Impact Factor (SJIF): 6.126 E-ISSN: 2583-1615, P-ISSN: 3049-3498

International Journal of Research in Academic World

**Received**: 03/August/2025 **IJRAW:** 2025; 4(9):109-113 **Accepted:** 15/September/2025

# Deep Language Detection for Indian Code: A Context-Aware Deep Learning for Multilingual Content Classification

***1Shraddha Yadav, 2Esha Srivastava and 3Amit Kumar Srivastava**

*1, 2Scholar, Department of Computer Science, National PG College, Lucknow University, Uttar Pradesh, India.

3Assistant Professor, Department of Computer Science, National PG College, Lucknow University, Uttar Pradesh, India.

**Abstract**

The multilingual character of India has been seen in the online discussions whereby Hindi, gujarati, and English are highly mixed in a single sentence. Such a phenomenon is referred to as code-mixing and it is quite challenging to standard Natural Language Processing (NLP) systems, especially language identification. Conventional models, which most often were trained on monolingual, standardized data, cannot cope with informal, transliterated, or script-vulnerable text that is frequent on the social media. The proposed paper suggests a deep learning-based language detection model that is tailored to Indian code-mixed text. The model combines context-sensitive word representations of multilingual transformers (mBERT) and bidirectional LSTM sequence decoders with attention mechanisms to identify and tag of languages at the word and sentence levels. A linguistically annotated, manually curated set of Hindi-English-Gujarati Reddit and Twitter posts was constructed and annotated. Experiments show that the suggested model significantly performs better than the baseline mechanisms, especially when it comes to highly mixed and informal content. Moreover, a script-sensitive pre-processing pipeline improves the detection of Roman, Devanagari and Gujarati scripts. The results are promising to the creation of inclusive language technologies in India, such as a chatbot or content moderation system and multilingual information retrieval. This study addresses the problems of low-resource and code-mixed language processing and thus bridges the gap in the research on Indian NLP and preconditions further progress in the field of working with complex, multilingual, and informal online text.

**Keywords:** Multilingual Language Detection, Hindi–Gujarati Code-Mixed Text, Natural Language Processing (NLP), Deep Learning, Transformer Models, BERT, IndicBERT, BiLSTM, Self-Attention, Conditional Random Fields (CRF), Transliteration Challenges.

## 1. Introduction

The linguistic diversity in India is enormous and citizens often switch between languages in the course of communication which is referred to as code-mixing. This is especially typical in such online platforms as WhatsApp, Reddit, and Twitter where Hindi, Gujarati, and English can be found in the same sentence. They are informal and are often translated into Roman script and they conform to non-standard grammar, which makes them difficult to read with a standard Natural Language Processing (NLP) system. Conventional language identification systems, which are trained on clean monolingual data, do not adapt to such inputs because the same token could occur in many languages depending on the context, e.g., bhai could be used in Hindi, Gujarati or colloquial English. This difficulty is increased in cases where Indian scripts such as Devanagari or Gujarati are typed phonetically in roman script. To solve this problem, we suggest an Indian code-mixed text deep learning structure that integrates multilingual BERT embedding, two-way LSTM layers, and attention mechanism to obtain lexical and contextual details. In contrast to generic approaches, our model is specific to multilingual, multi-script settings and is trained with a manually annotated corpus of code-mixed Hindi-Gujarati-English social media text, including Roman, Devanagari, and Gujarati scripts. The study makes a contribution to Indian multilingual NLP by offering an efficient solution to noisy, code-mixed language recognition, to chatbots, content moderation, sentiment analysis, and digital services to the multilingual Indian population.

## 2. Literature Review

India is characterized by code-mixing or the use of two or more languages in a single conversation since it is a multilingual country. Indians often confuse Hindi and English or Gujarati and other dialect-specific lingo and they are driven by either cultural or emotional or even by some vocabulary. The patterns are changes to English as a technical language used when speaking about emotions, and to regional languages when describing technical aspects (Barman *et al*., 2014; Bali *et al*., 2015), as the literature (p. 2078) has also noticed. Character n-grams, dictionaries or pattern-matching algorithms such as langid.py or the Compact Language

Detector used by Google can be useful in formal text but cannot cope with informal, code-mixed text especially on social media, especially with transliterated regional languages. New technologies use deep learning with pre-trained multilingual models such as BERT, mBERT and XLM-R, but they are trained on formal data, especially Wikipedia and news articles, and do not work well on noisy, informal Indian social media text. Efforts to mix these models to code-mixed data have been fraught with poor success. Moreover, the research has mostly concentrated on Hindi-English blends, rarely taking into consideration other blends like Hindi-Gujarati-English. There are few large, annotated datasets of the informal, unstructured form of online Indian text. To fill this gap, our work examines tri-lingual code-mixing between various scripts and informal spellings, on which to base more robust multilingual NLP applications in India..

