Slang or Not? Exploring NLP Techniques for Slang Detection using the SlangTrack Dataset

Anonymous ACL submission

Abstract

The widespread use of casual language, including slang, poses significant difficulties for natural language processing systems, particularly in automatically recognising varied word uses. Although previous research has addressed slang through the creation of dictionaries, sentiment analysis, word formation, and interpretation, there has been limited focus on the fundamental issue of detecting slang. This paper focuses on the detection of slang within natural English sentences. To comprehensively tackle this problem, we constructed a novel dataset that includes words commonly used in both slang and non-slang contexts. The dataset comprises ten target words that display at least one slang sense as well as one non-slang sense; each sentence has been manually annotated as either slang or non-slang, achieving high inter-annotator agreement. Additionally, we sought to identify the most effective approach for addressing this issue. To achieve this, we compared and evaluated different approaches, including (1) traditional machine learning-based models (ML), (2) deep learning-based models (DL) with both contextual and static embeddings, (3) fine-tuning various language models (LMs), and (4) fine-tuning different large language models (LLMs). The results show that fine-tuning language models, particularly BERT-large-uncased, achieved the highest performance, with an F1-score of 69% for slang and 92% for non-slang, a macro-averaged F1-score of 80%, a weighted-averaged F1-score of 87%, and an overall accuracy of 87%.

Keywords: Slang detection, Text classification, Annotated corpus.

1 Introduction

Disclaimer: This work includes offensive slang examples, which do not reflect the researchers' views.

Slang is a form of informal language that includes words and phrases used within particular groups (Dumas and Lighter, 1978; Adams, 2012; Green, 2015). The adaptability of slang captivates language users and learners, while also presenting unique challenges and opportunities for natural language processing (NLP) systems (Eisenstein, 2013; Blodgett et al., 2016).

Computer scientists and linguists have developed methods to understand semantics on multiple levels for years. However, understanding slang remains a significant roadblock to deciphering the true meaning behind conversations. Slang is crucial for grasping the true meaning of a sentence because slang words and phrases often carry specific cultural and contextual connotations (Bucholtz, 2006; Green, 2015) that standard language does not. Slang can convey nuanced emotions, attitudes, and social affiliations that are essential for fully comprehending the speaker's intent (Eisenstein et al., 2014; Partridge, 2015).

Additionally, identifying slang is vital for tracking semantic changes and handling semantics-based tasks (Adams, 2012; Sun et al., 2022). Slang often evolves rapidly, and understanding these changes can help in updating language models and improving the accuracy of various NLP tasks. For example, the word "*cool*" has undergone significant semantic changes over time, and recognising its current slang usage is crucial for accurate semantic analysis (Coleman, 2012; Dhuliawala et al., 2016).

Detecting slang is particularly challenging due to the complex semantics associated with slang words, which can be interpreted in multiple ways. The nuances and context often lead to varied interpretations, making it difficult to accurately identify and understand whether a sentence contains slang. This complexity is further heightened by the presence of double entendres (Kiddon and Brun, 2011). For instance, the phrase "He's a player on Sundays" could refer to someone who plays sports on Sundays or imply that he manipulates others

romantically on Sundays.

Despite its importance and prevalence, slang detection has received limited attention in NLP. This paper addresses this gap by proposing a computational approach to studying slang detection in historical and social media data. Our wider goal is to shed light on the role of slang in these data types and provide a foundation for further research in this area. By doing so, we aim to better understand how slang impacts our interactions and develop more effective ways to detect and interpret it.

This paper introduces a binary classification system designed to identify sentences that either contain slang or not. Our primary research question is: Can an algorithm be trained to determine if a sentence like "*Maybe it is some personal thing that eats me*" contains slang usage, specifically at the sentence level? We explore whether it is possible to automatically detect slang within natural sentences and identify the techniques that distinguish slang from standard language usage. Our contributions include the following:

- A new corpus annotated with slang and non-slang labels, supporting the development of binary classification methods to automatically detect whether a given sentence contains at least one instance of slang.
- A classification system developed to determine whether a given sentence contains slang or not using (1) traditional machine learning models (ML), (2) deep learning models (DL), (3) fine-tuning of language models (LMs), and (4) large language models (LLMs).
- A comparative study and error analysis exploring the performance of various algorithms.

2 Related work

2.1 Construction of Slang Dictionaries and Sentiment Analysis

In the evolving landscape of computational linguistics, developing resources to handle informal language, especially slang, is crucial. This section explores significant contributions made by innovative tools such as SlangNet, SlangSD, and SLANGZY, which focus on developing and expanding slang dictionaries and enhancing sentiment analysis for slang-laden content.

SlangNet (Dhuliawala et al., 2016) provides a WordNet-like structure specifically designed for English slang, using Urban Dictionary as its primary data source. It augments traditional WordNet by integrating slang word senses, thus supporting NLP applications like web mining and information retrieval. By distinguishing between conventional and slang word senses, SlangNet enhances the interpretation of internet-based language in NLP tools.

SlangSD (Wu et al., 2018) systematically builds a sentiment dictionary focused on slang words commonly used on social media. It addresses the dynamic nature of slang by continuously updating its database with new slang terms and their associated sentiments, making it a robust resource for sentiment analysis. The dictionary leverages web resources and social media corpora to automatically estimate sentiment polarity, thereby improving sentiment classification in user-generated content.

SLANGZY (Gupta et al., 2019) employs machine learning algorithms to interpret slang within online communication. It adapts dynamically to new slang forms, improving the performance of language models in applications ranging from chatbots to social media analytics. By integrating these capabilities, SLANGZY enhances the accuracy and relevance of sentiment analysis tools when dealing with informal, slang-heavy texts.

