IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 24 December 2024, accepted 17 January 2025, date of publication 28 January 2025, date of current version 4 February 2025.

*Digital Object Identifier 10.1109/ACCESS.2025.3535862*

APPLIED RESEARCH

# Urdu Toxic Comment Classification With PURUTT Corpus Development

**HAFIZ HASSAAN SAEED, TAHIR KHALIL, AND FAISAL KAMIRAN**
Department of Computer Science, Information Technology University, Lahore 54600, Pakistan

Corresponding author: Hafiz Hassaan Saeed (hassaan.saeed@itu.edu.pk)

**ABSTRACT** This study addresses the critical gap in toxic comment classification in Urdu, a widely spoken language devoid of high-quality standard datasets. To address this gap, we employed an existing labeled Roman Urdu (RU) corpus, which was developed originally for Roman Urdu toxic comment classification, and supplemented that corpus by adding its Urdu equivalent transliterations. The motivation behind such an extension is twofold: firstly, to provide a large comprehensive dataset for the classification of toxic comments in Urdu; secondly, to facilitate bidirectional transliteration between Urdu and RU, however, transliteration is currently outside the scope of this study and is envisioned as a future research direction. We introduce the extended corpus as PURUTT (Parallel Urdu and Roman Urdu Corpus for Toxic Comments and Transliteration), boasting 72,771 labeled comments as parallel comments in both Urdu and Roman Urdu scripts. Specific to Urdu toxic comment classification, our methodology begins by training those classification models that were trained on the original Roman Urdu corpus. We leverage pre-trained Word2Vec and FastText Urdu word embeddings to evaluate model performance through transfer learning. Furthermore, we fine-tune five multilingual large language models capitalizing on their inherent multilingual capabilities. To further enhance the classification performance, this study proposes an ensemble approach that aggregates the strengths of multiple base models. Our extensive empirical validation demonstrates the superiority of the ensemble model, achieving a state-of-the-art F1-score of 91.65% on PURUTT, setting a benchmark F1-score on PURUTT corpus for Urdu toxic comment classification.

**INDEX TERMS** Urdu, Urdu parallel corpus, Urdu toxic comments, Urdu toxic comment classification, toxic comment classification, transfer learning, ensemble.

## I. INTRODUCTION

The uncurbed proliferation of toxic language has come to the fore as an unenviable global social calamity [1]. With the increased internet connectivity, more people use social media platforms to express their opinions or emotions while some take advantage of this freedom and hurt the self-esteem of other individuals or groups [2]. The inimical repercussions of such toxic texts on social media users and society can drive the sufferers to extreme measures [3], such as emotional and psychological issues [4] and even suicide [5]. Social media platforms like Twitter and Facebook try to restrain toxic content and spend millions of euros on the detection and mitigation of toxic comments to decrease the damaging effects on their users [6]. However, given the sheer number and variety of languages spoken across the globe, filtering out toxic content in all languages is a challenging task [7].

To that end, several studies have been conducted for automated toxic language detection in various languages like English, French, Arabic, Dutch, Italian, etc., but only a few for Urdu. Urdu is a widely spoken language with more than 237 million speakers[1] worldwide. Pakistan, a nation of 220 million people [8], has Urdu as its official language [9]. It is spoken in several countries other than Pakistan including India, the Middle East, Europe, the United Kingdom, the United States, and Canada [10]. Urdu belongs

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

[1] Urdu. Ethnologue, https://www.ethnologue.com/language/urd (Accessed December 22 2024).

to the family of Indo-Aryan languages written in the Perso-Arabic script [11] from right to left [12] with an extended set of Arabic characters. Language processing in Urdu is comparatively complex as its grammar and morphology are a blend of Arabic, Persian, Sanskrit, Turkish, and a few other languages [13]. It lacks capital or small letters, has a free word order characteristic [10], and a tendency to code-switch [14], i.e., to accept lexical features and vocabulary from other languages. For example, “ہے۔ cricketer عمران خان ایک اچھا” translated as “Imran Khan is a good cricketer.”, the word “cricketer” is written from left to right instead of right to left.

Urdu is a low-resource language, and most Urdu corpora available for various language processing tasks are limited in size. To address this deficiency pertaining to toxic comment detection in Urdu, we had two choices: (i) create and label an entirely new corpus, or (ii) leverage an existing labeled Roman Urdu corpus and generate Urdu equivalents through transliteration as Roman Urdu is an informal writing script of Urdu. We opted for the latter option, selecting a large available Roman Urdu toxic comments corpus, which is referred to as RUT corpus, developed by [5] having 72,771 labeled comments with a strong inter-annotator agreement.

Initially, we attempted to generate transliterations from Roman Urdu (RU) to Urdu using iJunoon[2] Roman Urdu to Urdu webpage. However, the resultant transliterations were found to have transliteration errors with a high frequency and lacked reference to the context of the given RU sentences. To rectify this, we engaged native Urdu speakers to manually review the iJunoon-generated transliterations and correct the mistakes they found. This resulted in the extension of the RUT corpus with Urdu equivalent transliterations. We name the extended version of the RUT corpus as PURUTT which refers to “Parallel Urdu and Roman Urdu corpus for Toxic Comments and Transliteration”. The accuracy of iJunoon-generated transliterations is assessed, compared to human-reviewed versions, using both word error rate (WER) and character error rate (CER) metrics. To the best of our knowledge, this is the first instance of a dataset being transliterated between two writing scripts to create a language processing resource for different script variations of a single language.

Recent advancements in deep learning techniques or large language models have significantly improved performance in various natural language processing (NLP) tasks, surpassing traditional machine learning approaches. Word embeddings, in conjunction with deep learning models, dominate simple bag-of-words approaches. Our experimentation for Urdu toxic comment classification entails training classification models that were previously trained on the original RUT corpus, as well as exploring additional deep learning models and fine-tuning five transformer-based multilingual large language models. Furthermore, this paper compares the performance of task-tailored word embeddings and pre-trained word embeddings in the context of a resource-constrained language like Urdu.

In summary, the significant contributions of this paper are as follows:

- The foremost contribution of this paper is the detection of toxic comments in an under-resourced Urdu language. We classify given Urdu text into either toxic class or non-toxic class.
- To address the scarcity of labeled data in Urdu NLP, we introduce PURUTT, the first and the largest human-reviewed parallel corpus of Roman Urdu (RU) and standard Urdu comments. We extended an existing Roman Urdu corpus by adding equivalent Urdu transliterations. The PURUTT corpus serves two key purposes:
  - Facilitating toxic comment classification in Urdu (and Roman Urdu).
  - Enabling bidirectional transliteration between Urdu and Roman Urdu. However, the transliteration is out of the scope of this study.

  This resource significantly bridges the gap in labeled data for Urdu Natural Language Processing (NLP) tasks. The corpus will be publicly available to stimulate further advancements in Urdu NLP.
- We conduct an extensive empirical evaluation of several classification models on the developed Urdu corpus. Our experimentation covers classical machine learning algorithms, deep learning approaches with both pre-trained and task-tailored word embeddings (learned as part of model training) for Urdu language understanding, as well as fine-tuning of multilingual large language models.
- This study demonstrates the significant potential of ensemble methods in improving classification performance. By combining multiple individual models into several ensemble groups, we identify a group that achieves superior evaluation scores on the Urdu segment of the PURUTT corpus compared to any individual model.

The rest of the paper is structured as follows: Section II discusses the existing work pertinent to the scope of this research study, Section III describes the process of development of Urdu corpus for toxic comment classification, Section IV describes the overall methodology adopted in this paper along with the classification models employed, Section V covers the experimental details and the results of the experiments, finally, Section VI concludes this study.

## II. RELATED WORK

The detection of toxic comments in natural language processing has been a longstanding research area, with various studies employing interchangeable terms such as hate speech [15], cyberbullying [16], abusive language [17], offensive language [18], harassment [19], profanity detection [20],

[2] https://www.ijunoon.com/transliteration/roman-to-urdu

threatening language [21], uncivil language [22], and even the toxic term itself [23].

Existing research on toxic comment classification can be broadly categorized into two streams: rule/lexicon-based approaches and learning-based approaches. Rule/Lexicon-based approaches rely on developing rules or a lexicon or a dictionary of toxic words or phrases and classifying comments as toxic or non-toxic. For example, [24], [25], [26] used lexicon-based approaches. However, these approaches often struggle with capturing nuanced and context-dependent toxicity.

Learning-based methods have emerged as a more powerful alternative, leveraging large datasets and sophisticated algorithms to learn complex patterns associated with toxic language. Classical machine learning models, such as Naïve Bayes [27], Random Forests [28], Support Vector Machines and Logistic Regression [29] have demonstrated the efficacy of techniques in achieving high accuracy rates.

With the development of deep learning, more attention was given to neural-network-based models like simple feed-forward neural networks, convolution neural networks, and recurrent neural networks. Morzhov tested various deep learning algorithms, first individually and then ensemble for detecting and classifying toxicity in 314000 comments prepared and tagged by Google and Jigsaw [30]. Morzhov showed that while testing, the ensemble outperformed and got 0.9870 AUC-ROC score. Luu and Nguyen identified toxic words with 62.23% F1-score on task 5 of SemEval-2021 on the Toxic Spans Detection dataset [31]. They used a combination of BiLSTM-CRF and ToxicBERT classification to train the classification model.

Fang et al. introduced a combination of Bi-GRU and self-attention for cyberbullying classification [32]. They tested their approach on two twitter and one English Wikipedia datasets, containing 16090 and 24783 tweets and 115865 comments respectively, and achieved an F1-score of 0.976. Eronen et al. discovered that higher feature density is negatively correlated to most of the classifiers by 0.798 max F1 score, except for CNNs [33]. They trained it on English, Polish, and Japanese cyberbullying datasets with 300000 tokens, 11041 entries, and 2950 entries respectively. Yi et al. detected profanity in 4.4 million Korean twitter posts and 500000 sentences of Naver movie reviews [20] by using the LSTM model with FastText word embeddings and achieved 96.15% accuracy.

Kumar et al. used a multichannel convolutional bidirectional gated recurrent unit (MCBiGRU) for the detection of toxic comments in a multi-labeled English dataset [34]. The dataset contains 223,549 comments with six labels and they achieved 98.2% ROC AUC beating other models. We have used their models in our experiments too. Wang and Zhang addressed the problems of detecting toxicity in an imbalanced English toxic comment classification dataset [35]. They combined oversampling and cost-effective methods to make an improved Bi-GRU model with the highest Macro Geometric mean score of 0.5936.

Wang and Zhang proposed bidirectional gated recurrent unit and global pooling optimized convolution neural network (BG-GCNN) for binary toxic comment classification in English [36]. Kumar and Sachdeva proposed Bi-GRU-Attention-CapsNet (Bi-GAC) that learns to detect cyberbullying on social media using attention-based Bi-GRU and CapsNet [37]. They tested it on the MySpace and Formspring dataset containing 1753 and 13124 samples respectively and improved around 9% and 3% in F1-score in the respective datasets. Murshed et al. proposed a combination of RNN and optimized Dolphin Echolocation Algorithm, the DEA-RNN [38] and achieved 90.45% accuracy on a 130000 English tweets dataset collected through Twitter API.

More recent studies have focused on utilizing transformer-based architectures, which have shown promising results in detecting toxic comments. For instance, Isaksen and Gambäck discovered that the BERT-based models get confused in detecting hate or offense while pre-trained models performed better [39]. Davidson et al. used BERT and Logistic Regression for automatic identification of incivility in social media [40]. Their best-performing model achieved 0.802 F1-score. Özler et al. reached a new state-of-the-art achieving 0.990 AUC score in [22] using BERT-based model for detecting incivility in multi-label, multi-domain English datasets like Local-news comments, Wikipedia comments, and Russian troll tweets.

