TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

Advances and Challenges in Preprocessing Hindi–English Code-Mixed Text for Multilingual NLP

Shruti Gupta[a] , Lakshya Srivastava [b] , Amit Srivastava[c] and Gaurvi Shukla [d]

[a,b] Scholar, Computer Science Department, National P.G. College, Lucknow, India
[c,d] Assistant Professor, Computer Science Department, National P.G. College, Lucknow, India
sg718317@gmail.com[a], lakshyasri09@gmail.com[b]

KEYWORD

Hinglish; Code-Mixed Text; Text Preprocessing; Language Identification; Transliteration

ABSTRACT

In social media and on-line communication, Hinglish is a code-mixed language between Hindi and English that is widely used in linguistically mixed areas like India. It is informally structured, it transliterates and regularly switches between languages, which poses considerable problems to natural language processing (NLP) systems. Hinglish may not be processed with the traditional preprocessing pipelines that are intended to process monolingual text. The current review offers an in-depth description of Hinglish text preprocessing and linguistic features of this language. It also talks about big datasets, benchmarks and most frequently used preprocessing algorithms like language identification, transliteration, token normalization and multilingual embeddings. The recent developments, such as contextual and code-mixed pretrained models are also mentioned. In spite of this, there are still concerns over data sparsity, annotation inconsistency, transliteration variability, and real-time processing. The paper also discusses the new areas of research, such as adaptive preprocessing systems and multiscript corpora. On the whole, this survey provides useful information on the existing developments and perspectives of strong and culturally sensitive multilingual NLP applications.

1. Introduction

The easy access to social media tools, messaging applications, and other user-generated online materials have resulted in a sudden spurt of code-mixed language whereby the user simply switches between two or more languages in the same sentence or discussion. It has gained a significant role in Natural Language Processing (NLP) as it is a popular and linguistically complex field of study [1]. Hinglish, also known as a combination of Hindi and English, is a type of code-mixing that is widely used in the Indian environment. Particularly common in unofficial internet spaces, including Twitter, Facebook, WhatsApp, and YouTube, users in these spaces prioritize speed, convenience, and self-expressions. Hinglish provides an opportunity to communicate and express culture easily, but it also poses important challenges to the NLP systems [4]. Hinglish is not only a complicated language with regards to word switching between Hindi and English. It usually incorporates deviances in grammar, word formation, sentence structure and pronunciation thus rendering automatic language processing challenging [15]. Hinglish text often includes Romanized Hindi, non-standard spellings, slang, abbreviations, emojis, and sudden changes of language in comparison to formal monolingual text. As an illustration, the same word in Hindi could have several different forms like; “mera” and “meraa”, whereas English words are commonly integrated into Hindi grammatical construction. These discrepancies pose severe problems to text normalization and preprocessing [13].

Corresponding Author: Shruti Gupta, Department of Computer Science, National P.G. College, Lucknow, India
Email: sg718317@gmail.com

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

Preprocessing is an important step in any NLP pipeline, especially when training sentiment analysis, machine translation, text summarization, part-of-speech tagging, and named entity recognition [15]. Nonetheless, the standard preprocessing methods that are meant to work with monolingual text, e.g., tokenization, stop-word elimination, stemming, and lemmatization, tend to be unsuccessful with code mixed data. The second significant problem is the identification of the language of separate words or even sub-word units since spelling variants, borrowings, and colloquial manner of writing causes language boundaries to be unclear [16], [17]. To solve these issues, current studies have suggested preprocessing methods specific to Hinglish and other code-mixed languages. These are transliteration normalization, word level language tagging, spelling variation management and sub-word or context representation [9]. Advanced benchmark datasets and assessment systems like LinCE, GLUEcoS, L3Cube-HingCorpus or even the new COMI-LINGUA have also facilitated this further by now allowing comparable and uniform evaluation across studies [3]. Along with these, transformer-formed structures and multilingual pre-trained language models have enhanced the processing of code-mixed text, even though significant constraints are still evident [1], [14].