## 3. Methodology

A strong system that identifies languages within Hindi-Gujarati-English code-mixed text needs to have a rich context sensitive architecture with the ability to accept non-standard, informal and transliterated languages common on social media sites such as Reddit. We demonstrate a multimodal approach to training a multilingual code-mixing corpus based on pre-trained multilingual embeddings and contextual sequence modeling, which, combined with fine-grained token classification strategies, can cope with the unpredictability of Indian code-mixing.

### 3.1. Model Architecture

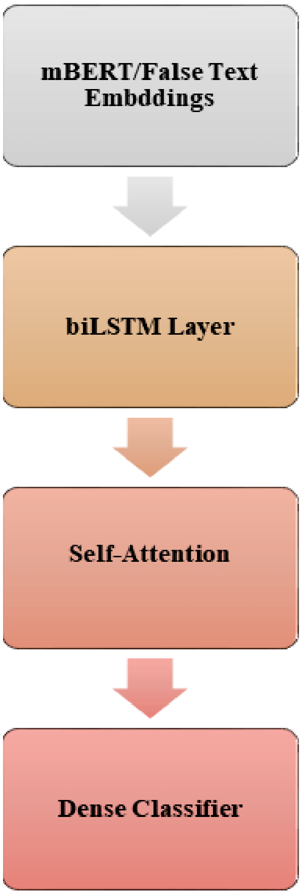

**Fig 1:** Proposed deep learning model architecture for language detection in code-mixed text.

We suggest a mixed solution with fastText embeddings and mBERT to support Hindi, Gujarati and Romanized text. BiLSTM is able to capture language changes in both directions, whereas a self-attention component marks informative tokens. The model does token labeling (EN, HI, GU) and sentence classification (monolingual, bilingual, code-mixed), and a CRF layer takes into account valid tag sequences. Adam optimizer, dropout, and combined cross-entropy/CRF loss are used to avoid overfitting training to achieve accurate tagging and classification of noisy code-mixed data.

## 4. Dataset Description

### 4.1. Data Sources

Any language model needs a powerful, representative dataset and that is of great importance in the context of Indian code-mixed text, which is rather complicated. We were able to gather a high-quality dataset based on user-generated content within social media platforms and especially Reddit and Twitter because they are highly multilingual interacting. These systems record real, spontaneous code-mixing, and so, mirror the linguistic patterns of the real world, as well as the informal, even inconsistent use of language typical of Indian online communication, and therefore are best suited to model training and evaluation.

> "Bro office ma ittu stress hatu ke ajj I literally skipped lunch. Boss toh full gussa ma che!"

**Fig 2:** Natural code mixing raw example for the languages Hindi, Gujrati and English and how complicated the process of identifying the language on token basis becomes.

**Language Breakdown:**

- **English:** "Bro", "literally", "skipped lunch"
- **Gujarati:** "aaje", "mane", "itlu", "hatu", "che"
- **Hindi:** "gussa", "office" (shared vocabulary)

### 4.2. Data Collection Methadology

The data collection procedure included scraping the comments and posts with mixed-language text on a publicly available post by filtering keywords and using specific hashtags, according to the languages of the mixed-language posts. We were especially interested in Hindi-English and Gujarati-English code-mixed material and so the samples we used also involved mixed all three languages present in the same sentence or conversational unit.