2.2 Slang word creation and interpretation

In the rapidly evolving field of NLP, the detection, interpretation, and analysis of slang have garnered increasing interest, particularly due to the prevalence of slang in digital communications. Early studies, such as those by (Pal and Saha, 2015), primarily focused on building and expanding dictionary-based resources for slang detection and interpretation.

Advancements in deep learning have led to more sophisticated automatic slang identification techniques and frameworks that analyse cognitive usage patterns, significantly enhancing detection accuracy at both the sentence and token levels. For instance, the study by (Pei et al., 2019) employs bidirectional recurrent neural networks (BiLSTM) (Hochreiter and Schmidhuber, 1997), conditional random fields (CRFs) (Lafferty et al., 2001), and multilayer perceptrons (MLPs) (Rauber and Berns, 2011), achieving an F1 score of 0.80 for sentence-level detection and 0.50 for token-level identification.

(Ni and Wang, 2017) introduced a neural sequence-to-sequence model designed to generate explanations for non-standard English phrases au-

tomatically. While this approach contributes to explaining slang expressions, it does not address the crucial task of detecting or identifying slang, which remains a key challenge in the field.

Furthermore, the study by (Lynn et al., 2019) applied deep learning methods, specifically BiLSTM and Bi-GRU, to detect misogynistic slang within a dataset derived from Urban Dictionary. Their findings indicate that deep learning models significantly outperform traditional machine learning approaches, such as logistic regression, Naive Bayes, and Random Forest, highlighting the superior performance of deep learning for detecting specific slang usages within a given context.

Additionally, the incorporation of semantically informed methodologies, as proposed by (Sun et al., 2022), marks a significant step forward. By merging semantic insights with contextual data, their approach refines slang interpretation, addressing critical gaps in traditional processing techniques that often overlook the nuanced meanings of slang.

Separate research efforts have been directed towards understanding the mechanisms behind slang word creation, a growing area of importance due to the increasing use of informal expressions online. For example, (Kulkarni and Wang, 2018) developed generative models for slang that effectively capture patterns such as blends, clippings, and reduplicatives, achieving top-tier performance on human-annotated datasets. These models offer valuable insights into slang word formation, which is becoming increasingly relevant in the digital age.

3 Dataset

In previous research, the only dataset available for the binary classification of slang is the one presented by (Pei et al., 2019). However, this dataset is not publicly accessible and comes with notable limitations. It does not differentiate between words that have both slang and non-slang meanings, nor does it provide illustrative examples for each sense. Instead, it compiles examples from sources such as the Online Slang Dictionary[1] and Wall Street News (2011-2016) in the Penn Treebank (Marcus et al., 1993), where the keywords often differ between classes, making generalisation easier.

To address this gap, we have constructed a new dataset, SlangTrack (ST) Dataset[2], specifically designed to include words that are commonly used in both slang and non-slang contexts. Our focus is on identifying and selecting target words that exhibit both slang and non-slang senses, providing examples for each usage. This approach is novel in slang research, aiming to create a binary classification system that can effectively distinguish between the dual meanings of words, thus improving the detection and interpretation of slang within natural language.

3.1 Data Collection (Target Words and Examples)

In our study, we selected target words that coexisted in the slangSD[3] wordlist and COHA (Davies, 2012), based on the number of senses per word. Each target word used in our experiment has at least one slang sense and one dominant sense. We collected the meanings of these target words from Green's Dictionary of Slang[4], Urban Dictionary[5], and the Online Slang Dictionary[6], and cross-referenced them with the Oxford English Dictionary[7] to confirm their dominant non-slang sense. We further verified that each target word appeared in both the COHA dataset and Twitter[8], ensuring the collection of relevant senses for comprehensive analysis. Ultimately, we selected ten target words that met our criteria, excluding proper nouns.

We collected our examples for every target word from multiple sources. This dataset includes comprehensive contextual information for each target word. Among the sources used, we selected the clean version of the Corpus of Historical American English (CCOHA) because of its detailed documentation, as noted by (Alatrash et al., 2020). To complement this, we also utilised Twitter data, which offers a rich, current source of contemporary slang and informal language. Twitter's dynamic and real-time communication makes it a valuable tool for observing and analysing the rapid changes in modern language, particularly slang. We extracted all available examples for each target word from COHA, covering the years 1980–2010, and collected 1,000 examples per target word from Twitter, spanning

[1] https://http://onlineslangdictionary.com

[2] The dataset utilised in this study will be made available upon request. Interested researchers may contact the corresponding author for access.

[3] https://www.rdocumentation.org/packages/lexicon/versions/1.2.1

[4] https://greensdictofslang.com

[5] https://www.urbandictionary.com

[6] https://http://onlineslangdictionary.com

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

[8] https://twitter.com

Example Sentences	Target Keyword	Category
Today I heard, for the first time, a short scientific talk given by a man dressed as **a rodent**...! An interesting experience.	Rodent	Slang
On the other. Mr. Taylor took food requests and with a stern look in his eye told the children to stay seated until he and his wife returned with the food. The children nodded attentively. After the adults left, the children seemed to relax, talking more freely and playing with one another. When the parents returned, the kids straightened up again, received their food, and began to **eat**, displaying quiet and gracious manners all the while.	Eat	Non-Slang
Greater than this one that washed between the shores of Florida and Mexico. He balanced between the breakers and the turning tide. Small particles of sand churned in the waters around him, and a small fish swam against his leg, a momentary dark streak that vanished in the surf. He began to swim. Buoyant in the **salty** water, he swam a hundred meters to a jetty that sent small whirlpools around its barnacle rough pilings.	Salty	Non-Slang
Mom was totally hating on my dance moves. She's so **salty**.	Salty	Slang

Table 1: Sample texts from the dataset.

the period from 2010–2020.

