

MULTILINGUAL CODE-MIXED TEXT NORMALIZATION USING DEEP LEARNING: A UNIFIED FRAMEWORK FOR ROBUST NLP SYSTEMS

Mr. Ganesh Bhagwat

Assistant Professor, MCA Department, DES's NMITD.

Abstract:

The increasing adoption of digital communication platforms has significantly altered the structure and style of written language. Users today communicate through social media, messaging applications, online forums, and service portals using short, informal, and conversational expressions rather than grammatically structured sentences. In multilingual societies such as India, this behaviour becomes more complex due to the frequent mixing of multiple languages within a single expression. Such communication, commonly referred to as code-mixed text, often combines vocabulary from different languages, phonetic spellings, abbreviations, and inconsistent grammatical structures. Words belonging to regional languages are frequently written using Roman script instead of native scripts, producing multiple orthographic variations for the same semantic concept. While humans interpret such text effortlessly using contextual knowledge, Natural Language Processing (NLP) systems struggle to process it reliably because most models are trained on clean and standardized datasets. The absence of standardized representation in multilingual user-generated text significantly affects the performance of computational systems. The same word may appear in multiple forms depending on pronunciation and typing habits, preventing machines from identifying equivalence between lexical variants. Consequently, downstream applications such as search engines, chatbots, sentiment analysis tools, machine translation systems, and automated customer support platforms generate inaccurate results when handling real-world user input. This limitation highlights the importance of preprocessing techniques capable of converting noisy and informal text into a consistent machine-processable format. Text normalization provides a solution by transforming irregular linguistic expressions into a standardized representation while preserving semantic meaning. In multilingual code-mixed communication, normalization requires identification of the language at word level, correction of phonetic spelling variations, expansion of abbreviations, contextual selection of appropriate lexical forms, and optional conversion to a unified script. However, existing approaches primarily address isolated aspects of the problem, such as spelling correction or transliteration, and often rely on handcrafted rules or limited language pairs. Such methods lack adaptability to unseen variations and fail to capture contextual dependencies present in real communication. This chapter presents a deep learning-based framework for multilingual code-mixed text normalization. The proposed approach treats normalization as a sequence transformation task in which noisy user-generated text is converted into standardized Hindi representation. The framework consists of multiple stages including tokenization, language identification, candidate generation using phonetic similarity, context-aware selection using neural models, and script standardization. By learning transformation patterns from annotated examples rather than relying on predefined dictionaries, the system generalizes to previously unseen word variations and produces consistent normalized output. The normalized representation significantly improves the reliability of NLP systems operating on real-world text. Search engines retrieve relevant information more accurately when queries are standardized. Chatbots and automated service systems interpret user requests correctly, reducing communication errors. Sentiment analysis produces more reliable insights because linguistic noise is minimized. Translation systems generate higher quality output due to consistent input representation. The framework therefore acts as a foundational pre-processing layer that enhances performance across multiple language processing applications. The chapter concludes that multilingual code-mixed normalization is a crucial step for Linking human language with computational systems machine language understanding. By addressing linguistic variability through a structured deep learning approach, the proposed framework enables computational systems to operate effectively in multilingual environments. Future research directions include extension to additional languages, real-time normalization during typing, and integration with speech-based interfaces.

Keywords: Code-Mixed Text, Text Normalization, Deep Learning, Multilingual NLP, Transformer, Sequence-to-Sequence.

Introduction

The rapid evolution of digital era has significantly changed the way individuals express themselves in written form. With the widespread adoption of social media such as WhatsApp, Twitter (now X) platforms, Instagram, and various messaging applications, communication has shifted from formal, grammatically structured sentences to short, informal, and conversational text. This transformation reflects the need for speed, convenience, and expressiveness in everyday digital interactions. As a result, users frequently employ abbreviations, phonetic spellings, emojis, and mixed-language expressions to convey meaning efficiently.

In multilingual societies like India, this changes in technology becomes even more complex due to the coexistence of multiple languages within the same communicative environment. India is home to hundreds of languages and dialects, and individuals often switch between languages such as Hindi, English, Marathi, Tamil, and others within a single conversation. This linguistic behaviour, known as **code-mixing**, now become a natural and widely accepted mode of communication in digital spaces. Users seamlessly blend languages depending on context, audience, and convenience, often without conscious effort.