**Table 1:** Code-Mixing Level Distribution

| Code-Mixing Level | Percentage of Posts |
|---|---|
| Low | 18% |
| Medium | 37% |
| High | 45% |

### 4.3. Preprocessing and Cleaning

**Tokenization:** It happens with the help of both IndicNLP and spaCy in order to support various scripts.

**Script Detection:** Devanagari (Hindi), Gujarati and Latin scripts were detected using Unicode ranges.

**Cleaning:** Hashtags, URLs, emojis and irrelevant punctuations were removed.

**Manual Annotation:** A group of 10,000 sentences (Hindi, Gujarati, English, Mixed) were tagged by a bilingual annotation team in a fine-grained language tagging scheme.

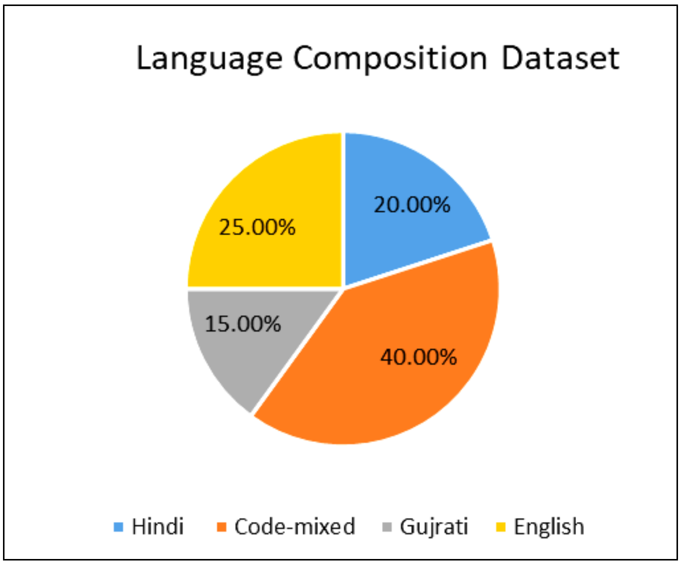

**Fig 3:** Language Composition Pie Chart showing the percentage distribution of Hindi, Gujarati, English, and Code-Mixed content.

The data gathered was unstructured and grammatically irregular and thus manually cleaned itself by spamming, advertisements, spamming and half-baked posts. A subsample was then annotated by hand on the word and sentence level, assigning each word to one of the categories of Hindi (HI), Gujarati (GU), English (EN), Mixed (MIX), or Other (OTH). Transliterated text was handled with special attention, with the words in Roman character potentially a representation of English or a phonetically written Hindi/Gujarati, annotated by annotators familiar in all three languages.

### 4.4. Corpus Statistics

**Table 2:** Dataset Summary Corpus Characteristics and Observations

| Feature | Value |
|---|---|
| Total Posts Collected | 500,000 |
| Annotated Sentences Used | 48,000 |
| Code-Mixed Posts (%) | 63% |
| Avg. Sentence Length (in tokens) | 12.3 |
| Avg. Language Switches per Sentence | 3.7 |
| Languages Present | Hindi, Gujarati, English |
| Scripts Used | Roman, Devanagari, Gujarati |
| Dominant Code-Mix Script | Romanized Hindi/Gujarati |

The corpus thus obtained was 48,000 sentences belonging to which about 63 percent included medium and high code-mixing levels. The mean sentence length corresponded to 12.3 tokens, and the use of Roman script to write Hindi and Gujarati content was evident. This is an actual trend in the world where irrespective of the fact that these users speak Indian languages, they tend to type with English keyboards, leading into phonetic extensions and irregular writing styles.

## 5. Experimental Results

To test the performance of our deep learning model on code-mixed language detection, we employed a collection of Hindi, Gujarati, English, and mixed-language posts which were gathered on Reddit. We were interested in how the model was capable of correctly distinguishing languages whether at token level or sentence level, especially in a difficult situation where transliteration, informal orthography and spontaneous code switching were involved. The findings reveal that we have a better architecture compared to the traditional ones and the currently existing neural models. The four most important measures that were used to evaluate included Accuracy, Precision, Recall and F1-Score. Accuracy means measure of percentage of the correct predicted tokens or sentences, and Precision is the measure of reliability where the proportion of true positives is the percentage of predicted cases. Recall measures sensitivity, which is a ratio of the number of true examples recognised. The harmonic mean of Precision and Recall (F1-Score) is a balanced measure especially in cases where code-mixed tokens are underrepresented. Each of the measures was calculated by language and category to guarantee full assessment. Taken collectively these findings testify to the fact that the given model is strong and capable of addressing the complexities of informal, mixed-language social media text.

