BBC*




*These screenshots demonstrate the SarcQuest Chrome extension in use on BBC News webpages. In the left-hand panel, the extension underlines a sarcastic statement ("When we have the children in every day the results are just better") while the control panel indicates scanning progress at 0%. In the right-hand panel, the extension successfully completes a scan (100%), with results displayed on a separate BBC article. These examples illustrate the extension's functionality when applied to real-world journalistic text, capturing sarcastic remarks in varied reporting contexts.*

*Source: https://www.bbc.co.uk/news/articles/cg7jk3rr2250, https://www.bbc.co.uk/news/articles/c4g814x9kkqo*

*Figure 6.4: Examples of the SarcQuest Browser Extension Highlighting Sarcastic Sentences on Reddit*




*These screenshots show the SarcQuest Chrome extension applied to Reddit discussion threads. On the left, the extension underlines sarcastic remarks within a language-learning thread (e.g., "Yes, this makes a lot of sense"). On the right, multiple sarcastic comments are detected in a thread explicitly discussing sarcasm, such as "What a sharp observation!" and "When a dumbass says something dumbass: 'bless your heart.'" The extension's control panel indicates successful scanning (100%). These examples highlight the system's ability to capture sarcasm in informal, user-generated text.*

*Source: https://www.reddit.com/r/Spanish/comments/1i7ydly/my_best_advice_for_learning_spanish/; https://www.reddit.com/r/AskReddit/comments/3og1n3/what_are_your_favorite_sarcastic_phrases/*

# Running the Code and Extension

## Repository Structure and Execution

*To access my GitLab repository please go to [https://git.cs.bham.ac.uk/projects-2024-25/emi436/-/tree/main](https://git.cs.bham.ac.uk/projects-2024-25/emi436/-/tree/main)*

*The repository is organised into multiple subdirectories relevant for code execution, data management, and deployment:*

- *Jupyter Notebook (Code)*
  - *This folder contains the main implementation notebooks.*
  - *It can be recognised by the Last Comit being "Replace NLP_Sarcasum Detector.ipynb"*
  - *NLP_Sarcasum Detector.ipynb provides the complete code for preprocessing, model training, and evaluation across baselines and deep learning architectures.*
  - *Untitled1.ipynb contains additional experimentation that were conducted alongside the main pipeline.*
  - *These notebooks reproduce all results with the exception of transformer models trained on the full dataset (executed separately in Google Colab due to GPU requirements).*
- *Jupyter Notebook (Data)*
  - *A separate folder of the same name, marked by the commit "Delete Untitled1.ipynb" contains the datasets that had to be uploaded manually (excluded from version control by .gitignore).*
  - *The key file is df_filtered.csv, the final cleaned dataset derived from the raw sources.*
  - *This file underpins all experiments: it was used both locally in Jupyter for baselines and deep learning models, and remotely in Google Colab for transformer training.*
- *Google Colab*
  - *Contains the notebook used for large-scale training of transformer architectures (RoBERTa, DistilRoBERTa, ELECTRA, etc.) on the full cleaned dataset.*
  - *Execution here was necessary due to the computational demands of fine-tuning, which exceeded the resources of local machines.*
- *Extension*
  - *Holds the deployment code. This includes (i) the code FastAPI service uploaded to HuggingFace Spaces to provide inference as an API endpoint, and (ii) the Chrome extension source code (SarcQuest) for real-time testing.*

## To reproduce the experiments:

1. *Download or clone the repository.*
2. *Place all folders (from the Jupyter Notebook (Data) folder) in the same working directory as NLP_Sarcasum Detector.ipynb.*
3. *Run the notebook step-by-step in Jupyter Notebook or JupyterLab.*
4. *For transformer training, open the Google Colab notebook, upload df_filtered.csv and execute the cells in order with GPU runtime enabled. The training was done on a L100 GPU*

# Running the Extension

*The SarcQuest Chrome extension can be installed and tested in two different ways:*

1. *Manual installation (developer mode)*
    - *Open the Chrome browser and navigate to chrome://extensions.*
    - *Enable Developer Mode (toggle in the top right corner).*
    - *Click Load unpacked and select the folder SarcQuest-Mario from the repository's Extension directory.*
    - *The extension will appear in the toolbar and can be pinned for quick access.*

2. *Installation from Chrome Web Store*
    - *Alternatively Navigate to the Chrome Web Store.*
    - *Install the published SarcQuest extension directly.*

# Functionality of the Extension

*The extension provides a simple user interface designed to demonstrate real-time sarcasm detection in a web-browsing environment.*

- *Scan button – initiates extraction of all visible text on the active webpage, which is then batched and sent to the HuggingFace API for classification. Sarcastic sentences are underlined in real time.*
- *Confidence inspection – clicking on any highlighted sentence reveals the model's probability score, allowing users to see the level of certainty behind each prediction.*
- *Feedback controls – a thumbs-up/down interface for user feedback was implemented but is currently disabled in the deployed version. This feature is intended for future iterations where user input will be logged to improve model accuracy.*
- *Done button – removes all underlining and restores the page to its original state, ensuring a non-intrusive browsing experience.*
- *Sound notification – an audible alert signals completion of scanning. Users may toggle a mute option located in the extension's top-right corner.*

# Privacy Considerations

*The extension is designed with privacy in mind. Text is processed in memory within the active browser tab. Only extracted sentences are sent to the HuggingFace inference API for classification, and no personal identifiers or browsing history are stored, logged, or transmitted. The extension does not persist data beyond the immediate prediction request, aligning with principles of lightweight, privacy-preserving NLP deployment.*