The most prominent characteristics of code-mixed communication in India is the **Roman script for regional languages use**. For example, Hindi or Marathi words are frequently typed using English alphabets instead of their native scripts. This leads to multiple spelling variations for the same word based on pronunciation and personal typing habits. For instance, the Hindi word “है” may appear as “hai,” “hey,” or “he” in Roman script. Such variations introduce inconsistency and ambiguity in textual data, making it challenging for computational systems to interpret meaning accurately.

Real-world examples from messaging platforms clearly illustrate this phenomenon. Consider a common WhatsApp message:

- “Kal meeting hai pls confirm”

This sentence combines Hindi (“Kal,” “hai”) with English (“meeting,” “pls confirm”), along with an abbreviation (“pls” for “please”). Similarly, a social media post might read:

- “Aaj mood off hai yaar”

In above the Hindi words are written in Roman script, and the expression conveys an emotional state in an informal manner. Another example from Twitter could be:

- “Party mast thi but traffic ne mood kharab kar diya”

This sentence blends Hindi and English while maintaining an informal conversational tone. Such examples highlight the complexity and richness of code-mixed text, where linguistic boundaries are fluid and context plays a crucial role in interpretation.

While humans can easily understand such expressions using contextual and cultural knowledge, **Natural Language Processing (NLP) systems face significant challenges** in processing them. Most of the NLP models are trained on clean, standardized, and monolingual datasets, which do not reflect the variability present in real-world user-generated content. As a result, these systems struggle with this kind of task such as language identification, sentiment analysis, machine translation, and information retrieval when dealing with code-mixed text. The impact of this challenge is particularly evident in AI-driven applications. For example, chatbots used in customer support may fail to understand user queries written in mixed languages, leading to incorrect or irrelevant responses. Search engines may retrieve inaccurate results when queries contain non-standard spellings or mixed-language phrases. Sentiment analysis systems may misinterpret emotional expressions due to the informal language and abbreviations presence. Similarly, machine translation systems may produce poor-quality output when the input text is inconsistent or ambiguous.

Literature Review

Early research in text normalization focused on monolingual English datasets using rule-based and dictionary-based approaches. These methods rely on predefined rules and lexical resources, which limit their ability to handle dynamic and evolving language patterns. Deep learning approaches, including RNNs, LSTMs, and Transformer models, have demonstrated improved performance by capturing contextual relationships. Pre-trained models for example BERT and mT5 further enhance multilingual understanding. However, existing work primarily focuses on isolated tasks such as transliteration or spelling correction rather than providing a unified framework. I am addressing this gap by proposing an integrated normalization pipeline in this paper.

Problem Statement

Multilingual code-mixed text introduces several challenges:

- Linguistic variability due to phonetic spelling
- Difficulty in word-level language identification
- Script inconsistency (Romanized regional languages)
- Contextual ambiguity
- Limited annotated datasets

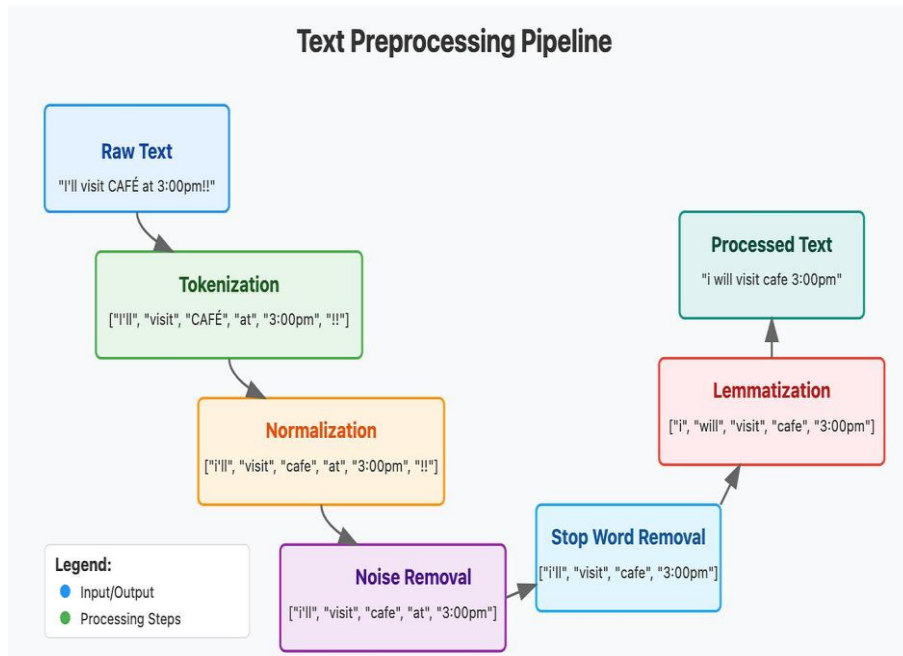
These issues significantly affect the NLP systems performance.