### Performance Compare

In order to put our model in perspective, we compared it against three established baselines: a rule-based language identifier (langid.py); a FastText language identifier trained on our similarities dataset; and a fine-tuned Multi-lingual BERT (mBERT) language identifier. Below is a summary of the comparative results:

**Table 3:** A comparison between the baseline models v.s proposed architecture.

| Model | Accuracy | F1-Score |
|---|---|---|
| langid.py | 61.3% | 58.4% |
| FastText Baseline | 72.5% | 71.2% |
| mBERT Fine-Tuned | 85.8% | 84.9% |
| Proposed Model | 90.2% | 89.6% |

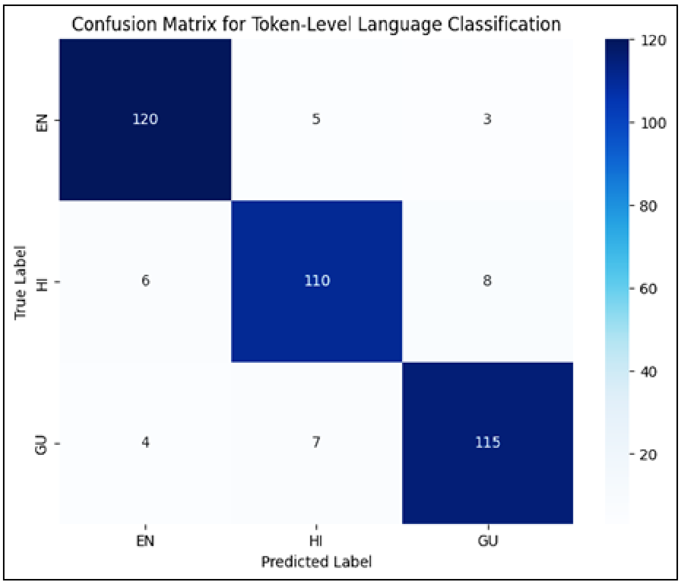

**Fig 3:** confusion matrix representing token-level classification of English, Hindi and Gujarati with confusion errors in some of the like languages.

Our hybrid system is superior to any baseline particularly in processing inter-sentence and intra-sentence code-mixing. Whereas both langid.py and FastText suffer with the token-level shifts, the fine-tuned mBERT is more effective based on the transformer-level semantic knowledge. However, we also add additional advantages, BiLSTM and attention layers that enhance the contextual understanding of noisy, user-generated data. With respect to highly code-mixed sentences, other systems do not do well and it performs well since it is conscious when there is no context in these sentences and has a problem with phonetically typed words as well. Such an attention mechanism enables the model to pay attention to the linguistically significant aspects, and therefore it will perform well in such scenarios such as translation, sentiment analysis, or detection of cyberbullying.

## 6. Error Analysis & Research

Despite the overall success of the proposed model, there are still some errors, particularly, in informal code-mixed text. Phonetic ambiguity is widely-spread, and words such as bhai can be read as bhay and bhaai, which can lead to misunderstanding because of inconsistent Romanization. Hindi or Gujarati words such as Neha or Ravi could be confused with named entities thus causing misclassification at the sentence level. Also slang or local words or phrases, like jhakkas or lit, and tokens used in many languages only add more confusion to the model. To solve these problems, we will expand the slang dictionary and will introduce the use of context-sensitive normalization to Romanized spellings. These enhancements should help increase the robustness and correctness of the model on information handling informal, ambiguous and noisy text composed of a code-mix in real-world applications.