Koufakou et al. proposed HurtBERT with a max 0.838 F1-macro score [41]. They combined BERT with lexicons to improve abusive language detection. They tested their approach on Waseem, Davidson, Founta, HatEval, OLID and AbuseEval English datasets with 16488, 24783, 99799, 11971, 14100, and 14100 instances respectively. Chinagundi et al. introduced an ensemble model for hate, offensive, or profane tweet classification in a code-mixed language dataset containing 76601 texts [42]. They used Hate-BERT along with transformer-based embedding and achieved a 79% Macro F1 score.

Park and Kim proposed a number of BERT variants to better classify the abuse in the Korean language into 5 classes [43]. The dataset had 3902 Korean conversations and their results showed that BERT with CNN after data augmentation outperformed others reaching 47.3% F1 score. Rezvani and Beheshti combined the contextual and lingual features of 2188 Instagram posts and 7321 Twitter tweets to come up with a more robust cyberbullying detection transformer-based multi-model [44] and achieved 0.87 F1 score.

Zhao et al. studied the performance of pre-trained BERT models for toxicity classification [45]. They compared BERT, RoBERTa, and XLM, and proved that BERT and RoBERTa perform better than XLM and that further fine-tuning improves the downstream toxic classification task. Huang et al. proposed multi-task framework combining abuse detection and emotion classification (MFAE), a combined framework of abuse detection and emotion classification that achieved 82.05% F1-score [46]. They used BERT to represent sentences and then proposed a cross-attention (CA) component in the decoder.

However, most existing research on toxic comment classification focuses on English or other widely-resourced languages. There is a significant gap in understanding the unique challenges and characteristics of toxicity within Urdu. Some preliminary work has been done exploring hate speech [15] or cyberbullying [16] detection in Urdu, but comprehensive studies on toxic comment classification remain limited.

This work aims to bridge this gap by developing a novel dataset to exploit learning-based approaches for classifying toxic comments in Urdu. We investigate the effectiveness of various machine learning, deep learning architectures, including Transformer models, and explore strategies such as ensemble for addressing the specific challenges posed by toxic comment classification in Urdu language.

## III. THE DATASET - PURUTT CORPUS

This section describes the development process of the PURUTT corpus. The primary objective is to develop a corpus for the classification of toxic comments in Urdu. Instead of gathering a corpus from scratch and labeling the collected sentences as toxic or non-toxic, we adopted a strategy where we transliterate an already labeled Roman Urdu (RU) corpus into Urdu since RU is simply the Latin writing script for Urdu. With this approach, we get a single corpus for two Urdu NLP problems i.e., classification of toxic comments in both Urdu and RU, and transliteration. However, we restrict the scope of this paper to toxic comment classification in Urdu solely. We selected a large available RU corpus labeled for RU toxic comment classification, the RUT corpus, developed by [5]. RUT corpus was collected from diverse controversial topics including religion, sectarianism, politics, etc., from YouTube, Facebook, Twitter, and other web sources.

### A. INACCURATE TRANSLITERATION FROM IJUNOON

The initial phase of this study involved utilizing iJunoon, a popular online Urdu and Roman Urdu transliteration tool. A Python script was implemented to automate the transliteration process. However, our analysis revealed significant inaccuracies in the iJunoon output (detailed in Table 1), demonstrating a sole reliance on phonetic transcription without any consideration of the context.

To address these inaccuracies, a rigorous manual review of all iJunoon-generated transliterations was conducted involving two human reviewers who: a) were males between the ages of 20 and 30, b) had a Masters degree, and c) were native Urdu speakers with Urdu as their first language.

Clear guidelines for transliteration correction, accompanied by a GUI-based annotation tool, were provided to the reviewers. A comprehensive discussion session, including a demonstration of the annotation tool, was held to ensure a shared understanding of the correction process and address any ambiguities. These guidelines are detailed in the subsequent section.

### B. TRANSLITERATION CORRECTION GUIDELINES

#### 1) PREAMBLE

Transliteration[3] is not the same as translation.[4] Transliteration is the process of writing words from one writing script to another. Roman Urdu (RU) uses Latin script whereas original Urdu uses Perso-Arabic script. The only goal here is to correctly transliterate the provided Roman Urdu (RU) comments to Urdu.

#### 2) REFERENCE RESOURCES FOR URDU

Urdu is a formal language, hence, attention to transliteration accuracy is vital. Reviewers are advised to consult reputable Urdu dictionaries to get clarification on word meanings for appropriate transliterations from Roman Urdu (RU) to standard Urdu script. Recommended resources include:

(i) Farhang Aasfia[5] by Syed Ahmed Dehlvi
(ii) Feroz Ul Lughat[6] By Maulvi Ferozuddin
(iii) Rekhta[7] Online Urdu Dictionary
(iv) If a word is not found in any of the dictionaries then the reviewer must consult online Urdu newspapers like BBC Urdu,[8] Wikipedia Urdu,[9] or other web pages written in Urdu.

#### 3) COMMENT MODIFICATION

If the iJunoon-generated transliteration from Roman Urdu (RU) to Urdu contains errors, the sentences shown in Urdu must be corrected.

#### 4) TRANSLITERATION NOT TRANSLATION

This is a transliteration correction task, not a translation task, thus if Roman Urdu (RU) sentences contain words that can be translated into Urdu, transliterate instead of translation. For example, “Ahmad ne blackboard saaf kia.” should be transliterated as “احمد نے بلیک بورڈ صاف کیا۔” rather than “احمد نے تختہ سیاہ صاف کیا۔”.

#### 5) CONTEXTUAL CONSIDERATION

The reviewers must take the context of each word into account while transliterating from RU to Urdu. For example, in the comment, “shadi haal mein ghutan kay

[3] https://dictionary.cambridge.org/dictionary/english/transliteration
[4] https://dictionary.cambridge.org/dictionary/english/translation
[5] https://archive.org/details/FarhangAsifiya/00511_Farhang_Asifiya_1/
[6] https://archive.org/details/FerozUlLughat
[7] https://www.rekhtadictionary.com/?lang=ur
[8] https://www.bbc.com/urdu
[9] https://ur.wikipedia.org

**TABLE 1.** iJunoon's Mistransliterations (Errors marked as red).

| # | Roman Urdu | iJunoon Transliterations | Actual Transliterations |
|---|---|---|---|
| 1 | pak sarzamin shad bad | پاک سارزامین شاد بعد | پاک سرزمین شاد باد |
| 2 | Rizvi shatan ka baccha hei | رضوی شاتان کا بچا ہے | رضوی شیطان کا بچہ ہے |
| 3 | tu sigret chodega kab | تو سیجریت چودیجا کب | تو سگریٹ چھوڑیگا کب |
| 4 | Belkool hakeeekat hai yeh | بیلکول حاکیکات ہے یہ | بالکل حقیقت ہے یہ |
| 5 | Tum chudvaogi mere sath | تم چودVاوجی میرے ساتھ | تم چودواؤگی میرے ساتھ |
| 6 | Lakh lanatt tere tay | لاکھ لاناتت تیرے طے | لاکھ لعنت تیرے تے |
| 7 | yaahaan se chal saly chutiye | یاحان سے چل سلی چوتیے | یہاں سے چل سالے چوتیے |
| 8 | saale beshram lodu | سالے بیشرام لودو | سالے بےشرم لوڑو |
| 9 | Assalamu aalaikum walekum asalam | آسلام و آلایکوم وعلیکم اسلام | السلام علیکم وعلیکم السلام |
| 10 | Gand mar Jake Sali bhosdi | گند مر جاکے سالی بجوسدی | گانڈ مار جاکے سالی بھوسڑی |

bais logon ka bura haal ho gaya.'', the word ``haal'' appears twice and each occurrence should be transliterated according to the context. The correct transliteration is ‘‘براحال ہوگیا۔شادی ہال میں گھٹن کے باعث لوگوں کا’’.

### 6) TYPOS HANDLING

If RU sentences contain typos, do not correct those words; instead, substitute an Urdu word that is phonetically equivalent in the transliterated Urdu sentences.

### 7) ACRONYMS/ABBREVIATIONS

If RU sentences contain acronyms or abbreviations, maintain the integrity by using space or zero-width-non-joiner to separate the letters in the transliterated Urdu sentences, e.g., PTI → (پی ٹی آئی), L.D.A. → (ایل.ڈی.اے.).

### 8) HOMOPHONES

If RU sentences include homophones, substitute the appropriate word based on contextual clues in the transliterated Urdu sentences. For example, sadaa → (صدا/سدا), arz → (ارض/عرض), qamar → (قمر/کمر).

### 9) HINDI DIALECT NORMALIZATION

If RU sentences contain Hindi dialect, normalize those words in the transliterated Urdu sentences. For example, fir → (پھر), riwaaz → (رواج), jyadti → (زیادتی).

### 10) HASHTAG HANDLING

If RU sentences contain hashtags of multiple words, phrases, acronyms, or abbreviations, use space or zero-width-non-joiner for the separation in the transliterated Urdu sentences, e.g., #bankhadimrizvi → (#بین خادم رضوی).

### 11) EMOTICONS/URLS

If RU sentences include emoticons/URLs, leave them typed in English in the transliterated Urdu sentences.

### 12) ALREADY SPLIT WORDS

If RU sentences contain already split words, do not join those words in the transliterated Urdu sentences. For example, ley tey → (لے تے), kay liye → (کے لیے).

### 13) MEANINGFUL ALPHA-NUMERIC WORDS

If RU sentences contain alpha-numeric meaningful words, try as much as possible to use the alpha-numeric style in the transliterated Urdu sentences too, e.g., ri8 → (8رائی).

### 14) COMPOUND WORDS WRITTEN AS A SINGLE-WORD

If RU sentences contain compound words written as a single word, separate them if needed in the transliterated Urdu sentences. For example, chashmenam → (چشمِ نم), jaanejahaan → (جانِ جہاں).

### 15) COMPOUND WORDS SEPARABLE WITH ZAIR DIACRITIC

If RU sentences contain compound words separable with zair diacritic, replace those words with zair diacritic in the transliterated Urdu sentences, e.g., wazir e azam → (وزیرِاعظم), khadim e aala → (خادمِ اعلیٰ).

### 16) HYPHENATED COMPOUND WORDS SEPARABLE WITH ZAIR DIACRITIC

If RU sentences contain hyphenated compound words separable with zair diacritic, replace those words with zair diacritic and remove extra hyphens in the transliterated Urdu sentences, e.g., Rabbe-Kareem → (ربِ-کریم), Shab-e-Hijr → (شبِ-ہجر), Haal-e-Dil → (حالِ-دل).

### 17) PROPER NOUNS - PLACES/LOCATIONS

If RU sentences contain proper nouns such as geographical locations (e.g., countries, capitals, cities, provinces, etc.), then:

- Word Boundary Preservation: Maintain the original word boundaries according to the given RU sentence, e.g., ``New Zealand'' → (نیوزیلینڈ).
- Supplementary Resource Exploration: Consult the dictionaries and use the appropriate place/location name in the transliterated Urdu sentences. If they are not found in dictionaries, try searching Wikipedia Urdu, online newspapers in Urdu, or other online web pages explicitly written in Urdu in the listed order and use the appropriate place/location name.
  - Handling Multiple Variants: For a place/location, if multiple Urdu spellings are found then select

the most frequent word in the transliterated Urdu sentences, e.g., different spellings for “kerala” are found to be (کیرلا), (کیرالا), and (کیرالہ), in this case, if (کیرالا) is found to be more frequent, then this word should be selected in the transliterated Urdu sentences.