3.2 Annotation Guidelines and Details

The selected target words, along with their corresponding examples and all unique senses, were provided to the annotators. The annotators were tasked with identifying and labelling each example based on its relevant slang or non-slang sense. For this experiment, different instances of both slang and non-slang classes were grouped together to form a binary classification setting. Table 1 presents examples of target words categorised into slang and non-slang classes.

Our annotation team consisted of three individuals, all with strong proficiency in English and at least a bachelor's degree. One of the team members, who held a degree in Linguistics, was appointed as the primary annotator. To ensure consistency in the annotation process, two annotators independently labelled a random sample of 1,000 sentences in a pilot task. This pilot phase was instrumental in refining the annotation guidelines. Any discrepancies between the annotators were resolved by the primary annotator, ensuring consensus on the final annotations.

The inter-annotator agreement was measured using Cohen's Kappa (Cohen, 1960), with the final value of 88.7% indicating a high agreement between annotators.

3.3 Data Statistics

The complete dataset comprises a vocabulary of 48,508 unique words (vocabulary); those words make a total of 310,170 (tokens) across all the instances, with an average post length of 34.6 words per post and an average sentence length of 3.74 sentences per post. The dataset has been divided into three subsets: training, validation, and testing using stratified sampling, with the proportions set at 70%, 15%, and 15%, respectively.

Keyword	Non-slang	Slang	Total
Bmw	1083	14	1097
Brownie	582	382	964
Chronic	1415	270	1685
Climber	520	122	642
Cucumber	972	79	1051
Eat	2462	561	3023
Germ	566	249	815
Mammy	894	154	1048
Rodent	718	349	1067
Salty	543	727	1270
Total	9755	2907	12662

Table 2: Data statistics: Total number of instances (examples) categorized as slang or non-slang per keyword.

4 Methodology

4.1 Pre-processing

Our approach followed the standard pre-processing procedures, such as eliminating duplicate instances (i.e., repeated text entries with identical content), punctuations, URLs, and usernames, and transforming all text to lowercase. We removed all cases when the target word was a part of the URL or the username during the early filtering of the text during extraction.

4.2 Evaluation

In this study, the evaluation of model performance focused on handling the imbalanced nature of slang detection by using a variety of metrics: weighted and macro-averaged scores alongside the F1 score.

Precision assessed the model's ability to correctly identify slang or non-slang instances without false positives, while recall measured its capacity to capture true examples, minimising false negatives. The F1 score provided a balanced view, combining precision and recall, which is particularly crucial given the class imbalance between slang and non-slang. Macro-averaged scores treated all classes equally, highlighting performance in minority classes like slang, whereas weighted scores reflected real-world class distributions by accounting for their relative frequency. Additionally, accuracy offered a straightforward measure of overall correct classifications.

4.3 Model Architecture

4.3.1 Traditional Machine Learning-based Approach

We investigate several Machine Learning models, including Random forest (RF) (Breiman, 2001), logistic regression (LR) (Menard, 2002), Support Vector Machines (SVM) (Hearst et al., 1998), Adaptive Boosting (AdaBoost) (Freund and Schapire, 1996) and Category Boosting (Catboost) (Prokhorenkova et al., 2018). The classifiers are trained using the default parameters provided in Table 7. In our experiments, the following features were used:

- TF-IDF: We employed Term Frequency-Inverse Document Frequency (TF-IDF) (Sparck Jones, 1972) for feature extraction in traditional models, focusing on the importance of word frequency to capture the most relevant terms within the documents.

- TF-IDF and N-gram: Additionally, we utilized n-gram (Sidorov et al., 2014) in combination with TF-IDF. This approach provides valuable contextual insights, enhancing the model's ability to manage negations and ambiguity.

4.3.2 Deep learning Approach fed with Contextual and Static embeddings

Convolutional Neural Networks (CNN) (LeCun et al., 1995) and Bidirectional Long Short-Term Memory (BiLSTM) are frequently used in natural language processing, especially for text classification tasks. CNNs excel at feature extraction from structured data like text, enabling efficient classification (Lai et al., 2015). BiLSTMs, on the other hand, process sequences in both directions, capturing contextual information more effectively (Liu and Guo, 2019; Pei et al., 2019), which makes them particularly suitable for tasks requiring deep contextual understanding, such as slang detection.

- FastText embeddings (wiki-news-300d-1M.vec): These embeddings consist of 1 million word vectors trained on the Wikipedia 2017, UMBC webbase corpus, and statmt.org news dataset (16B tokens) (Grave et al., 2018).

- BERT embeddings (bert-base-uncased) (Devlin et al., 2018).

- GloVe embeddings: GloVe is a pre-trained word embedding model developed from a vast text corpus, utilising an algorithm known as"co-occurrence matrix factorisation" for its training (Pennington et al., 2014).

The embeddings were employed sequentially to train the CNN and BiLSTM models. Texts were converted to sequences using the Keras Tokenizer, with all sequences padded to the maximum sequence length in the dataset. The resulting sequences, along with the embedding matrix derived from the word embeddings, were used to train the deep learning models. To achieve optimal performance, hyperparameter tuning was conducted on both the CNN and BiLSTM models using Keras Tuner's Random Search with 50 trials. This technique systematically explored a range of hyperparameter configurations, such as the number of units in the layers, dropout rates, and types of optimisers, to identify the best parameters for each model. The best parameters used for the deep learning models are detailed in Table 6.