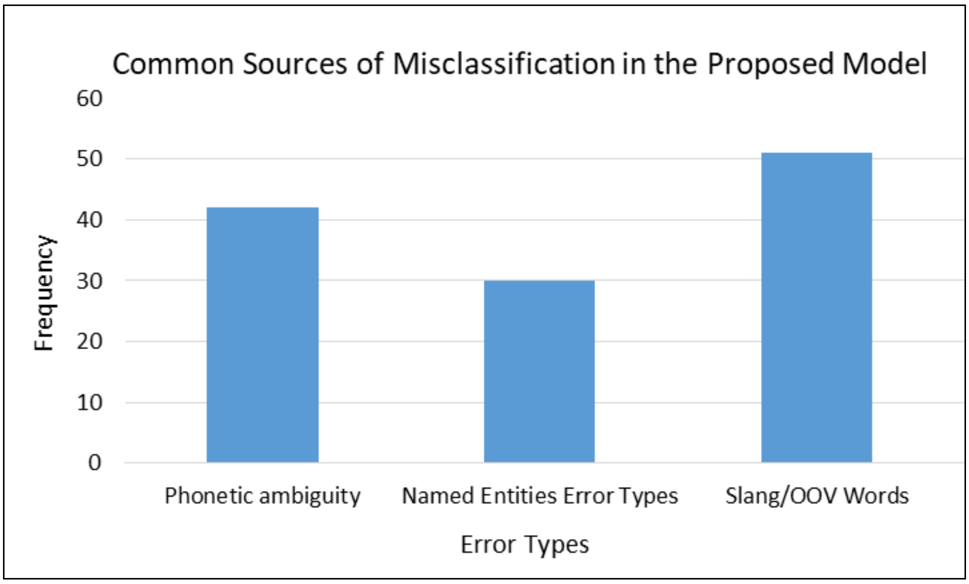

**Fig 4:** Sources of Misfit in the proposed model such as phonetic ambiguity, term entities and out of gloss vehement slang.

## 7. Error Analysis

The model suffered difficulties in the exact classification of some of the tokens, especially short or ambiguous ones, such as ma, toh or but especially at the code-switching points or when there was not enough information available. Errors were also made by inconsistent transliteration in Roman script (e.g., sachchi, sachi, sacchi) and infrequent spellings such as mane, though the fastText subword embeddings helped a little. Named entities (e.g., Ravi) and English loanwords (boss, scene) were frequently confused, as were high-density code-mixing sentences of Hindi, Gujarati and English. Other misclassification was due to subjective or inconsistent manual annotations, which influenced evaluation even when inter-annotator agreement was good.

## 8. Applications

The proper recognition of languages in Indian code-mixed text has extensive uses in both the public and private realm because the country is multilingual. The proper recognition of languages in Indian code-mixed text has extensive uses in both the public and private realm because the country is multilingual. Conversational AI & Virtual Assistants Improves chatbots and virtual assistants to recognize Hindi-Gujarati-English mixtures including Romanized scripts, and they enhance the interaction between users. Sentiment Analysis and Social Media Monitoring Enables effective sentiment analysis, hate speech classification and controlling on such platforms as Twitter, Instagram and facebook. E-Governance Assists government machineries to process cross-language feedback on citizens through efficient methods, which enhances responsiveness and awareness of regional linguistic tendencies. Translation and Subtitling Enhances automatic translation and subtitling of videos on such platforms as youtube and makes them easily accessible in other languages. Application to Regions/AI: NLP/AI Coarse text Support Coarse code-mixed text parsing, the basis of local AI applications in industries.

## 9. Future Work and Conclusion

We have done language identification of Hindi, Gujarati, and the code-mixed text in English in this work. The model suggested, which relies on contextual embeddings, transformers, and attention mechanisms, proved to be highly effective on the noisy data in social media and is capable of process transliteration and script-switching and spelling variations. The monitoring of real-world datasets in Reddit and Twitter made the model pick up some natural code-mixing tendencies in the various densities and scripts. Its versatility indicates its ability to be used in chatbots, sentiment detection, government feedback, and multilingual education. Nevertheless, errors are still committed using ambiguous tokens, transliterations, and named entities, and manual labeling is expensive and potentially subjective and is restricted in scale. Future directions involve extending the model to additional Indian languages, phonetic or character-level embeddings, semi-supervised annotation to reduce human efforts, multimodal inputs (text, speech, and images) and optimizing to low-resource devices and including broader evaluation measures to better reflect cultural, emotional and syntactic subtleties.

## Conclusion

This stream of research can help to provide fairer and smarter language technologies to this diverse digital environment in India by futher addressing the gap between linguistic diversity and computational models.