Proposed Methodology

Framework Overview

The proposed system is a unified pipeline consisting of:

1. Input pre-processing
2. Tokenization
3. Language identification
4. Candidate generation
5. Context-aware normalization
6. Output generation



Algorithm

Input: Code-mixed text

Output: Normalized text

1. Preprocess input
2. Tokenize sentence
3. Identify language of each token
4. Generate normalization candidates
5. Apply deep learning model (Seq2Seq/Transformer)
6. Select best output using context
7. Generate normalized sentence

Model Description

The normalization task is treated as a sequence transformation problem using:

- Encoder–Decoder architecture
- Attention mechanism
- Transformer-based models

These models enable:

- Context-aware learning
- Handling of ambiguity
- Generalization to unseen variations

Example

Input: "Mujhe order cancel karna hai pls help"

Output: "I want to cancel my order, please help"

Experimental Evaluation

Dataset

L3Cube, HinglishNorm (These datasets have been used for experiments)

Sample Data L3Cube style

ID	Raw Text (Hinglish)	Normalized Text
1	yeh movie bht mast h	yeh movie bahut mast hai
2	service bilkul bakwas hai	service bilkul bakwas hai
3	product acha hai but delivery late	product achha hai but delivery late hai
4	kya faltu app hai yaar	kya faltu app hai yaar
5	amazing experience loved it	amazing experience loved it

Sample Data HinglishNorm style

ID	Input (Noisy Text)	Normalized Output
1	m kal clg nhi jaunga	main kal college nahi jaunga
2	tum kahan ho bro	tum kahan ho bhai
3	ye pic bohot cool h	yeh picture bahut cool hai
4	pls mujhe help kro	please mujhe help karo
5	apka reply kyu nhi aya	aapka reply kyu nahi aaya

The dataset consists of multilingual code-mixed text collected from:

- Social media platforms
- Messaging applications
- Online forums

Parameter	Value
Total Sentences	10,000
Languages	Hindi-English, Marathi-English
Tokens	~85,000

Experimental Setup

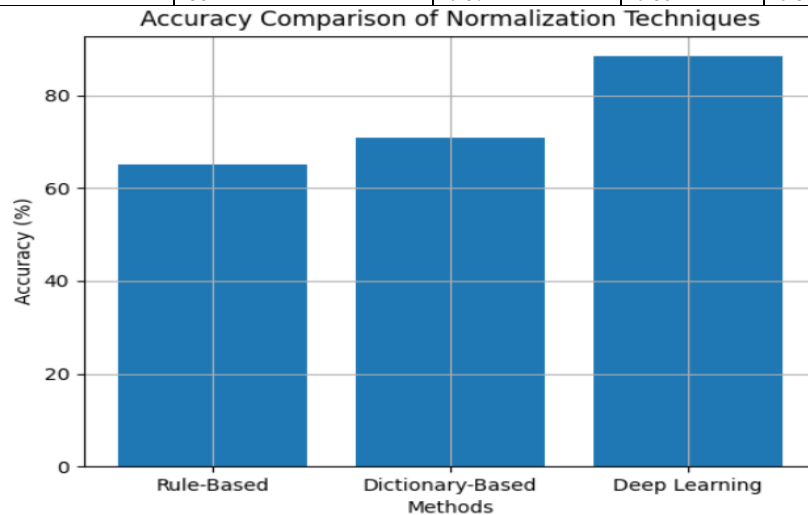
- Model: Transformer-based Seq2Seq
- Optimizer: Adam
- Epochs: 20
- Batch Size: 32
- Train/Test Split: 80/20