#### 18) PROPER NOUNS - NAMES STARTING WITH ABDUL/ABDUR

If RU sentences contain proper nouns that begin with Abdul or Abdur, standardize those nouns in the transliterated Urdu sentences, e.g., Abdur Razaaq → (عبدالرزاق), Abdul Qudoos → (عبدالقدوس).

#### 19) CONTEXT-SPECIFIC STANDARDIZATION OF WORDS

If RU sentences contain words that can be standardized according to the context of the sentence, standardize them in the transliterated Urdu sentences. A few use cases are shown:

- One-letter word “s” in “hum s mulaqat” can be standardized to (سے).
- One-letter word “k” can be standardized to (کہ/کو/کی/کے/کا).
- One-letter word “r” can be standardized to (اور).
- One-letter word “n” can be standardized to (نے/نا).
- One-letter word “m” or two-letter word “mn” can be standardized to (میں).
- Two-letter word “gy”/“ge” can be standardized to (جی/گئی/گئے/گی/گے).
- Two-letter word “hn” can be standardized to (ہیں/ہوں/ہاں).
- Two-letter word “ha” can be standardized to (ہے/ہاں).
- Two-letter word “wa” in “pyar wa muhabat” or “fazl wa karam” can be standardized to “(پیار و محبت)”/“(فضل و کرم)” respectively.
- Three-letter word “nae” can be standardized to (نے/نئے/نہیں/نئی).

### C. PURUTT CORPUS

Following the manual review, we compare human-reviewed transliterations with those produced by the iJunoon system as shown in Fig. 1. The comparison is drawn based on Word Error Rates (WER) and Character Error Rates (CER) comment-by-comment. We plot Word Error Rates in Fig. 1a, illustrating the proportion of words that were incorrectly transliterated by the iJunoon system compared to human reviewers. The Character Error Rates are shown in Fig. 1b. CER provides a more granular view, as it accounts for individual character-level discrepancies between transliterations produced by iJunoon system and human-reviewed transliterations.

To illustrate the contrast between human-reviewed and iJunoon-generated transliterations, we created bins to categorize instances based on WER and CER values. In particular, a bin of 0.5 signifies that it contains all cases where the WER or CER is greater than 0.4 and less than or equal to 0.5.

It can be seen from Fig. 1 that 15,312 out of 72,771 instances, which is approximately 21.04% of the total corpus, exhibit a perfect transliteration alignment between humans and the iJunonn system, resulting in WERs and CERs of 0.0. Conversely, around 78.96% of the transliterations produced by the iJunoon system required manual corrections from human reviewers to attain acceptable transliteration levels.

With the corrected Roman Urdu to Urdu transliterations, we expand the existing RUT corpus by adding the Urdu equivalent transliterations. This expansion results in the creation of a novel dataset, referred to as the PURUTT corpus, a comprehensive resource containing both Roman Urdu and Urdu comments labeled as either toxic or non-toxic. Similar to the RUT corpus, the PURUTT corpus boasts a substantial number of samples, with 13,097 toxic comments and 59,674 non-toxic comments. A detailed statistical breakdown of the PURUTT corpus, encompassing both Roman Urdu and Urdu segments, is presented in Table 2.

#### 1) COMPARISON WITH OTHER URDU TOXIC COMMENT CORPORA

To the best of our knowledge, other corpora available in the literature for the identification of toxic comments in Urdu are not substantially large in size. A comprehensive comparison of our created Urdu corpus with other notable corpora is presented in Table 3, which clearly highlights that the PURUTT corpus significantly surpasses existing corpora in terms of dataset size.

#### 2) COMPARISON WITH OTHER ROMAN URDU AND URDU TRANSLITERATION CORPORA

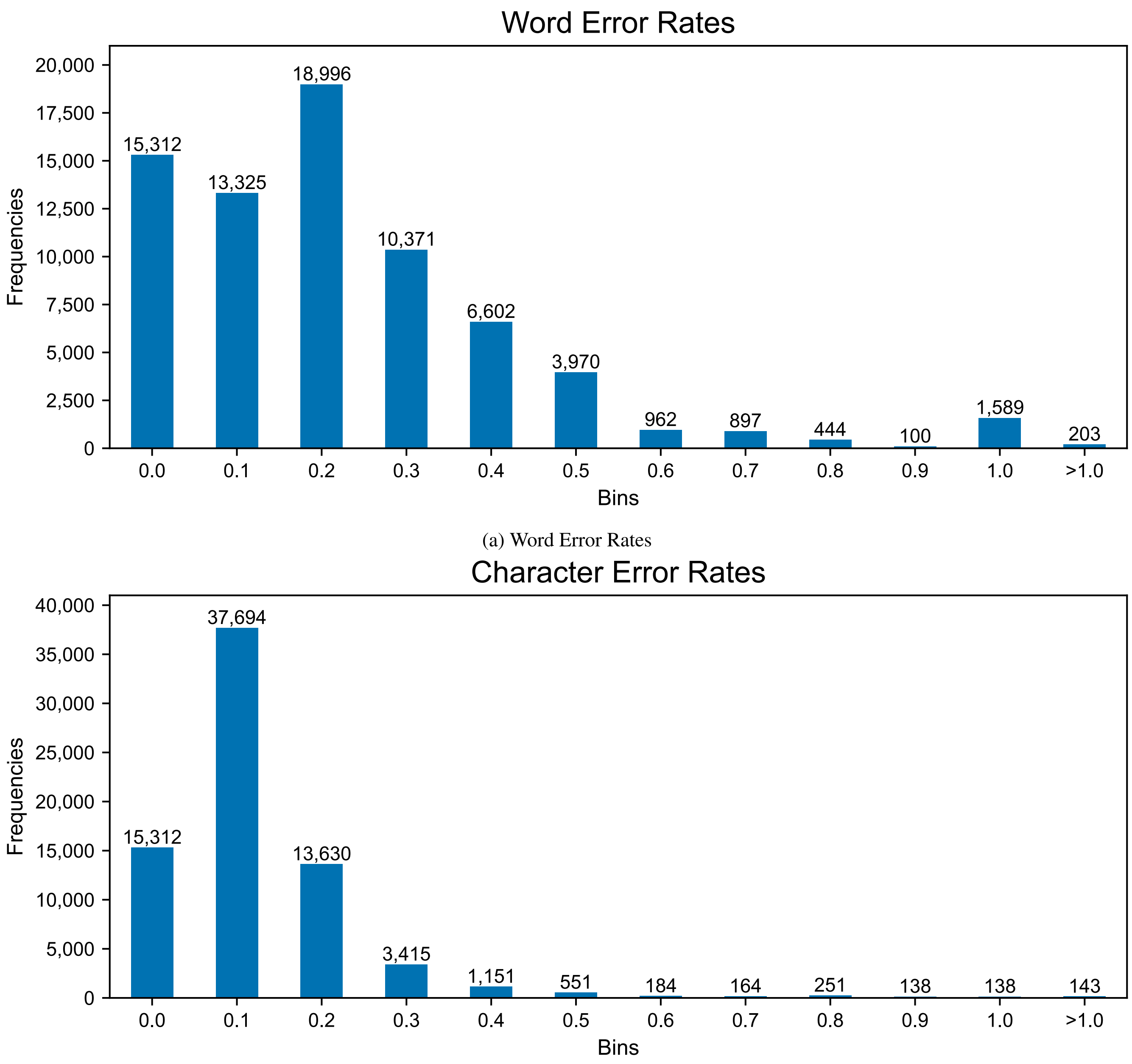
While several corpora exist for various Urdu language processing tasks, a dedicated resource focusing on parallel Roman Urdu and Urdu text for bidirectional transliteration remains limited. To the best of our knowledge, Roman-Urdu-Parl, developed in citealam2022roman, is the only publicly available corpus offering such parallel data. Notably, the corpora available in [59] and [60] are the older versions of Roman-Urdu-Parl corpus.

Roman-Urdu-Parl is a large corpus of 6.37 million parallel instances, generated primarily through the iJunoon transliteration system. The authors collected Urdu and Roman Urdu comments separately from several online sources and generated the per contra transliterations from iJunoon. However, as demonstrated in Table 1 and Fig. 1, iJunoon-generated output exhibits inaccuracies.

In contrast, PURUTT presents a meticulously curated corpus of 72,771 parallel instances, each meticulously reviewed by native Urdu speakers to ensure accuracy and linguistic fidelity. As shown in Table 4, PURUTT encompasses a significantly richer vocabulary than Roman-Urdu-Parl, with approximately 91 thousand unique terms in Roman Urdu and 51 thousand in standard Urdu.

## IV. URDU TOXIC COMMENT CLASSIFICATION

This section outlines the classification pipeline and employed models for Urdu toxic comment classification using the



(a) Word Error Rates

(b) Character Error Rates

**FIGURE 1.** **Comparison of iJunoon with human-reviewed transliterations in terms of word and character error rates.**

**TABLE 2.** **Statistics of PURUTT corpus.**

| Classes | Toxic | | Non-Toxic | | Total | |
|---|---|---|---|---|---|---|
| Scripts | Roman Urdu | Urdu | Roman Urdu | Urdu | Roman Urdu | Urdu |
| Comments | 13097 | 13097 | 59674 | 59674 | 72771 | 72771 |
| Maximum Length | 139 | 141 | 194 | 199 | 194 | 199 |
| Average Length | 13.82±14.96 | 14.01±15.11 | 20.73±18.85 | 20.80±19.01 | 19.49±18.40 | 19.58±18.55 |
| Unique Words | 25366 | 12454 | 78775 | 46958 | 91244 | 51347 |
| Total Words | 181086 | 183595 | 1237494 | 1241582 | 1418580 | 1425177 |

constructed Urdu corpus. We frame this problem as a binary text classification task with the goal of differentiating between toxic and non-toxic comments.

Our Urdu dataset is derived from the transliteration of the RUT corpus; hence, we leverage all models trained on the RUT corpus for comparison purposes. Additionally,

**TABLE 3. Size of PURUTT and other urdu toxic corpora.**

| # | Year | Ref | Corpus | Dataset Size |
|---|---|---|---|---|
| 1 | 2017 | [47] | Mustafa et al. | 8,000 |
| 2 | 2020 | [11] | Haq et al. | 6,420 |
| 3 | 2020 | [12] | Akhter et al. | 2,171 |
| 4 | 2021 | [48] | Ali et al. | 16,000 |
| 5 | 2021 | [49] | HSOC[a] Task A[1] | 3,500 |
| 6 | 2021 | [49] | HSOC[a] Task B[2] | 9,950 |
| 7 | 2021 | [21] | Amjad et al. | 3,564 |
| 8 | 2021 | [50] | Akram and Shahzad | 3,297 |
| 9 | 2022 | [51] | Ali et al. | 10,526 |
| 10 | 2022 | [52] | Khan and Qureshi | 7,625 |
| 11 | 2022 | [53] | Hussain et al. | 7,500 |
| 12 | 2023 | [54] | Akram et al. | 21,759 |
| 13 | 2023 | [55] | Arshad et al. | 7,800 |
| 14 | 2023 | [56] | Malik et al. | 2,400 |
| 15 | 2024 | [16] | Adeeba et al. | 12,428 |
| 16 | 2024 | [17] | Khan et al. | 12,082 |
| 17 | 2024 | [15] | Malik et al. | 9,771 |
| 18 | 2024 | [57] | Razi and Ejaz[b] | 8,880 |
| 19 | 2024 | [58] | Khan et al. | 4,808 |
| 20 | 2025 | **Proposed** | **PURUTT** | **72,771** |

[a] Abusive and Threatening Language Detection Competition in Urdu https://www.urduthreat2021.cicling.org/
[1] Urdu Abusive Language Detection dataset available at https://github.com/MaazAmjad/Urdu-abusive-detection-FIRE2021
[2] Urdu Threatening Language Detection dataset available at https://github.com/MaazAmjad/Urdu-threat-detection-FIRE2021
[b] Mixed Urdu, Roman Urdu, and English conversations

**TABLE 4. PURUTT and other transliteration corpora.**

| | Roman-Urdu-Parl | PURUTT |
|---|---|---|
| Parallel Sentence Pairs | **6.37 Million** | 72.7 Thousand |
| Roman Urdu Vocabulary | 42,927 | **91,244** |
| Urdu Vocabulary | 43,786 | **51,347** |

we finetune five multilingual large language models to address the Urdu toxic comment classification challenge more effectively. Fine-tuning of these LLMs is achieved by adding a fully-connected layer preceding the final output layer, allowing the model to adapt to the specific nuances of Urdu toxic language.