4.3.3 Transformer methods for slang detection

Our transformer-based models, BERT (Devlin et al., 2018), ALBERT (Lan, 2019), RoBERTa (Liu, 2019), and XLNet (Yang, 2019) were trained using five-fold cross-validation over 30 epochs. Each model differs in architecture, with BERT using bidirectional context learning, ALBERT focusing on efficiency through parameter sharing,

[9]Keras refers to embeddings created using https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding, which are randomly initialised and trained during the model's learning process on task-specific data.

Models	Features	Non-slang			Slang			Avg. Macro Scores			Avg. Weighted Scores			Acc
		Pr	Rec	F1	Pr	Rec	F1	Pr_M	Rec_M	F1_M	Pr_W	Rec_W	F1_W	
LR	TF-IDF	0.84	0.97	0.90	0.79	0.36	0.49	0.81	0.67	0.70	0.83	0.83	0.81	0.83
SVM-RBF		0.83	0.98	0.90	0.84	0.35	0.49	0.83	0.66	0.70	0.83	0.83	0.81	0.83
SVM-Lin		0.85	0.96	0.90	0.76	0.41	0.53	0.80	0.69	0.72	0.82	0.83	0.81	0.83
RF		0.84	0.97	0.90	0.82	0.37	0.51	0.83	0.67	**0.71**	0.83	0.84	0.81	**0.84**
AB		0.83	0.96	0.89	0.72	0.36	0.48	0.78	0.66	0.68	0.81	0.82	0.80	0.82
CB		0.84	0.97	0.90	0.77	0.36	0.49	0.80	0.67	0.70	0.82	0.83	0.80	0.83
LR	TF-IDF + n-grams	0.84	0.97	0.90	0.81	0.37	0.51	0.82	0.67	0.70	0.83	0.84	0.81	0.84
SVM-RBF		0.84	0.98	0.90	0.85	0.37	0.51	0.85	0.67	0.71	0.84	0.84	0.81	0.84
SVM-Lin		0.85	0.96	0.90	0.76	0.42	0.54	0.81	0.69	**0.72**	0.83	0.84	0.82	**0.84**
RF		0.84	0.98	0.90	0.85	0.37	0.52	0.84	0.68	0.71	0.84	0.84	0.82	0.84
AB		0.84	0.96	0.89	0.72	0.37	0.49	0.78	0.66	0.69	0.81	0.82	0.80	0.82
CB		0.84	0.97	0.90	0.80	0.37	0.50	0.82	0.67	0.70	0.83	0.83	0.81	0.83
BiLSTM	GloVe	0.88	0.92	0.90	0.69	0.56	0.62	0.78	0.74	0.76	0.83	0.84	0.83	0.84
	BERT	0.86	0.96	0.91	0.78	0.47	0.58	0.82	0.72	**0.75**	0.84	0.85	0.83	**0.85**
	FastText	0.88	0.91	0.90	0.66	0.58	0.62	0.77	0.75	0.76	0.83	0.84	0.83	0.84
BiLSTM-CRF (full features) (Pei et al., 2019)	Keras [9]	0.84	0.88	0.86	0.52	0.45	0.48	0.68	0.66	0.67	0.77	0.78	0.77	0.78
CNN	GloVe	0.87	0.87	0.87	0.61	0.61	0.61	0.74	0.74	0.74	0.81	0.81	0.81	0.81
	BERT	0.87	0.93	0.90	0.70	0.52	0.60	0.78	0.73	0.75	0.83	0.84	0.83	0.84
	FastText	0.88	0.91	0.90	0.66	0.58	0.62	0.77	0.75	**0.76**	0.83	0.84	0.83	**0.84**
CNN-CRF (full features) (Pei et al., 2019)	Keras	0.87	0.90	0.89	0.63	0.56	0.59	0.75	0.73	0.74	0.82	0.82	0.82	0.82

Table 3: Machine learning and deep learning results on test data.

Model	Non-slang			Slang			Avg. macro scores			Avg. weighted scores			Acc
	Pr	Rec	F1	Pr	Rec	F1	Pr_M	Rec_M	F1_M	Pr_W	Rec_W	F1_W	
BERT-large-uncased	0.90	0.94	0.92	0.76	0.63	0.69	0.83	0.79	**0.80**	0.86	0.87	0.87	**0.87**
RoBERTa-large	0.88	0.91	0.90	0.66	0.58	0.62	0.77	0.75	0.76	0.84	0.84	0.83	0.84
XLNET-large-cased	0.87	0.95	0.90	0.60	0.49	0.54	0.73	0.72	0.72	0.83	0.82	0.82	0.82
ALBERT-xxlarge-v2	0.89	0.94	0.92	0.76	0.61	0.68	0.82	0.82	0.80	0.86	0.86	0.85	0.86
gpt-4o-mini	0.91	0.92	0.91	0.72	0.66	0.69	0.81	0.79	**0.80**	0.86	0.86	0.86	**0.86**
gpt-4o	0.85	0.90	0.89	0.76	0.63	0.69	0.78	0.78	0.80	0.85	0.85	0.85	0.85

Table 4: Transformer and large language models (LLMs) results on test data.

RoBERTa improving on BERT by training on larger datasets, and XLNet utilising a permutation-based method for enhanced context understanding. Though trained similarly, each model has unique strengths in language processing. The transformers are trained using the parameters provided in table 6.