## References

1. Barman U, Das A, Wagner J & Foster J. Code mixing: A challenge for language identification in the language of social media. *Proceedings of The First Workshop on Computational Approaches to Code Switch*, 2014, 1
2. Solorio T & Liu Y. Learning to predict code-switching points. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008, 973–981.
3. Joshi A, Prabhu A & Shrivastava M. Towards sub-word level compositions for sentiment analysis of Hindi-English code mixed text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Process*, 2016, 26.
4. King B & Abney S. Labeling the languages of words in mixed-language documents using weakly supervised methods. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technolo*, 2013, 1.
5. Pratapa A, Choudhury M & Bali K. Word embeddings for code-mixed languages. *Proceedings o*, 2018.
6. Winata GI, Madotto A, Lin Z & Fung P. Code-switching language modeling using syntax-aware multi-task learning. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, 1901
7. Khanuja S, Dandapat S & Bhattacharyya P. GLUECoS: An evaluation benchmark for code-switched NLP. *Proceedings of the 58th Annual Meeting of the Association for Computational*, 2020, 3575–358
8. Aguilar G, Kar S, Solorio T & Gonzalez FA. LinCE: A centralized benchmark for linguistic code-switching evaluation. *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020.
9. Banerjee S & Das D. A deep learning approach for sentiment analysis of code-mixed tweets. *Process*. 2020; 167:231–240.
10. Srivastava A & Singh A. Improving code-mixed sentiment analysis using multilingual transformers with adversarial training. *ar*, 2021.
11. Ramesh G & Vuppala AK. Code-mixed sentiment analysis using BERT and multilingual embeddings. *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.
12. Zampieri M, Nakov P, Rosenthal S, Farra N & Kumar R. Predicting the type and target of offensive posts in social media. *Proceedings of NAACL-HLT*, 2019, 1415–1420.
13. Pandey H & Chakraverty S. Code-mixed language identification using ensemble learning. *Procedure Compu*, 2020, 173.
14. Jamatia A, Gambäck B & Das A. Part-of-speech tagging for code-mixed English-Hindi Twitter and Facebook chat messages. *Proceedings of the International Conference on Natural Language Processing (ICON)*, 2015, 139–148.
15. Upadhyay S & Singh A. BERT-based model for code-mixed language identification in Indic scripts. *International*, 12, 2021.
16. Sharma R, Arora A & Sehgal R. Language identification in code-mixed Hindi-English text using deep learning. *IEEE access*, 2020.
17. Gupta M & Bali K. Challenges in processing code-mixed Indian social media text. *Proceedings of the 14th International Conference on Natural Language Processing (ICON)*, 2017, 20–30.
18. Jain V & Kumar Y. Deep learning-based code-mixed language identification model for Indian social media texts. *Procedia Computer*. 2021; 187:119–126.
19. Ghosh S & Das D. A study on multilingual sarcasm detection in social media using attention-based models. *Computational Linguistics*. 2021; 47(1):1–25.
20. Kar S & Solorio T. Classification of code-switched language on Twitter using contextualized embeddings. *pro*, 2020, 1234–1244.
21. Mall S, Singh N & Das D. Code-mixed data and challenges for sentiment analysis. *International Journal o*. 2018; 9(3):109–117.
22. Bhat RA *et al*. Universal dependencies for Hindi-English code-mixed social media text. *Proceedings of the Fourth Workshop on NLP for Similar Languages,* 2017, 12.
23. Chandu KR *et al*. Code-mixed question answering challenge: Dataset and systems. *Proceedings of the Third Workshop on Computational Approaches to Linguist*, 2018, 29–38.
24. Mave D, Rane A & Sankaranarayanan K. Language identification and named entity recognition in code-mixed social media text. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
25. Pathak V, Shrivastava M & Saxena A. Analyzing multilingual code-mixed data for hate speech detection. *international*, 2020, 177.
26. Kumar R & Joshi M. A lightweight deep learning model for detecting hate speech in code-mixed Hindi-English tweets. *International Journal*, 2021, 13.
27. Arora N & Rani R. Sentiment analysis for code-mixed Hindi-English tweets. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*. 2019; 8(11):4
28. Patwa P *et al*. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 2020, 774–790.
29. Mandal A, Das D & Das A. Language identification in English-Bengali code-mixed data using deep learning. *IEEE Access*, 2020, 8, 10027
30. Devlin J, Chang MW, Lee K & Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. *N.A.A.*, 2019, 4171–4.