### A. URDU PRE-PROCESSING

In the preprocessing phase for the Urdu comments, we implemented a series of transformations to prepare the text for toxic comment classification in Urdu. These steps include the removal of: a) non-printable characters, b) extra white-spaces, c) English and Urdu punctuation marks, d) URLs, e) emoticons, f) English words, g) English and Urdu digits, and h) all Arabic and Urdu diacritics which include fathah, kasrah, dammah, tanwin al fathah, tanwin al kasrah, tanwin al dammah, sukun, shaddah, kharha fathah, kharha kasrah, inverted dammah, takhalus sign, verse sign, year sign, maddah, and high maddah.

Furthermore, to create a more consistent and standardized representation of Urdu text, we replaced Arabic Kaaf (ك) with Urdu Kaaf (ک), Arabic Yaa (ي) with Urdu Yaa (ی), and two-character Alif Mad (آ) with single-character Alif Mad (آ).

### B. COMMENT ENCODING

Text data cannot be given directly as input to the classification models, a numerical representation is needed. Classical numerical encoding techniques include bag-of-words representations like Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF.IDF), etc. The text input to machine learning models used in this study is given as TF.IDF weights such that each comment becomes a $V$-dimensional vector where $V$ is the size of the vocabulary.

The input to deep learning models used in this paper is given as Word vectors. Thus, each comment becomes a matrix $M \in \mathbb{R}^{W \times E}$, where $W$ is the maximum allowed length (number of words) of the comment and $E$ is the size of the embedding vector for each word in the corpus. Comments longer than $W$ are truncated whereas comments shorter than $W$ are padded with a special reserved value. For word vectors, we use pre-trained Urdu Word2Vec [61], FastText[10] and compare these with task-tailored word embeddings. By task-tailored word embeddings, we mean that classification models learn the word embeddings as part of the training process.

For large language models (LLMs), comments are encoded using their inherent contextual embeddings generated during both training and/or inference time.

### C. CLASSIFICATION MODELS

The classical machine learning models used in this paper are Naïve Bayes, Logistic Regression, Random Forests, and Support Vector Machines. We now describe the architectural details of the rest of the deep learning and large language models.

#### 1) CNN-GEORGE

The CNN-George model used in [62] has been successfully applied to classify binary toxic comments in English.

This architecture features an input embedding layer followed by three parallel convolutional blocks. Each block employs fixed filter widths equal to the dimension of the word vectors. The filter heights within each block are set to 3, 4, and 5, respectively, allowing for the capture of local contextual patterns at varying granularities. A max-over-time pooling layer is applied after the convolutional layer in each block, reducing dimensionality while retaining salient information. The outputs from all three convolutional blocks are concatenated and fed into a fully connected layer. The training of the model in the original paper was executed using Stochastic Gradient Descent (SGD) with a learning rate of $5e^{-3}$.

#### 2) CNN_GRAM

The CNN_Gram architecture proposed in [63], is specifically designed for the classification of toxic comments in Roman Urdu. This model leverages a hierarchical approach to capture

[10] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.ur.300.vec.gz

contextual information through the utilization of n-gram features.

The network's first layer consists of an embedding layer that serves as the input to a 1D convolutional neural network (CNN) with a kernel size of 1, followed by max-pooling and average-pooling layers. This results in the generation of unigram-based feature maps, which capture local patterns in the text data. The output from the max-pooling layer is then connected to a second CNN layer with a kernel size of 2, accompanied by max-pooling and average-pooling layers, which generates bigram-based feature maps. This process is repeated to obtain representations for trigrams and quadgrams, yielding four sets of max-pooling and average-pooling outputs, which are concatenated together. To capture higher-order dependencies, a global average-pooling and global max-pooling layer are applied in parallel across these concatenated features.

The resulting output features are then fed into two sequential blocks of fully connected layers, dropout layers, and batch normalization layers. This allows for the exploration of non-linear relationships within the feature space. The output from these two sequential blocks is then fed into the final output layer.

### 3) CNN_RUT

Saeed et al. report CNN_RUT architecture as tweaked Convolutional Neural Network (CNN) in [5] for Roman Urdu toxic comment classification. The network's first layer comprises an embedding layer with dimensions 200 × 300, which serves as the input to a 1D spatial dropout layer. This 1D spatial dropout layer is followed by five parallel convolutional blocks, each consisting of a convolutional layer and a global max-pooling layer. Notably, the convolutional layers in each block feature 32 filters with a fixed width of 300, and their heights vary incrementally from 1 to 5. The output of each convolutional block yields a 32-dimensional feature vector due to the fixed number of filters, which is then concatenated using a concatenation layer to form a 1 × 160 dimensional vector.

To mitigate the effects of overfitting and improve the model's interpretability, a dropout layer is added after the concatenation layer. Subsequently, three fully connected layers with 100, 50, and 1 units are sequentially attached, utilizing the binary cross-entropy loss function for optimization. This architecture effectively leverages the spatial hierarchies present in the text data to capture subtle patterns and relationships.

### 4) CNN+GRU

CNN+GRU taken from [64] is a Convolution-GRU-based deep model. The network's first layer comprises an input embedding layer with dimensions 100 × 300, which serves as the input to a dropout layer with a dropout rate of 20%.

The next layer is a 1D convolutional layer, equipped with ReLU activation function, which exhibits robustness to non-linearities. Notably, this layer features 100 filters with a spatial size of 4, and is followed by a max-pooling layer with a pool size of 4, which reduces the dimensionality of the feature maps while retaining essential information. Subsequently, the output of the convolutional layer is fed into a GRU (Gated Recurrent Unit) layer with 100 units, which enables the model to capture temporal dependencies. This GRU layer is further attached with a global max-pooling layer.

To introduce additional structure and reduce overfitting, a fully connected layer with softmax activation function and elastic-net regularization ($L_1$ and $L_2$ norms) is appended. This layer optimizes the model's parameters using Adam optimizer and categorical cross-entropy loss function.

### 5) MCBIGRU

The Multichannel Convolutional Bidirectional Gated Recurrent Unit (MCBiGRU) was introduced in [34] as a novel architecture for the classification of toxic comments in the English language in a multi-label setting.

The MCBiGRU model comprises 5 parallel channels, each consisting of an embedding layer, a 1D convolutional layer, a dropout layer, a 1D max-pooling layer, a bidirectional GRU layer, and another dropout layer, arranged sequentially. The output from each channel is concatenated using a concatenation layer, which is subsequently connected to a fully connected layer, a batch normalization layer, and an output layer with 6 neurons to accommodate the multi-label classification task. Notably, our implementation deviates from this original formulation by utilizing a single neuron in the final layer due to the binary nature of our classification problem.

### 6) BGRU-P

BGRU-P is the best-reported model by [65] for detecting toxic comments in English in a multi-label setting. The architecture takes a 100 × 300 dimensional input matrix and processes it through an embedding layer before applying a 1D spatial dropout layer with a dropout rate of 40%. This is followed by two parallel Bidirectional Gated Recurrent Unit (BGRU) layers, which are connected to the 1D spatial dropout layer. Notably, the first BGRU layer comprises 128 units, while the second BGRU layer features 64 units.

The model employs a focal loss function with fixed parameters $\alpha = 0.25$ and $\gamma = 5.0$. The output from both BGRU layers is concatenated using a concatenation layer, which is subsequently connected to global max-pooling and global average-pooling layers. Both pooling layers are concatenated and followed by a dropout layer with a dropout rate of 10%. This dropout layer is then connected to three fully connected layers, with sequential neurons of 100, 50, and 6. The last layer containing 6 neurons is due to the multi-labeled (6-labeled) classification setting in the original paper. However, due to the binary nature of the classification problem addressed in this research, we make the final layer consist of a single neuron.

#### 7) BGRU AND BLSTM

We also investigate the performance of two popular recurrent neural network (RNN) architectures: Bidirectional Gated Recurrent Unit (BGRU) and Bidirectional Long-Short Term Memory (BLSTM). Both models are reported to be effective for binary toxic comment classification tasks, as demonstrated in [5]. The primary difference between the two lies in the choice of recurrent unit cells: BGRU employs LSTM cells, while BLSTM utilizes GRU cells.

The input to both architectures is a 100 × 300 dimensional matrix, which serves as the embedding layer. This layer is followed by a 1D spatial dropout layer, which helps prevent overfitting and enhances model robustness. The output of this layer is then fed into the recurrent layer, where BGRU and BLSTM are implemented, respectively.

The recurrent layer's output is subsequently passed through another dropout layer, which further reduces overfitting and improves generalization performance. This layer is then connected to a fully connected layer with 50 neurons, where the model learns complex patterns in the data. Finally, a single neuron output layer with a sigmoid activation function is attached, enabling the model to produce binary class probabilities.

#### 8) MULTILINGUAL BERT (M-BERT)

Multilingual BERT (m-BERT) proposed by [66] is a pre-trained 12-layer transformer trained on Wikipedia pages of 104 languages. The m-BERT language model has a shared vocabulary of all 104 languages without any marker to represent the input language. WordPiece Tokenizer, a well-known subword-based tokenization method, is used to provide input to m-BERT. The version of m-BERT in this paper is bert-base-multilingual-cased. Embeddings from m-BERT are generated in a 768-dimensional feature vector.

#### 9) DISTILMBERT

DistilmBERT proposed by [67] is a smaller, faster, and distilled version of m-BERT retaining 97% of the language understanding capabilities. DistilmBERT tokenizer is identical to WordPiece m-BERT tokenizer. The version of DistilmBERT in this paper is bert-base-multilingual-cased. Similar to m-BERT, embeddings from DistilmBERT are 768-dimensional.

#### 10) M-T5

A massively multilingual pre-trained text-to-text transformer (m-T5) proposed by [68] is trained on the Common Crawl dataset for 101 languages. It is a transformer-based encoder-decoder architecture. The version of m-T5 in this paper is mt5-base. Embeddings from m-T5 are generated in a 768-dimensional feature vector.

#### 11) XLMROBERTA (XLM-R)

XLMRoBERTa (XLM-R) proposed by [69] uses a Sentence Piece based tokenizer and is a multilingual language model based on Facebook's 2019-released RoBERTa model. It is a large multilingual language model that was trained on 2.5 terabytes of Common Crawl data in 100 distinct languages. It uses SentencePiece as its tokenizer. The version of XLM-R in this paper is xlm-roberta-base. Embeddings from XLM-R are generated in a 768-dimensional feature vector.

#### 12) MURIL

Multilingual Representations for Indian Languages (MuRIL) proposed by [70] is a BERT-based model that supports 17 Indian languages including Urdu. It was trained using CommonCrawl OSCAR and Wikipedia corpus by augmenting the data with translation and transliteration of the document pairs. The tokenization technique used for MuRIL is the WordPiece algorithm trained from scratch on the gathered corpora. The output embeddings of MuRIL are projected into a 768-dimensional feature space.