4.3.4 Large Language Models

We fine-tuned two Large Language Models (LLMs) for our experiments:

- gpt-4o (version gpt-4o-2024-08-06)[10]: A larger model with advanced capabilities.
- GPT-4o-mini (version gpt-4o-mini-2024-07-18)[11]: A smaller and more cost-effective version of gpt-4o, designed for similar tasks at reduced computational expense.

Both models were fine-tuned using 3 epochs, with the generation temperature set to zero to minimise randomness.

[10] https://platform.openai.com/docs/models/gpt-4o

[11] https://platform.openai.com/docs/models/gpt-4o-mini

5 Results

Table 3 presents an evaluation of machine learning models for slang detection using two sets of features: TF-IDF alone and TF-IDF combined with n-grams. The results demonstrate that incorporating n-grams into TF-IDF generally enhances model performance across various metrics. Specifically, models using TF-IDF with n-grams achieved higher precision, recall, and F1 scores for the "Slang" category compared to using TF-IDF alone, indicating an improved balance between sensitivity and specificity. Overall, models utilising the combined features maintained or improved accuracy, with several models reaching an accuracy of 0.84. The RF model with TF-IDF + n-grams can be considered the most effective overall.

The Table also details the performance outcomes of deep learning models for detecting slang using different word embeddings across various evaluation metrics. The results show that the BiLSTM model with BERT embeddings achieves the best overall performance, with an accuracy of 0.85. This configuration demonstrates particularly strong re-

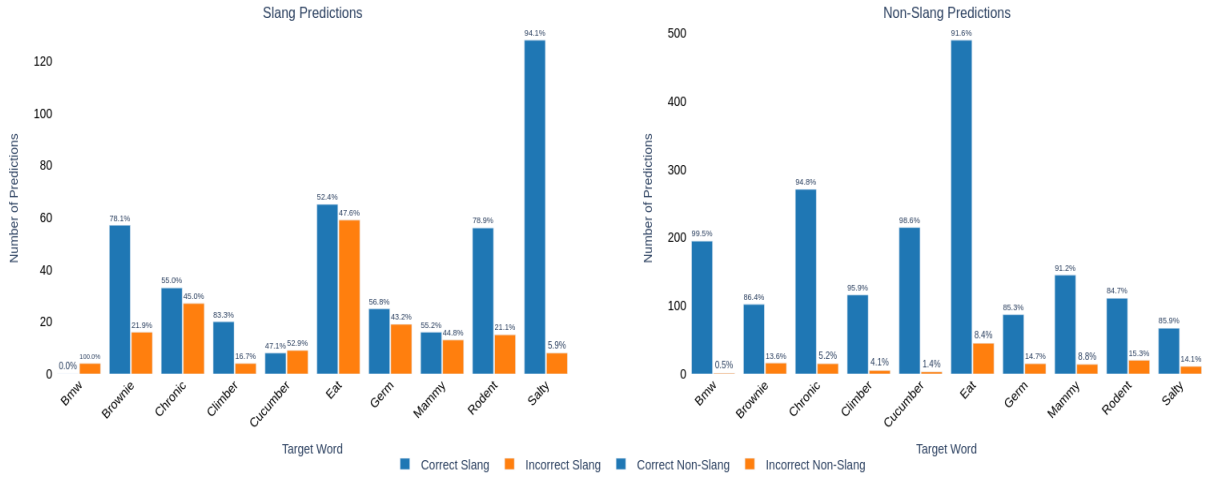

Figure 1: The chart offers a detailed breakdown of the BERT-large-uncased model's accuracy for each target word, categorising sentences that contain either slang or non-slang elements. Each bar represents the number of predictions categorised as correct or incorrect for both slang and non-slang, with percentage labels for enhanced clarity.

sults in the slang category, achieving a precision of 0.75 and a weighted F1 score of 0.83, highlighting its efficacy in accurately identifying and classifying slang. Additionally, this model excels in macro and weighted average scores, reflecting its robustness across diverse linguistic contexts.

Our analysis indicates that incorporating additional linguistic features, such as Part-of-Speech (POS) tagging and Pointwise Mutual Information (PMI), following the approach by (Pei et al., 2019), does not improve the performance of deep learning models for slang detection. Specifically, models like BiLSTM-CRF and CNN-CRF, which integrate these features, exhibit reduced accuracy and lower F1 scores, particularly in detecting slang, compared to simpler models without these features.

The results presented in Table 4 detail the performance of various transformer-based models in detecting slang within textual data. The BERT-large-uncased model emerges as the most effective, achieving the highest accuracy of 0.87, with robust precision of 0.90, recall of 0.94 for non-slang, and notable F1 scores of 0.69 for slang detection. The improvement offered by BERT-large-uncased is particularly significant when compared to earlier results from machine learning models such as SVMs or RF, which typically show lower adaptability to the semantic complexities of slang. Similarly, when compared to basic deep learning models like CNNs or standard BiLSTM, which leverage bidirectional context, the fine-tuned language understanding of BERT-large-uncased provides a more refined analysis of text, resulting in higher accuracy and better generalisation across varied slang expressions. Figure 1 shows a detailed breakdown of the BERT-large-uncased model's performance, and Figure 2 presents its confusion matrix, illustrating classification accuracy for slang and non-slang detection.

The performance of LLMs is summarised in Table 4, where GPT-4o-mini slightly outperforms GPT-4o in slang detection. GPT-4o-mini achieves a higher overall accuracy, 0.86, and delivers better performance across all evaluation metrics, particularly in the slang category.

6 Error Analyses

We performed an in-depth error analysis of our top-performing model for slang detection. This analysis involved sampling 100 misclassified instances to investigate the underlying causes of errors. These errors were then categorised into several categories, as illustrated with examples in Table 5:

- **Bad neighbours (23%):** During error analysis, we observed that many instances were misclassified due to the influence of problematic neighbouring words. These words, such as abusive language, drug references, or harsh tones, skew the model's classification. The meaning of slang often depends heavily on the surrounding context.