Evaluation Metrics

- Accuracy
- Precision
- Recall
- BLEU Score

Results

Method	Accuracy (%)	Precision	Recall	BLEU Score
Rule-Based	65.2	0.63	0.61	0.55
Dictionary-Based	70.8	0.69	0.67	0.60
Deep Learning Model	88.4	0.87	0.86	0.82



Task-Based Improvements

Sentiment Analysis

Input Type	Accuracy
Raw Text	68%
Normalized Text	85%

Machine Translation

Input Type	BLEU Score
Raw Text	0.58
Normalized Text	0.80

Discussion

The results demonstrate that:

- Deep learning improves performance by over 20%
- Context-aware models outperform traditional approaches
- Normalization significantly enhances downstream tasks

Applications

The proposed framework is applicable in:

- Chatbot's → improved user interaction
- Sentiment analysis → better emotion detection
- Machine translation → enhanced translation quality
- Search engines → improved query understanding

Challenges

- Limited annotated datasets
- High computational cost
- Language bias toward dominant languages
- Handling evolving slang and expressions

Ethical Considerations

- Risk of linguistic diversity loss
- Bias in multilingual models
- Privacy concerns in data usage
- Lack of interpretability in deep learning

Conclusion

This paper presents a unified deep learning-based framework for multilingual code-mixed text normalization. The proposed approach effectively reduces linguistic variability and improves the performance of NLP systems. Experimental results confirm the modernization of deep learning models over traditional methods in handling contextual and multilingual complexities.

Future Work

Future directions include:

- Real-time normalization systems
- Integration with speech processing
- Support for low-resource languages
- Use of large language models
- Multimodal normalization

References

1. Barman, U., Das, A., Wagner, J., & Foster, J. (2014). Code mixing: A challenge for language identification in the language of social media. *Proceedings of the First Workshop on Computational Approaches to Code Switching*, 13–23.
2. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 4171–4186.
4. Ghosh, S., Ghosh, S., & Chakraborty, T. (2017). A deep learning approach to code-mixed language identification. *Workshop on Computational Approaches to Linguistic Code Switching*, 45–54.
5. Gupta, P., Bali, K., Choudhury, M., & Sharma, V. (2012). Query expansion for mixed-script information retrieval. *WWW Conference*, 677–686.
6. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
7. Joshi, A., Prabhu, A., & Shrivastava, M. (2016). Sub-word level compositions for sentiment analysis of Hindi-English code mixed text. *COLING*, 2482–2491.
8. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR*.
9. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ICLR Workshop*.
10. Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. *ACL*, 1715–1725.
11. Solorio, T., et al. (2014). Overview for the first shared task on language identification in code-switched data. *Workshop on Code Switching*, 62–72.
12. Vaswani, A., et al. (2017). Attention is all you need. *NeurIPS*, 5998–6008.
13. Winata, G. I., Kampman, O., & Fung, P. (2019). Code-switched language models using neural-based synthetic data. *CoNLL*, 271–280.
14. Zhang, Y., & Yang, Q. (2018). An overview of multi-task learning. *National Science Review*, 5(1), 30–43.
15. Pratapa, A., et al. (2018). Language modeling for code-mixing: The role of linguistic theory. *ACL*.
16. Aguilar, G., et al. (2020). LinCE: A benchmark for code-switching evaluation. *LREC*.
17. Khanuja, S., et al. (2020). MuRIL: Multilingual representations for Indian languages. *arXiv*.
18. Conneau, A., et al. (2020). Cross-lingual representation learning at scale. *ACL*.
19. Xue, L., et al. (2021). mT5: Multilingual text-to-text transformer. *NAACL*.
20. Patel, N., & Sasikumar, M. (2021). Code-mixed Indian text: A survey. *ACM Computing Surveys*.
21. Kumar, A., & Singh, A. (2022). Deep learning for multilingual NLP: A review. *IEEE Access*.