## V. EXPERIMENTS AND RESULTS

This section details the experimental setup and performance evaluation of various machine learning models for Urdu toxic comment classification. We leveraged the Scikit-learn library for training machine learning models, including Naïve Bayes, Logistic Regression, Random Forests, and Support Vector Machines. For deep learning architectures, we employed Keras[11] with the TensorFlow backend to facilitate efficient training using the GPU support. Furthermore, for fine-tuning pre-trained large language models, we utilized Hugging Face.[12]

Our experiments were executed on an NVIDIA 2080Ti GPU featuring 11 GB of dedicated GPU memory with running Ubuntu 18.04 as the operating system. Additionally, the dataset and source code repository are publicly accessible,[13] allowing researchers to replicate our findings and contribute to the development of more robust toxic comment classification models.

### A. EVALUATION

Model performance was assessed via stratified 5-fold cross-validation. The Urdu corpus was partitioned into five strata, ensuring each fold maintained almost the same proportion of toxic and non-toxic comments. In each iteration, one partition served as the test set while the remaining four were utilized for training. Model hyperparameters were optimized for F1-score to facilitate a fair comparative analysis across models. To ensure reproducibility, the seed point for the stratified 5-fold data split was fixed at zero.

[11] https://keras.io/
[12] https://huggingface.co/
[13] https://github.com/hafizhassaan/Urdu-Toxic-Comments

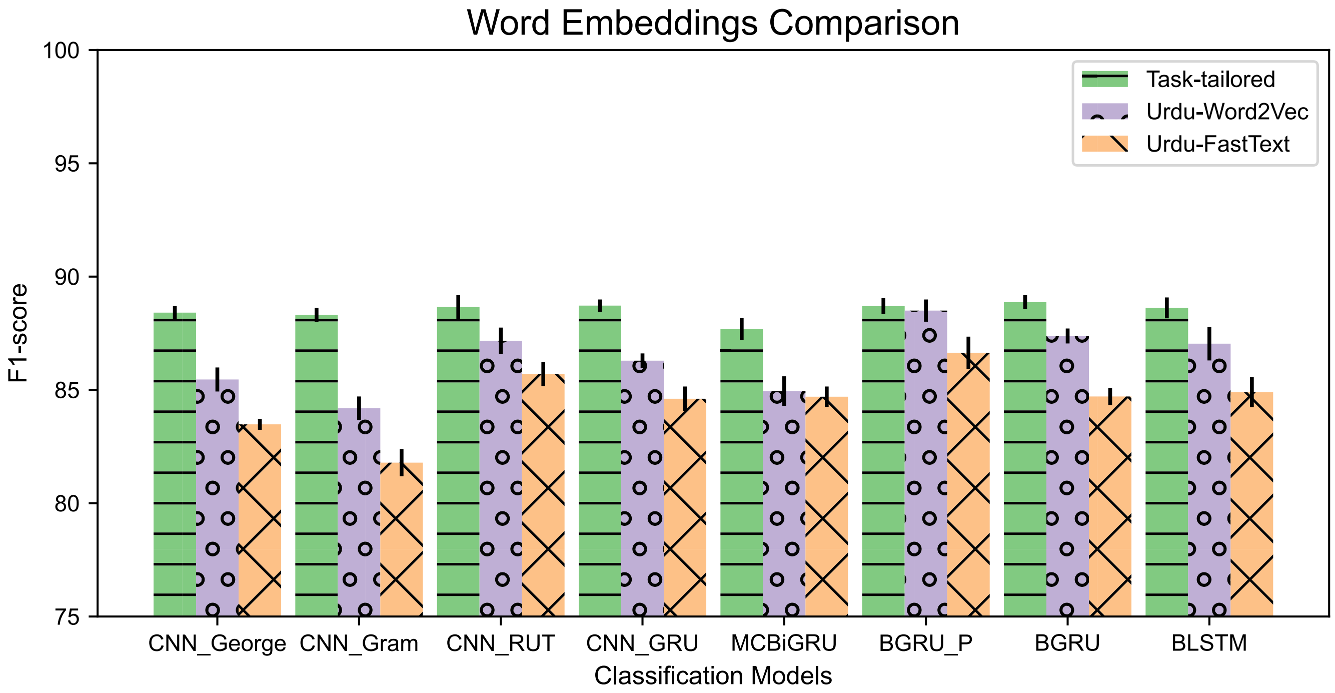


**FIGURE 2.** Analysis of word embeddings in deep learning models (Best viewed in color).

We evaluate the classification models using a suite of metrics, including average accuracy, average precision, average recall, and average F1 score across all folds. Given the significant class imbalance in our dataset, F1 serves as the primary evaluation metric in this paper.

### B. HYPER-PARAMETER TUNING

To optimize classification performance for toxic comment detection in Urdu, we conduct extensive experiments on 33 distinct learning models. Each model's hyper-parameters are carefully curated from their respective parameter spaces, ensuring that the optimal configuration is identified. To determine the maximum sequence length input to our models, we employ a data-driven approach based on the $\mu + 3\sigma$ rule, which provides a robust estimate of the underlying distribution.

Our tuning process involves iteratively exploring each model's hyper-parameters while taking out one validation set separate from the training set in each of the five stratified folds. We adopt manual tuning approach to hyper-parameter optimization, focusing on key hyper-parameters that significantly impact model performance.

For logistic regression, we investigate the inverse of regularization strength (C), penalty type, and solver to identify optimal configurations. For random forests, we systematically adjust the number of estimators, tree depth, and criterion to achieve improved F1 scores. For support vector machines, we perform a thorough exploration of hyper-parameters including C, penalty coefficient, kernel choice, and gamma value to determine the most effective settings.

For deep learning and large language models, we conduct an exhaustive search across various hyper-parameters such as weight initialization schemes, activation functions, dropout rates, learning rate decay, optimizers, and early stopping strategies to identify the best-performing configurations. Through this meticulous tuning process, we aim to maximize validation F1 scores, ensuring that our classification models are optimized for accurate toxic comment classification in Urdu.

### C. MODEL RESULTS

The performance of the deep learning models outlined in Section IV-C was evaluated using three distinct word embedding strategies: (i) Urdu pre-trained FastText, (ii) Urdu pre-trained Word2Vec, and (iii) Task-tailored word embeddings learned during training. To provide comprehensive insights into model performance, we present a detailed analysis in Fig. 2, which visually depicts the F1-score trends across these embedding types for each deep learning architecture.

Our results indicate that deep models trained with task-tailored word embeddings exhibit substantial improvements in F1 scores compared to those utilizing pre-trained Urdu Word2Vec or pre-trained Urdu FastText. Furthermore, Figure 2 demonstrates that models leveraging pre-trained Urdu Word2Vec embeddings generally outperformed those employing pre-trained Urdu FastText embeddings. Table 5 provides a detailed breakdown of the accuracy, precision, and recall scores for each model and embedding type.

**TABLE 5.** Evaluation of individual classification models (average of 5 folds).

| | Models | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| ML with TFIDF | NB | 94.56 ± 0.2 | 88.83 ± 1.1 | 79.83 ± 0.4 | 84.09 ± 0.6 |
| | LR | 95.49 ± 0.2 | 88.41 ± 0.8 | 86.26 ± 0.3 | 87.32 ± 0.4 |
| | RF | 94.53 ± 0.1 | 88.17 ± 1.7 | 80.51 ± 1.9 | 84.13 ± 0.2 |
| | SVM | 95.85 ± 0.1 | 88.74 ± 0.9 | 88.17 ± 1.1 | 88.44 ± 0.3 |
| DL with Task-tailored Word Embeddings | CNN_George | 95.93 ± 0.1 | 90.75 ± 1.1 | 86.20 ± 1.0 | 88.41 ± 0.3 |
| | CNN_Gram | 95.90 ± 0.1 | 90.82 ± 1.8 | 86.00 ± 1.9 | 88.31 ± 0.3 |
| | CNN_RUT | 96.06 ± 0.2 | 91.95 ± 0.7 | 85.61 ± 1.1 | 88.66 ± 0.5 |
| | CNN_GRU | 96.06 ± 0.1 | 91.54 ± 0.7 | 86.08 ± 0.6 | 88.72 ± 0.3 |
| | MCBiGRU | 95.76 ± 0.2 | 91.74 ± 1.0 | 84.00 ± 0.9 | 87.69 ± 0.5 |
| | BGRU_P | 96.09 ± 0.1 | 92.55 ± 0.6 | 85.15 ± 0.4 | 88.70 ± 0.4 |
| | BGRU | 96.08 ± 0.2 | 90.82 ± 1.7 | 87.04 ± 1.1 | 88.87 ± 0.3 |
| | BLSTM | 96.05 ± 0.2 | 92.04 ± 0.9 | 85.45 ± 0.9 | 88.62 ± 0.5 |
| DL with Pre-trained Urdu Word2Vec | CNN_George | 95.00 ± 0.2 | 89.60 ± 0.6 | 81.71 ± 1.1 | 85.46 ± 0.5 |
| | CNN_Gram | 94.53 ± 0.2 | 87.70 ± 1.1 | 80.98 ± 1.4 | 84.19 ± 0.5 |
| | CNN_RUT | 95.59 ± 0.2 | 91.57 ± 0.3 | 83.17 ± 1.0 | 87.17 ± 0.6 |
| | CNN_GRU | 95.22 ± 0.1 | 89.28 ± 1.4 | 83.52 ± 1.3 | 86.29 ± 0.3 |
| | MCBiGRU | 94.88 ± 0.2 | 90.16 ± 1.2 | 80.35 ± 1.9 | 84.95 ± 0.7 |
| | BGRU_P | 96.01 ± 0.1 | 91.86 ± 0.8 | 85.40 ± 1.3 | 88.50 ± 0.5 |
| | BGRU | 95.65 ± 0.1 | 91.42 ± 1.3 | 83.71 ± 1.6 | 87.38 ± 0.3 |
| | BLSTM | 95.49 ± 0.3 | 90.17 ± 1.5 | 84.14 ± 1.1 | 87.04 ± 0.7 |
| DL with Pre-trained Urdu FastText | CNN_George | 94.29 ± 0.2 | 87.23 ± 2.6 | 80.16 ± 2.0 | 83.48 ± 0.2 |
| | CNN_Gram | 93.73 ± 0.2 | 85.76 ± 1.8 | 78.25 ± 2.0 | 81.79 ± 0.6 |
| | CNN_RUT | 95.03 ± 0.2 | 88.98 ± 1.0 | 82.67 ± 1.4 | 85.70 ± 0.5 |
| | CNN_GRU | 94.65 ± 0.1 | 87.76 ± 0.9 | 81.71 ± 1.7 | 84.61 ± 0.5 |
| | MCBiGRU | 94.76 ± 0.1 | 89.17 ± 0.8 | 80.68 ± 1.2 | 84.70 ± 0.5 |
| | BGRU_P | 95.36 ± 0.2 | 89.91 ± 0.9 | 83.64 ± 1.9 | 86.64 ± 0.7 |
| | BGRU | 94.65 ± 0.1 | 87.30 ± 1.2 | 82.31 ± 1.3 | 84.71 ± 0.4 |
| | BLSTM | 94.74 ± 0.2 | 87.74 ± 0.4 | 82.25 ± 1.1 | 84.90 ± 0.7 |
| Finetuned Large Language Models | m-BERT | 96.43 ± 0.1 | 92.28 ± 1.2 | 87.53 ± 1.0 | 89.83 ± 0.3 |
| | DistilmBERT | 96.21 ± 0.1 | 92.47 ± 1.1 | 85.99 ± 1.9 | 89.09 ± 0.6 |
| | m-T5 | 96.37 ± 0.2 | 91.59 ± 0.8 | 87.94 ± 0.9 | 89.72 ± 0.5 |
| | XLM-R | 96.52 ± 0.2 | **92.69 ± 1.0** | 87.61 ± 1.0 | 90.07 ± 0.5 |
| | MuRIL | **96.63 ± 0.1** | 91.78 ± 0.8 | **89.28 ± 1.7** | **90.50 ± 0.5** |

Detailed performance metrics, including accuracy, precision, and recall scores, are presented in Table 5.