Error Category	Examples	Reason for Misclassification	Gold Label → Predicted Label
Bad Neighbors	I think y'all understand the intense hate and fear for that **rodent**-looking motherfucker.	The strong slang word "motherfucker" triggers misclassification, while "rodent" adds a negative tone but isn't slang.	Non-Slang → Slang
Proper Nouns	Wow, believe still remember **brownie** smile song girl scout memories. Good burger, man. I wish you could come to the sweet brownie party.	Phrases like "Brownie Smile Song" and "sweet brownie party" are proper nouns. Informal phrasing misleads the model into treating them as slang.	Non-Slang → Slang
Lost in Length	Post-1960s growth small, expensive underclass resulted in structural problems... **chronic** joblessness and welfare dependency.	Long, complex sentences make identifying context difficult, and pre-processing can reduce clarity.	Non-Slang → Slang
Polysemy	@Officer_Grayson Once **a germ**, always **a germ**. He's as unclean as pork.	The word "germ" has multiple meanings, either as a microorganism or an insult. Lack of clear context causes errors.	Non-Slang → Slang
Polysemy	As a tiny **rodent**... I see things from a unique angle. Like that guy over there... he's not wearing underpants.	The metaphor "tiny rodent" was interpreted literally instead of as slang, leading to misclassification.	Slang → Non-Slang
Ambiguity	The book's protagonist is **a mammy** figure who is both nurturing and deeply flawed, becomes a symbol of resistance against systemic oppression.	The context links "mammy" to literary analysis, suggesting non-slang usage, but informal or stereotypical connotations mislead the model.	Non-Slang → Slang
Ambiguity	My mom is really starting to get on my fucken nerves being the **germ** freak she is.	The word "germ" can be literal (bacteria) or slang (obsession with cleanliness). Ambiguous context caused misclassification.	Slang → Non-Slang
Unknown	Ugh, can't wait to **eat** something after this workout! Abs are killing me, lol hoebag move, though.	Informal abbreviations like "lol" and rare slang terms like "hoebag" confuse the model. Structure does not match patterns.	Non-Slang → Slang

Table 5: Examples of misclassified samples for each error category.

- **Proper nouns (10%):** Proper nouns, especially those appearing as bi-grams or tri-grams, can confuse the model due to their compact and informal structure. These structures may be misinterpreted as slang or colloquial expressions, particularly when they lack clear distinguishing features.

- **Lost in length (13%):** Long sentences with multiple clauses or overly concise phrases due to pre-processing can challenge the model. Pre-processing often involves removing stop words or punctuation, which can reduce the context available for accurate classification.

- **Polysemy (17%):** Words with multiple meanings can lead to misclassification when the context is unclear or insufficient. Polysemous terms often require a broader or more detailed context for the model to interpret them correctly.

- **Ambiguity (7%):** Ambiguity arises when words or phrases can have multiple interpretations, and the context does not sufficiently clarify their meaning. For example, the word "salty" might mean "bitter" in a slang sense or "overly seasoned" in a literal sense. Similarly, terms like "germ" or "chronic" can have both literal and figurative meanings, leading to misclassification if the context is vague.

- **Unknown (30%):** Texts containing unconventional abbreviations, rare slang, or novel constructions are particularly challenging for the model. These terms often deviate from standard language patterns, making them difficult for the model to classify accurately.

7 Conclusion

We present SlangTrack (ST) Dataset, a corpus designed to enhance fine-grained slang detection. It overcomes the limitations of previous corpora, which do not distinguish between words with both slang and non-slang meanings. To evaluate its effectiveness, we explored approaches across ML, DL, fine-tuned LMs, and LLMs. Our experiments show that fine-tuning models, particularly BERT-large-uncased, delivers strong performance in detecting both slang and non-slang uses.

8 Limitations

Despite the promising results of this study, there are several limitations that leave room for further research. First, the corpus used in this work could benefit from a broader representation of slang, especially rare and evolving expressions. This limitation impacts the generalizability of the model across a wider variety of slang terms. To address this, future work could explore data augmentation techniques to generate synthetic examples, enhancing the diversity of the slang corpus.

Additionally, the current approach does not fully exploit the potential of large language models (LLMs) in distinguishing between subtle nuances in slang and non-slang expressions. Incorporating LLMs to assess their reasoning and contextual understanding in this domain could lead to more sophisticated slang detection mechanisms.

Lastly, while the models employed in this study show reasonable efficacy, they may struggle with edge cases or ambiguous slang terms. Future research could benefit from ensemble methods, combining multiple models to create a more robust and accurate system for slang detection. This would likely improve overall performance, especially in handling less common or context-dependent slang terms.

References

Michael Adams. 2012. *Slang: The people's poetry*. Oxford University Press.

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte Im Walde. 2020. CCOHA: Clean corpus of historical American English. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6958–6966.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of african-american english. *arXiv preprint arXiv:1608.08868*. Accessed: 2024-10-16.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45:5–32.

Mary Bucholtz. 2006. Word up: Social meanings of slang in California youth culture. *A cultural approach to interpersonal communication: Essential readings*, 243(267):54.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Julie Coleman. 2012. *The life of slang*. Oxford University Press, USA.

Mark Davies. 2012. Expanding Horizons in Historical Linguistics with the 400-Million Word Corpus of Historical American English. *Corpora*, 7(2):121–157.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Shehzaad Dhuliawala, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016. SlangNet: A WordNet like resource for English slang. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4329–4332.