Among the deep learning architectures evaluated, the BGRU-P model demonstrated superior F1 scores across all three types of word embeddings, consistently achieving higher F1 scores than other models except for BGRU with task-tailored word embeddings.

The performance of individual classification models on Urdu toxic comment classification is presented in Table 5. This table reports the average accuracy, precision, recall, and F1-score for each classification model evaluated.

Among traditional machine learning approaches utilizing TF-IDF features and deep learning models employing word embeddings, BGRU_P with task-tailored word embeddings exhibits the highest accuracy of 96.09% and the highest precision of 92.55%. Conversely, SVM with TF-IDF achieves the highest recall of 88.17%, while BGRU with task-tailored word embeddings yields the highest F1-score of 88.87%.

Our findings demonstrate that fine-tuning pre-trained multilingual large language models (LLMs) outperforms traditional machine learning algorithms and deep learning architectures, including CNNs, RNNs, their variants, or their combinations. Notably, fine-tuned MuRIL achieved the highest F1-score of 90.50%, indicating its effectiveness in capturing nuanced toxic language patterns within Urdu text.

Further analysis reveals that XLM-R achieves the highest precision (92.69%). Meanwhile, MuRIL demonstrates a strong balance across all metrics, achieving both high accuracy (96.63%) and recall (89.28%). These results demonstrate the effectiveness of fine-tuning large language models for Urdu toxic comment classification.

### D. ENSEMBLE MODEL

Ensemble methods have demonstrated their effectiveness in enhancing performance across a range of learning-based tasks. The fundamental principle behind ensembles lies in aggregating predictions from multiple individual (base) models, often termed "weak learners", to construct a more robust and accurate predictive model. This paper explores the efficacy of ensemble techniques for Urdu toxic comment classification by combining previously learned classification models into ten distinct ensemble groups.

These groups primarily encompass machine learning (ML), deep learning (DL), best deep learning (BDL) models incorporating task-tailored word embeddings, and fine-tuned multilingual large language models (LLM). The ten ensemble groups are:

1) Ensemble group of machine learning (ML) models
2) Ensemble group of deep learning (DL) models
3) Ensemble group of best deep learning (BDL) models i.e., models with task-tailored word embeddings

**TABLE 6.** Performance analysis - ensemble combinations (average of 5 folds).

| # | Models | Technique | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| 1 | ML | AP | 96.01 ± 0.1 | 90.98 ± 0.3 | 86.42 ± 0.4 | 88.64 ± 0.3 |
| | | MV | 95.91 ± 0.1 | 88.78 ± 0.5 | 88.45 ± 0.6 | 88.61 ± 0.2 |
| 2 | DL | AP | 96.34 ± 0.1 | 94.97 ± 0.3 | 84.12 ± 0.5 | 89.22 ± 0.3 |
| | | MV | 96.42 ± 0.1 | 93.69 ± 0.5 | 85.92 ± 0.7 | 89.63 ± 0.3 |
| 3 | BDL | AP | 96.37 ± 0.1 | 94.16 ± 0.6 | 85.15 ± 0.7 | 89.42 ± 0.3 |
| | | MV | 96.43 ± 0.1 | 92.59 ± 0.7 | 87.14 ± 0.7 | 89.78 ± 0.4 |
| 4 | LLM | AP | 97.00 ± 0.1 | 93.53 ± 0.4 | **89.55 ± 0.7** | 91.50 ± 0.4 |
| | | MV | 96.94 ± 0.1 | 94.34 ± 0.5 | 88.31 ± 1.1 | 91.22 ± 0.4 |
| 5 | ML + DL | AP | 96.43 ± 0.1 | 94.96 ± 0.4 | 84.65 ± 0.5 | 89.51 ± 0.3 |
| | | MV | 96.50 ± 0.1 | 93.84 ± 0.5 | 86.20 ± 0.6 | 89.85 ± 0.3 |
| 6 | ML + BDL | AP | 96.43 ± 0.1 | 94.00 ± 0.5 | 85.66 ± 0.6 | 89.63 ± 0.3 |
| | | MV | 96.48 ± 0.2 | 92.85 ± 0.7 | 87.13 ± 0.6 | 89.90 ± 0.5 |
| 7 | ML + LLM | AP | **97.07 ± 0.2** | 94.11 ± 0.2 | 89.33 ± 0.8 | **91.65 ± 0.5** |
| | | MV | 96.97 ± 0.1 | 94.58 ± 0.3 | 88.20 ± 1.0 | 91.28 ± 0.5 |
| 8 | DL + LLM | AP | 96.63 ± 0.1 | **95.24 ± 0.3** | 85.58 ± 0.6 | 90.15 ± 0.4 |
| | | MV | 96.61 ± 0.1 | 94.49 ± 0.3 | 86.18 ± 0.7 | 90.14 ± 0.4 |
| 9 | BDL + LLM | AP | 96.88 ± 0.2 | 95.00 ± 0.3 | 87.26 ± 0.7 | 90.97 ± 0.5 |
| | | MV | 96.77 ± 0.2 | 94.54 ± 0.5 | 87.10 ± 0.9 | 90.66 ± 0.5 |
| 10 | All | AP | 96.66 ± 0.1 | 95.12 ± 0.3 | 85.82 ± 0.5 | 90.23 ± 0.4 |
| | | MV | 96.66 ± 0.1 | 94.53 ± 0.3 | 86.42 ± 0.7 | 90.29 ± 0.4 |

4) Ensemble group of fine-tuned multilingual large language models (LLM) classifiers
5) Ensemble group of ML and DL
6) Ensemble group of ML and BDL
7) Ensemble group of ML and LLM
8) Ensemble group of DL and LLM
9) Ensemble group of BDL and LLM
10) Ensemble group of all 33 classifiers (All)

For each of the above-mentioned ensemble groups, we implemented two aggregation strategies: averaging probability (AP) and majority voting (MV). Table 6 presents a comparative analysis of the performance metrics (average accuracy, average precision, average recall, and average F1-score) achieved by each ensemble group across all folds.

The baseline for comparison is MuRIL, which demonstrated the highest individual model performance with an accuracy of 96.63%, recall of 89.28%, and F1-score of 90.50%. However, the highest precision (92.69%) was achieved by XLM-R as shown in Table 5.

Analysis of Table 6 reveals that averaging probability (AP) generally achieves higher F1 scores than majority voting (MV) within ensemble groups comprised primarily of ML models (ML) or incorporating LLM models (LLM, ML+LLM, DL+LLM, and BDL+LLM), with the exception of the all (All) group. In terms of precision, while ML with AP, ML with MV, and BDL with MV exhibit lower scores compared to XLM-R, all other ensemble groups surpass XLM-R's precision.

Considering accuracy, ensemble groups incorporating LLM models (LLM, ML+LLM, DL+LLM, BDL+LLM, and All) outperform the baseline MuRIL model. Furthermore, only two groups, i.e., LLM with AP and ML + LLM with AP, achieve higher recall than the baseline MuRIL model. Finally, in terms of the F1-score, three groups (LLM, ML + LLM, and BDL + LLM) surpass the baseline MuRIL model's performance.

**TABLE 7.** Confusion matrix of the best ensemble set.

| Confusion Matrix | | Predicted | |
|---|---|---|---|
| | | Toxic | Non-Toxic |
| Actual | Toxic | 11699 | 1398 |
| | Non-Toxic | 732 | 58942 |

Overall, the highest precision of 95.24% is achieved by DL + LLM with AP, while the best recall score of 89.55% is obtained by LLM with AP. The highest accuracy of 97.07% and F1-score of 91.65% are attained by ML + LLM by averaging the probability scores of each base model in this group. Our findings emphasize the potential of ensemble learning strategies to significantly improve toxic comment classification performance for Urdu language texts.

Furthermore, the Friedman test was employed to assess the statistical significance of the classification models. This test was selected for two primary reasons: (i) the assumption of normality is not made for the distribution of F1 scores, and (ii) the data splits (5 folds) are repeated across all models.

The null hypothesis ($H_0$) posits that there is no significant difference in performance (F1-score) among the classification models. Conversely, the alternative hypothesis ($H_a$) suggests that at least two models exhibit significantly different performance levels.

The results of the statistical analysis indicate a test statistic of 89.6258, with a $p$-value less than 0.05. This finding demonstrates a statistically significant difference in the performance of the models.

### E. DISCUSSION AND LIMITATIONS

This section undertakes a thorough examination of the impact of out-of-embedding-vocabulary (OOEV) words/tokens on

**TABLE 8.** Demonstration of ensemble model predictions of urdu comments from PURUTT corpus.

| | # | Urdu Comments | Translations | Actual | Predicted |
|---|---|---|---|---|---|
| tp | 1 | میر النڈ چوس مادر چود کی بچی | Suck my dick motherfucker's daughter | 1 | 1 |
| tp | 2 | لا کھ دی لعنت ایسے شوز پہ | Damn these shows | 1 | 1 |
| tp | 3 | گانڈو بڈھا لعنت ہے تم پر | Damn you you old gay | 1 | 1 |
| tn | 4 | ہم نے بھی ابھی دیکھا ہے | We too have seen just now | 0 | 0 |
| tn | 5 | انکا بہت احترام کرنا آپ لوگ | Respect them a lot | 0 | 0 |
| tn | 6 | صدر مملکت نے بھی فلم دیکھی | The President also watched the film | 0 | 0 |
| fp | 7 | یہ ہے ماں کی ممتا | This is mother's love | 0 | 1 |
| fp | 8 | ڈاکٹر ذاکر نالائق نہیں | Dr. Zakir is not incompetent | 0 | 1 |
| fp | 9 | کسی پر لعنت کرنا بری بات ہے | Cursing someone is bad | 0 | 1 |
| fn | 10 | زبان سنبھال کے بات کرو | Mind/watch your language | 1 | 0 |
| fn | 11 | ختم کرو مسلمان | Finish the muslims | 1 | 0 |
| fn | 12 | بھاڑ میں جاتو | Go to hell | 1 | 0 |

the performance of deep learning models trained for Urdu toxic comment classification. We observe that approximately 47.11% of the corpus vocabulary falls outside the pre-trained Urdu Word2Vec embedding space and 39.23% of the corpus vocabulary is OOEV in case of the pre-trained Urdu FastText. This highlights a significant limitation of Urdu pre-trained embeddings, as a substantial portion of the Urdu language data remains uncaptured within the vocabularies of these embeddings.

In contrast, task-tailored word embeddings encompass all words incorporated into the embedding vocabulary, effectively mitigating the impact of OOEV words. Similarly, large language models (LLMs) do not exhibit any instances of OOEV in our analysis. This suggests that both task-specific embedding methods and LLMs handle the vocabularies better within the Urdu language context.