Bethany K Dumas and Jonathan Lighter. 1978. Is slang a word for linguists? *American Speech*, 53(1):5–17.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2014. Diffusion of lexical change in social media. *PloS one*, 9(11):e113114.

Yoav Freund and Robert E. Schapire. 1996. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, pages 148–156, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Jonathon Green. 2015. *The vulgar tongue: Green's history of slang*. Oxford University Press, USA.

Anshita Gupta, Sanya Bathla Taneja, Garima Malik, Sonakshi Vij, Devendra K Tayal, and Amita Jain. 2019. SLANGZY: A fuzzy logic-based algorithm for English slang meaning Selection. *Progress in Artificial Intelligence*, 8:111–121.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Chloe Kiddon and Yuriy Brun. 2011. That's what she said: double entendre identification. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 89–94.

Vivek Kulkarni and William Yang Wang. 2018. Simple models for word formation in slang. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1424–1434.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Z Lan. 2019. Albert: A LiteBert for Self-supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942*.

Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Gang Liu and Jiabao Guo. 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Theo Lynn, Patricia Takako Endo, Pierangelo Rosati, Ivanovitch Silva, Guto Leoni Santos, and Debbie Ging. 2019. A comparison of machine learning approaches for detecting misogynistic speech in urban dictionary. In *2019 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, pages 1–8. IEEE.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Scott Menard. 2002. *Applied logistic regression analysis*. 106. Sage.

Ke Ni and William Yang Wang. 2017. Learning to explain non-standard English words and phrases. *arXiv preprint arXiv:1709.09254*.

Alok Ranjan Pal and Diganta Saha. 2015. Detection of slang words in e-data using semi-supervised learning. *arXiv preprint arXiv:1702.04241*.

Eric Partridge. 2015. *Slang: To-day and yesterday*. Routledge.

Zhengqi Pei, Zhewei Sun, and Yang Xu. 2019. Slang detection and identification. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 881–889.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Thomas Rauber and Karsten Berns. 2011. Kernel multilayer perceptron. In *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 337–343. IEEE.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2014. Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, 41(3):853–860.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Zhewei Sun, Richard Zemel, and Yang Xu. 2022. Semantically informed slang interpretation. *arXiv preprint arXiv:2205.00616*.

Liang Wu, Fred Morstatter, and Huan Liu. 2018. Slangsd: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification. *Language Resources and Evaluation*, 52:839–852.

Zhilin Yang. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

9 Appendices

9.1 Implementation Details

- **Hardware Configuration:**
  - **Machine Learning and Deep Learning Models:** All experiments, including both machine learning and deep learning models, were conducted on a single Tesla V100 GPU equipped with 32 GB of RAM. These experiments were performed using the Google Colab Pro+ platform [12], which provides enhanced computational capabilities and extended runtime for intensive tasks.
  - **Fine-Tuning Experiments (LM and LLMs):** Fine-tuning of language models was executed on a single NVIDIA A100 GPU, which features 80 GB of RAM. This setup was specifically chosen to handle the increased computational demands of optimising large-scale models for high accuracy in NLP tasks.

- **Software Frameworks** : All experiments and model implementations were carried out using Python 3.10.12. For evaluation, we used the metrics implemented in the scikit-learn toolkit [13]. For reproducibility, we set the seed parameter to 42 in all experiments.
  - **Machine Learning models:** Different machine learning frameworks were utilised. Specifically, the scikit-learn toolkitwas employed to develop the Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and AdaBoost (AB) models. Additionally, Additionally, the CatBoost library [14] was utilised to develop our CatBoost models, leveraging its advanced capabilities for processing categorical data efficiently.
  - **Deep Learning Models:** TensorFlow [15] and Keras [16] were the primary frameworks employed for deep learning tasks. These frameworks facilitated the development of complex neural network architectures, including Convolutional Neural Networks (CNNs) and BiLSTM for text classification. Utilising TensorFlow's powerful computational capabilities, models were optimised with techniques such as extensive hyperparameter optimisation using Keras Tuner [17]. Moreover, scikit-learn utilities were integrated for performance evaluation, providing metrics such as precision, recall, and F1-score, which were crucial for assessing model effectiveness across various classes.
  - **Transformer Models:** For our transformer-based models, we utilized the SimpleTransformers [18] and HuggingFace Transformers [19] libraries. These tools streamlined the loading and fine-tuning of pre-trained models such as BERT, ALBERT, RoBERTa, and XLNet, enhancing our capacity to efficiently conduct NLP tasks with high accuracy. also Scikit-learn is employed for stratified K-fold cross-validation.

[12] https://https://colab.research.google.com/
[13] https://scikit-learn.org/stable/
[14] https://catboost.ai
[15] https://www.tensorflow.org/
[16] https://keras.io/
[17] https://keras-team.github.io/keras-tuner/
[18] https://simpletransformers.ai
[19] https://huggingface.co/transformers/

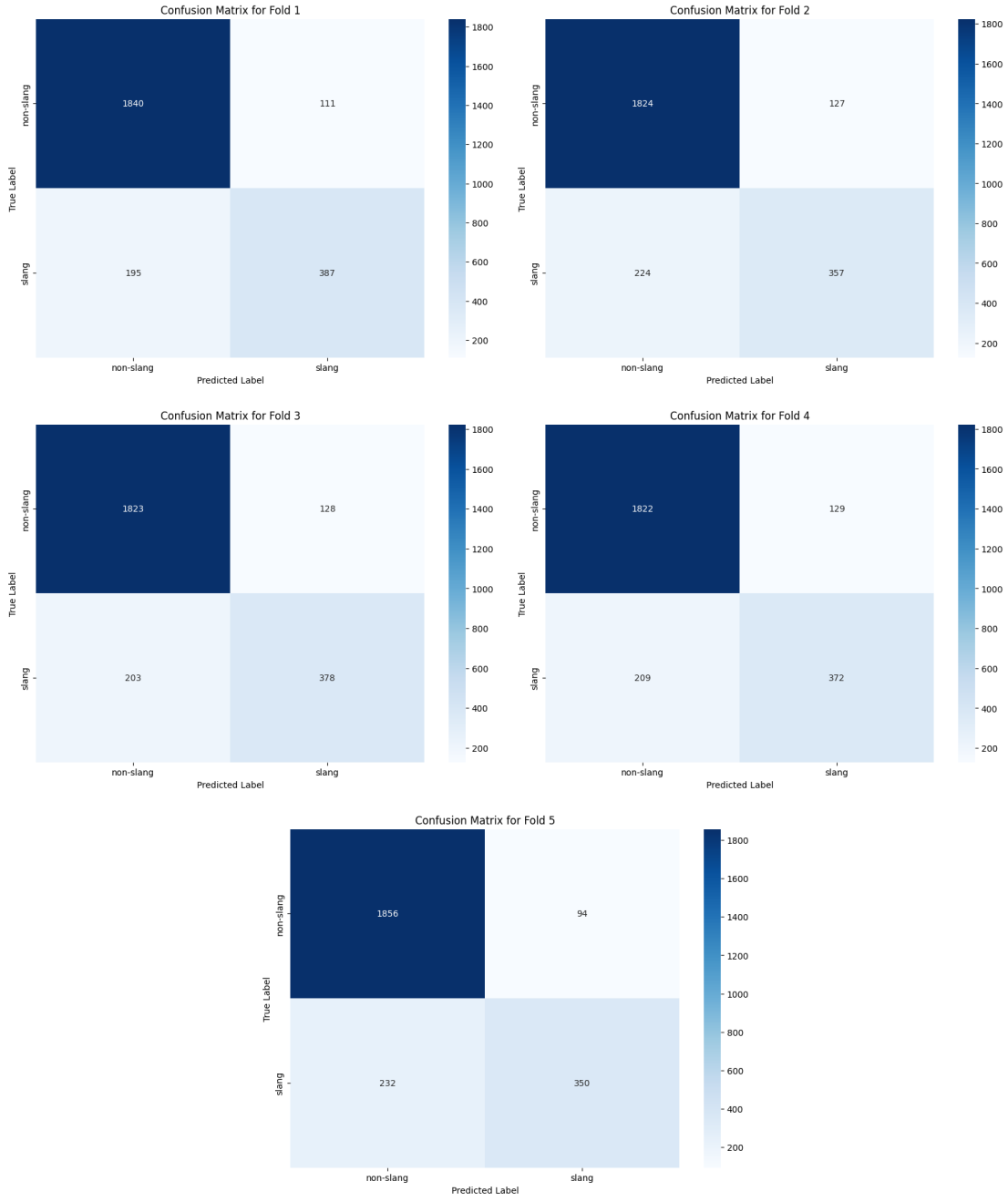

Figure 2: Confusion matrices for the BERT-large-uncased model, illustrating its classification performance for both slang and non-slang categories.

Models	Parameters
Finetuned Models (BERT, RoBERTa, ALBERT, XLNET)	num_train_epochs= 30, learning_rate= 4e-5, train_batch_size= 64, eval_batch_size= 64
Bilstm + GloVe embeddings	Embedding Dimension= 300, BiLSTM Units= 32, Dense Units= 64, Dropout Rate= 0.2, Optimiser= 'Adam', Learning Rate= 2.93e-03, epochs= 30
Bilstm + FastText embeddings	Embedding Dimension= 300, BiLSTM Units= 256, Dense Units= 64, Dropout Rate= 0.3, Optimiser= 'Adam', Learning Rate= 7.02e-04, epochs= 30
Bilstm + BERT embeddings	Embedding Dimension= 768, BiLSTM Units= 128, Dense Units= 128, Dropout Rate= 0.3, Optimiser= 'rmsprop', Learning Rate= 7.44e-03, epochs=30
CNN + GloVe Embeddings	Embedding Dimension= 300, conv_units= 128, Dense Units= 32, Dropout Rate= 0.4, Optimiser= 'Adam', Learning Rate= 1.00e-03, epochs= 30
CNN + FastText embeddings	Embedding Dimension= 300, Conv Units= 224, Dense Units= 128, Dropout Rate= 0.2, Optimizer= 'rmsprop', Learning Rate= 1e-03, epochs= 30
CNN + BERT embeddings	Embedding Dimension= 768, Conv Units= 64, Dense Units= 32, Dropout Rate= 0.3, Optimiser= 'rmsprop', Learning Rate= 1e-03, epochs= 30

Table 6: Important parameters of deep models.

Estimator	Hyperparameters
LR	penalty= 'l2', C= 1.0, solver= 'lbfgs', max_iter= 100, verbose= 0
SVM	C= 1.0, gamma= 1.0, cache_size= 200, verbose= False, max_iter= -1, random_state= None, class_weight= None
RF	n_estimators= 100, max_depth= 10, min_samples_split=2
AdaBoost	n_estimators= 50, learning_rate= 1.0, base_estimator= DecisionTreeClassifier, algorithm = 'SAMME.R'
CatBoost	iterations= 1000, learning_rate= 0.03, depth= 6, verbose=True

Table 7: Important parameters for Machine Learning models.