Table 7 presents the confusion matrix of our proposed ensemble model (ML+LLM), which combines the strengths of four traditional machine learning (ML) models and five large language models (LLM) as base classifiers for Urdu toxic comment classification. This granular view of the ensemble model's performance reveals that it successfully identified 11699 toxic comments (True Positives) and 58942 non-toxic comments accurately (True Negatives). However, the model also produced 732 instances where non-toxic comments were misclassified as toxic (False Positives) and 1398 toxic comments were incorrectly classified as non-toxic (False Negatives). The table further shows that, the ensemble model exhibits a misclassification rate of approximately 10.67% for the toxic class [(1398/13097) × 100] and a significantly lower rate of 1.23% for the non-toxic class [(732/59674) × 100]. This disparity is obviously because of the inherent class imbalance within the dataset, where the non-toxic class comprises approximately 82% of the corpus.

Furthermore, Table 8 provides a concrete illustration of the ensemble model's prediction capabilities, showcasing its performance on a set of sample Urdu comments. The table displays the actual labels alongside the predicted labels, providing valuable insights into the model's accuracy in terms of True Positives (tp), True Negatives (tn), False Positives (fp), and False Negatives (fn). Comments 1-3 exemplify tp cases, accurately identifying toxic language. Comments 4-5 represent tn instances, correctly classifying non-toxic expressions.

Conversely, comments 7-9 are classified as fp, indicating the model's tendency to misidentify non-toxic statements as toxic. This highlights a potential challenge in distinguishing nuanced expressions from genuine toxicity. For instance, "ماں کی ممتا" (mother's love) in comment 7 might be misinterpreted due to its contextual relatively stronger association with toxic class. "ذاکر نالائق" (Zakir the incompetent) in comment 8 triggers the model's bias towards negative sentiment. Similarly, "لعنت" (curse) in comment 9 possesses a direct association with negativity, which might not always capture genuine toxicity depending on the context.

The fn category, encompassing comments 10-12, reveals instances where the model fails to recognize toxic language. Analyzing these cases suggests potential limitations in the model's understanding of certain linguistic patterns. For example, the word "بھاڑ" (hell) appears seven times in the toxic class but is missed by the model. Similarly, "زبان سنبھال" (watch your language), appears three times in toxic class, and "ختم کر" (finish), occurring 17 times in toxic class, are not accurately captured as true negatives. These misclassifications highlight a need to further refine the model's ability to identify subtle forms of toxicity and contextual nuances within Urdu text.

## VI. CONCLUSION AND FUTURE DIRECTIONS

We address a critical problem of toxic comment detection in an under-resourced Urdu language. The detection of toxic comments in Urdu is challenging due to the scarcity and quality of language resources. In this paper, we manually transliterated an existing labeled Roman Urdu toxic comment corpus into Urdu to generate a large corpus for Urdu toxic comment classification. With this transliteration approach, we build one corpus for two Urdu language processing tasks, (a) classification of toxic comments in Urdu and Roman Urdu, and (b) transliteration of Roman Urdu to Urdu and vice versa. The developed corpus, the

PURUTT corpus, has 72,771 parallel Urdu and Roman Urdu comments labeled as either toxic or non-toxic. We limited the scope of this paper to the classification of toxic comments leaving the transliteration for future research.

For the classification of toxic comments in Urdu, we employed all models that were trained on the Roman Urdu corpus as well as a few additional models, including five multilingual large language models. We also compared pre-trained Urdu word embeddings with task-tailored word embeddings in deep learning models. The results achieved on the prepared corpus indicate that: a) learning task-tailored word embeddings outperforms available pre-trained Urdu word embeddings such as Word2Vec and FastText; b) Finetuning multilingual large language models achieve higher scores than word embeddings based deep learning models; c) MuRIL achieves the highest empirical scores among all of the individually trained classifiers; d) ensembling the individual classifiers improves the overall scores on the PURUTT corpus for toxic comment classification in Urdu, reaching an F1-score of 91.65%.

Moreover, this paper provides a detailed analysis of the ensemble's strengths and limitations, shedding light on its performance characteristics. The PURUTT corpus is made publicly available, with the expectation that it will inspire novel transliteration models from Roman Urdu to Urdu and vice versa. We also plan to investigate the impact of different LLM quantization techniques and various prompt engineering techniques on the performance of toxic comment classification models in Urdu.

## REFERENCES

[1] S.-H. Lee and H.-W. Kim, "Why people post benevolent and malicious comments online," *Commun. ACM*, vol. 58, no. 11, pp. 74–79, Oct. 2015.

[2] M. M. Khan, K. Shahzad, and M. K. Malik, "Hate speech detection in Roman Urdu," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 1, pp. 1–19, Jan. 2021.

[3] A. Atif, A. Zafar, M. Wasim, T. Waheed, A. Ali, H. Ali, and Z. Shah, "Cyberbullying detection and abuser profile identification on social media for Roman Urdu," *IEEE Access*, vol. 12, pp. 123339–123351, 2024.

[4] J. S. Vedeler, T. Olsen, and J. Eriksen, "Hate speech harms: A social justice discussion of disabled Norwegians' experiences," *Disability Soc.*, vol. 34, no. 3, pp. 368–383, Mar. 2019.

[5] H. H. Saeed, M. H. Ashraf, F. Kamiran, A. Karim, and T. Calders, "Roman Urdu toxic comment classification," *Lang. Resour. Eval.*, vol. 55, no. 4, pp. 971–996, Dec. 2021.

[6] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proc. 1st Workshop Abusive Lang. Online*, Z. Waseem, W. H. K. Chung, D. Hovy, and J. R. Tetreault, Eds., Vancouver, BC, Canada, 2017, pp. 85–90.

[7] K. A. Karthikraja, A. S. Kumar, B. Bharathi, J. Bhuvana, and T. T. Mirnalinee, "Abusive and threatening language detection in native Urdu script tweets exploring four conventional machine learning techniques and MLP," in *Proc. Forum Inf. Retr. Eval.*, vol. 3159, P. Mehta, T. Mandl, P. Majumder, and M. Mitra, Eds., Gandhinagar, India, Dec. 2021, pp. 806–812.

[8] Wikipedia. (2024). *Pakistan—Wikipedia, The Free Encyclopedia*. Accessed: Dec. 22, 2024. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Pakistan&oldid=1263247705

[9] M. K. Das, S. Banerjee, and P. Saha, "Abusive and threatening language detection in Urdu using boosting based and BERT based models: A comparative approach," in *Proc. Forum Inf. Retr. Eval.*, vol. 3159, P. Mehta, T. Mandl, P. Majumder, and M. Mitra, Eds., Gandhinagar, India, Dec. 2021, pp. 791–798.

[10] A. Daud, W. Khan, and D. Che, "Urdu language processing: A survey," *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 279–311, Mar. 2017.

[11] N. U. Haq, M. Ullah, R. Khan, A. Ahmad, A. Almogren, B. Hayat, and B. Shafi, "USAD: An intelligent system for slang and abusive text detection in PERSO-Arabic-Scripted Urdu," *Complexity*, vol. 2020, pp. 1–7, Nov. 2020.

[12] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, and M. T. Sadiq, "Automatic detection of offensive language for Urdu and Roman Urdu," *IEEE Access*, vol. 8, pp. 91213–91226, 2020.

[13] M. A. Syed, A. U. Rahman, and M. Khan, "Quantifying the use of English words in Urdu news-stories," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, M. Lan, Y. Wu, M. Dong, Y. Lu, and Y. Yang, Eds., Shanghai, China, Nov. 2019, pp. 1–6.

[14] M. Noor, B. Anwar, F. Muhabat, and B. Kazemian, "Code-switching in Urdu books of Punjab text book board, Lahore, Pakistan," *Commun. Linguistics Stud.*, vol. 1, no. 2, pp. 13–20, May 2015.

[15] M. S. I. Malik, A. Nawaz, and M. M. Jamjoom, "Hate speech and target community detection in nastaliq Urdu using transfer learning techniques," *IEEE Access*, vol. 12, pp. 116875–116890, 2024.

[16] F. Adeeba, M. I. Yousuf, I. Anwer, S. U. Tariq, A. Ashfaq, and M. Naqeeb, "Addressing cyberbullying in Urdu tweets: A comprehensive dataset and detection system," *PeerJ Comput. Sci.*, vol. 10, p. e1963, Apr. 2024.

[17] A. Khan, A. Ahmed, S. Jan, M. Bilal, and M. F. Zuhairi, "Abusive language detection in Urdu text: Leveraging deep learning and attention mechanism," *IEEE Access*, vol. 12, pp. 37418–37431, 2024.

[18] S. Saumya, A. Kumar, and J. P. Singh, "Filtering offensive language from multilingual social media contents: A deep learning approach," *Eng. Appl. Artif. Intell.*, vol. 133, Jul. 2024, Art. no. 108159.

[19] L. F. Benassou, S. Bendaouia, O. Salem, and A. Mehaoua, "Detecting virtual harassment in social media using machine learning," in *Proc. Mach. Learn. Netw.*, in Lecture Notes in Computer Science, vol. 14525, É. Renault, S. Boumerdassi, and P. Mühlethaler, Eds., Cham, Switzerland: Springer, 2024, pp. 185–198.

[20] M. Yi, M. Lim, H. Ko, and J. Shin, "Method of profanity detection using word embedding and LSTM," *Mobile Inf. Syst.*, vol. 2021, pp. 1–9, Feb. 2021.

[21] M. Amjad, N. Ashraf, A. Zhila, G. Sidorov, A. Zubiaga, and A. Gelbukh, "Threatening language detection and target identification in Urdu tweets," *IEEE Access*, vol. 9, pp. 128302–128313, 2021.

[22] K. B. Ozler, K. Kenski, S. Rains, Y. Shmargad, K. Coe, and S. Bethard, "Fine-tuning for multi-domain and multi-label uncivil language detection," in *Proc. 4th Workshop Online Abuse Harms*, S. Akiwowo, B. Vidgen, V. Prabhakaran, and Z. Waseem, Eds., Nov. 2020, pp. 28–33.

[23] J. Zhang, Q. Wu, Y. Xu, C. Cao, Z. Du, and K. Psounis, "Efficient toxic content detection by bootstrapping and distilling large language models," in *Proc. 38th Conf. Artif. Intell. (AAAI), 36th Conf. Innov. Appl. Artif. Intell. (IAAI), 14th Symp. Educ. Adv. Artif. Intell. (EAAI)*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds., Vancouver, BC, Canada, 2024, pp. 21779–21787.

[24] S. Mahbub, E. Pardede, and A. S. M. Kayes, "Detection of harassment type of cyberbullying: A dictionary of approach words and its impact," *Secur. Commun. Netw.*, vol. 2021, pp. 1–12, Jun. 2021.

[25] N. Zainuddin, A. Selamat, and R. Ibrahim, "Twitter hate aspect extraction using association analysis and dictionary-based approach," in *Proc. 16th Int. Conf.*, vol. 297, H. Fujita, A. Selamat, and S. Omatu, Eds., Kitakyushu City, Japan, Sep. 2017, pp. 641–651.

[26] J. Zhang, T. Otomo, L. Li, and S. Nakajima, "Automatic cyberbullying detection on Twitter using bullying expression dictionary," in *Proc. 13th Asian Conf. (ACIIDS)*, Phuket, Thailand, in Lecture Notes in Computer Science, vol. 12672, N. T. Nguyen, S. Chittayasothorn, D. Niyato, and B. Trawinski, Eds., Cham, Switzerland: Springer, Apr. 2021, pp. 314–326.

[27] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proc. Social Mobile Web, Papers ICWSM Workshop*, Catalonia, Spain, Jul. 2011, pp. 11–17.

[28] S. Agrawal and A. Awekar, "Deep learning for detecting cyberbullying across multiple social media platforms," in *Proc. 40th Eur. Conf. IR Res. (ECIRG)*, Grenoble, France, in Lecture Notes in Computer Science, vol. 10772, P. B. Piwowarski, L. Azzopardi, and A. Hanbury, Eds., Cham, Switzerland: Springer, Mar. 2018, pp. 141–153.

[29] T. Davidson, D. Warmsley, M. W. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. 11th Int. Conf. Web Social Media*, Montréal, QC, Canada, May 2017, pp. 512–515.

[30] S. V. Morzhov, "Modern approaches to detecting and classifying toxic comments using neural networks," *Autom. Control Comput. Sci.*, vol. 55, no. 7, pp. 607–616, Dec. 2021.

[31] S. T. Luu and N. Nguyen, "UIT-ISE-NLP at SemEval-2021 task 5: Toxic spans detection with BiLSTM-CRF and ToxicBERT comment classification," in *Proc. 15th Int. Workshop Semantic Eval.*, A. Palmer, N. Schneider, N. Schluter, G. Emerson, A. Herbelot, and X. Zhu, Eds., Bangkok, Thailand, 2021, pp. 846–851.

[32] Y. Fang, S. Yang, B. Zhao, and C. Huang, "Cyberbullying detection in social networks using bi-GRU with self-attention mechanism," *Information*, vol. 12, no. 4, p. 171, Apr. 2021.

[33] J. Eronen, M. Ptaszynski, F. Masui, A. Smywiński-Pohl, G. Leliwa, and M. Wroczynski, "Improving classifier training efficiency for automatic cyberbullying detection with feature density," *Inf. Process. Manage.*, vol. 58, no. 5, Sep. 2021, Art. no. 102616.

[34] A. Kumar, S. Abiramy, T. E. Trueman, and E. Cambria, "Comment toxicity detection via a multichannel convolutional bidirectional gated recurrent unit," *Neurocomputing*, vol. 441, pp. 272–278, Jun. 2021.

[35] Z. Wang and B. Zhang, "Improved bi-GRU model for imbalanced English toxic comments dataset," in *Proc. 5th Int. Conf. Natural Lang. Process. Inf. Retr. (NLPIR)*, Sanya, China, Dec. 2021, pp. 24–29.

[36] Z. Wang and B. Zhang, "Toxic comment classification based on bidirectional gated recurrent unit and convolutional neural network," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 21, no. 3, pp. 1–12, May 2022.

[37] A. Kumar and N. Sachdeva, "A bi-GRU with attention and CapsNet hybrid model for cyberbullying detection on social media," *World Wide Web*, vol. 25, no. 4, pp. 1537–1550, Jul. 2022.

[38] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, and H. D. E. Al-Ariki, "DEA-RNN: A hybrid deep learning approach for cyberbullying detection in Twitter social media platform," *IEEE Access*, vol. 10, pp. 25857–25871, 2022.

[39] V. Isaksen and B. Gambäck, "Using transfer-based language models to detect hateful and offensive language online," in *Proc. 4th Workshop Online Abuse Harms*, S. Akiwowo, B. Vidgen, V. Prabhakaran, and Z. Waseem, Eds., 2020, pp. 16–27.

[40] S. Davidson, Q. Sun, and M. Wojcieszak, "Developing a new classifier for automated identification of incivility in social media," in *Proc. 4th Workshop Online Abuse Harms*, S. Akiwowo, B. Vidgen, V. Prabhakaran, and Z. Waseem, Eds., 2020, pp. 95–101.

[41] A. Koufakou, E. W. Pamungkas, V. Basile, and V. Patti, "HurtBERT: Incorporating lexical features with BERT for the detection of abusive language," in *Proc. 4th Workshop Online Abuse Harms*, S. Akiwowo, B. Vidgen, V. Prabhakaran, and Z. Waseem, Eds., pp. 34–43.

[42] B. Chinagundi, M. Singh, T. Ghosal, P. S. Rana, and G. S. Kohli, "Classification of hate offensive and profane content from tweets using an ensemble of deep contextualized and domain specific representations," in *Proc. Forum Inf. Retr. Eval.*, vol. 3159, P. Mehta, T. Mandl, P. Majumder, and M. Mitra, Eds., Gandhinagar, India, Dec. 2021, pp. 491–500.

[43] H. Park and H. K. Kim, "Verbal abuse classification using multiple deep neural networks," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Jeju Island, (South) Korea, Apr. 2021, pp. 316–319.

[44] N. Rezvani and A. Beheshti, "Attention based context boosted cyberbullying detection in social media," *J. Data Intell.*, vol. 2, no. 4, pp. 418–433, Nov. 2021.

[45] Z. Zhao, Z. Zhang, and F. Hopfgartner, "A comparative study of using pre-trained language models for toxic comment classification," in *Proc. Companion Web Conf.*, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds., Ljubljana, Slovenia, Apr. 2021, pp. 500–507.

[46] Y. Huang, R. Song, F. Giunchiglia, and H. Xu, "A multitask learning framework for abuse detection and emotion classification," *Algorithms*, vol. 15, no. 4, p. 116, Mar. 2022.

[47] R. Ul Mustafa, M. S. Nawaz, J. Farzund, M. I. Lali, B. Shahzad, and P. F. Viger, "Early detection of controversial Urdu speeches from social media," *Data Sci. Pattern Recognit.*, vol. 1, no. 2, pp. 26–42, 2017.

[48] M. Z. Ali, S. Rauf, K. Javed, and S. Hussain, "Improving hate speech detection of Urdu tweets using sentiment analysis," *IEEE Access*, vol. 9, pp. 84296–84305, 2021.

[49] M. Amjad, A. Zhila, G. Sidorov, A. Labunets, S. Butt, H. I. Amjad, O. Vitman, and A. Gelbukh, "Overview of abusive and threatening language detection in Urdu at FIRE 2021," in *Proc. Forum Inf. Retr. Eval.*, vol. 3159, P. Mehta, T. Mandl, P. Majumder, and M. Mitra, Eds., Gandhinagar, India, Dec. 2021, pp. 744–762.

[50] M. H. Akram and K. Shahzad, "Violent views detection in Urdu tweets," in *Proc. 15th Int. Conf. Open Source Syst. Technol. (ICOSST)*, Dec. 2021, pp. 1–6.

[51] R. Ali, U. Farooq, U. Arshad, W. Shahzad, and M. O. Beg, "Hate speech detection on Twitter using transfer learning," *Comput. Speech Lang.*, vol. 74, Jul. 2022, Art. no. 101365.

[52] S. Khan and A. Qureshi, "Cyberbullying detection in Urdu language using machine learning," in *Proc. Int. Conf. Emerg. Trends Electr., Control, Telecommun. Eng. (ETECTE)*, Dec. 2022, pp. 1–6.

[53] S. Hussain, M. S. I. Malik, and N. Masood, "Identification of offensive language in Urdu using semantic and embedding models," *PeerJ Comput. Sci.*, vol. 8, p. e1169, Dec. 2022.

[54] M. H. Akram, K. Shahzad, and M. Bashir, "ISE-hate: A benchmark corpus for inter-faith, sectarian, and ethnic hatred detection on social media in Urdu," *Inf. Process. Manage.*, vol. 60, no. 3, May 2023, Art. no. 103270.

[55] M. U. Arshad, R. Ali, M. O. Beg, and W. Shahzad, "UHated: Hate speech detection in Urdu language using transfer learning," *Lang. Resour. Eval.*, vol. 57, no. 2, pp. 713–732, Jun. 2023.

[56] M. S. I. Malik, U. Cheema, and D. I. Ignatov, "Contextual embeddings based on fine-tuned Urdu-BERT for Urdu threatening content and target identification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 7, Jul. 2023, Art. no. 101606.

[57] F. Razi and N. Ejaz, "Multilingual detection of cyberbullying in mixed Urdu, Roman Urdu, and English social media conversations," *IEEE Access*, vol. 12, pp. 105201–105210, 2024.

[58] M. S. Khan, M. S. I. Malik, and A. Nadeem, "Detection of violence incitation expressions in Urdu tweets using convolutional neural network," *Exp. Syst. Appl.*, vol. 245, Jul. 2024, Art. no. 123174.

[59] M. Alam and S. Ul Hussain, "Sequence to sequence networks for roman-Urdu to Urdu transliteration," in *Proc. Int. Multitopic Conf. (INMIC)*, Nov. 2017, pp. 1–7.

[60] M. Alam and S. Ul Hussain, "Deep learning-based roman-Urdu to Urdu transliteration," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 35, no. 4, Mar. 2021, Art. no. 2152001.

[61] S. Haider, "Urdu word embeddings," in *Proc. 11th Int. Conf. Lang. Resour. Eval.*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds., Miyazaki, Japan, May 2018. [Online]. Available: https://aclanthology.org/L18-1155/

[62] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," in *Proc. 10th Hellenic Conf. Artif. Intell.*, Patras, Greece, Jul. 2018, pp. 1–6.

[63] H. Rizwan, M. H. Shakeel, and A. Karim, "Hate-speech and offensive language detection in Roman Urdu," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Nov. 2020, pp. 2512–2522.

[64] Z. Zhang, D. Robinson, and J. Tepper, in *Proc. 15th Int. Conf.*, Heraklion, Greece, in Lecture Notes in Computer Science, vol. 10843, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., Cham, Switzerland: Springer, Apr. 2018, pp. 745–760.

[65] H. H. Saeed, K. Shahzad, and F. Kamiran, "Overlapping toxic sentiment classification using deep neural architectures," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, H. Tong, Z. J. Li, F. Zhu, and J. Yu, Eds., Singapore, Nov. 2018, pp. 1361–1366.

[66] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. for Comput. Linguistics, Human Lang. Technol. NAACL-HLT*, vol. 1, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[67] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[68] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "MT5: A massively multilingual pre-trained text-to-text transformer," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds., 2021, pp. 483–498.

[69] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., Jul. 2020, pp. 8440–8451.

[70] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, and P. Talukdar, "MuRIL: Multilingual representations for Indian languages," 2021, *arXiv:2103.10730*.

**HAFIZ HASSAAN SAEED** received the B.S. degree from COMSATS University Islamabad, Pakistan, and the M.S. degree from FAST-NUCES Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the Data Science Laboratory (DSL), Information Technology University (ITU), Lahore, Pakistan. Concurrently, he is the Team Lead and a Senior Data Scientist with ADDO AI. His research interests include machine learning, deep learning, text mining, natural language processing, hate speech and its repercussions in online space, and language models.

**TAHIR KHALIL** received the B.S. degree in mechatronics from the University of Engineering and Technology, Lahore, Pakistan, and the M.S. degree from Information Technology University, Pakistan. He is currently pursuing the Ph.D. degree in computer science with Kyung Hee University, South Korea. His research interests include model compression focusing on quantization techniques for diffusion models. He aims to optimize the performance of these models on resource-constrained hardware while maintaining high generative quality. His work explores innovative approaches to improve the efficiency and scalability of generative models for real-world applications.

**FAISAL KAMIRAN** received the Ph.D. degree from Eindhoven University of Technology (TU/e), The Netherlands, in 2011. Following his studies, he became a Postdoctoral Fellow with the King Abdullah University of Science and Technology (KAUST). During his doctorate research, he developed fairness-aware AI and machine learning models that have significant real-world applications in legal, social, and economic domains, particularly with the enforcement of strict privacy regulations like GDPR. He is currently an Associate Professor with Information Technology University. Additionally, he is the Co-Founder and the President of ADDO AI. He has published over 50 refereed papers in reputable journals and conferences appearing in esteemed data science venues, such as ICDM, ECML/PKDD, EMNLP, PAKDD, CIKM, SAC, and *Knowledge and Information Systems*. His research interests include fairness-aware data analytics and machine learning to protect marginalized communities and individual rights, ICTD, social media analysis, and text mining. According to Google Scholar, he has an H-index of 22 and his research has garnered more than 5000 citations.