In spite of these developments, preprocessing of Hinglish text is an open research problem. Communication via online platforms is growing exponentially, users are developing new terms, unstructured shortcuts, and messages full of emojis. Such dynamic character demands flexible, adaptive and scalable preprocessing techniques. The proper comprehension of both linguistic features as well as the recent technological advancements is thus quite crucial to the construction of the accurate and context-sensitive NLP systems. The paper under review explores the major issues related to preprocessing Hinglish text and outlines the modern approaches, tools, datasets, and evaluation systems. Through the research analysis of the existent research, we also reveal the current limitations of the research, the new trends of the research and the prospective directions of future research in code-mixed NLP [1], [2].

2. Background/Related Work

2.1. Hinglish and Code-Mixed Language

The process of online communication has changed greatly over the last few years especially in the multilingual states like India. Code-mixed communication is becoming more and more common among users as two or more languages are mixed up in a sentence or conversation. Hinglish is one of the most common code-mixing types that are frequently used in social media, messaging apps, and forums on the internet, including WhatsApp, Twitter, Facebook, and YouTube. This increased use of Hinglish has also drawn the interest of Natural Language Processing (NLP) community whose multiple surveys and research on code-switching and multilingual text processing confirm this point of view [1], [2]. Although Hinglish allows to communicate effectively and to express culture, its informal and dynamic character poses a serious challenge to standard NLP systems.

2.2. Linguistic Characteristics of Hinglish

Hinglish has varied attributes of linguistic features, which make it complex. Hindi and English can be used in the same sentence where Hindi words are combined with English words like the word review in the sentence “Mujhe movie ka review bata do”, where English words are incorporated in the Hindi grammars. Also, Hindi is often spelled with the Roman writing system, rather than the Devanagari writing system; therefore, one word may end up spelled in different ways i.e. “mera”, “meraa” or “mra”. The English words are also widely modified to the grammar of Hindi, and hybrid syntactic structures are produced. Earlier studies of part-of-speech tagging and code-switching prediction have demonstrated that these hybrid structures are challenging to the standard language models to process correctly [15]. All of these issues are enhanced by the prevalence of slang, abbreviations, emojis, and hashtags in the text of social media.

2.3. Preprocessing Challenges in Hinglish Text

Processing Problems in Hinglish Text. Considering the language variation of Hinglish, preprocessing is an important method to make the NLP analysis effective. Conventional preprocessing methods such as tokenizing, stop-word eliminating, stemming and lemmatization methods are mostly monolingual text based and these techniques do not apply effectively on code-mixed data. Intricate preprocessing has thus been necessitated by word-level language recognition, spelling standardization and transliteration among scripts [7], [8], [13]. Research has also shown that correct identification of language at the word level or sub-word level is more difficult in Hinglish as a result of language switching, borrowed words and informal forms of writing [17]. Poor preprocessing may

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

hugely deteriorate the way downstream NLP tasks are performed, including sentiment analysis, machine translation, and named entity recognition.

2.4. Datasets and Benchmark Resources

In order to facilitate studies in Hinglish and other code-mixed languages, a number of datasets and benchmark frameworks have been created. GLUECoS and LinCE are resources that offer standard evaluation environments of multilingual and code-mixed NLP tasks [3], [4]. In the case of Hinglish-specific studies, databases such as L3Cube-HingCorpus have gained significant popularity as they provide annotated data-sets in language identification as well as other preprocessing activities. Moreover, HingBERT as a type of a language model, trained on Hinglish data, has shown a better performance than general-purpose multilingual models [5]. Other data sets are devoted to Hinglish text generation and evaluation, like HinGE, and normalization and preprocessing quality is also targeted by such tools as hinglishNorm.

2.5. Recent Advances and Remaining Challenges

In recent researches, there is an extension of the basic preprocessing and classification activities. A large-scale expert-annotations resource called the COMI-LINGUA has been offered to aid a variety of multilingual NLP activities such as language identification and named entity recognition [19]. Mixed-code machine translation Systems like CoMeT and MixMT have also been investigated to advance in machine translation of code-mixed text [9], [12]. Semi-supervised and data augmentation methods have been suggested to create Hinglish fake text to train the model. Such SentiMix evaluation campaigns have demonstrated the difficulties of sentiment analysis in Hinglish tweets, and the HinglishEval challenge has demonstrated the scarcity of useful resources to this language variety. Although they have done this, there are still a number of challenges that are not addressed. Variation of transliteration, inconsistency of spellings, and emojis, hash tags and abbreviations persist to complicate preprocessing [13]. Even though the use of multilingual pre-trained language models like mBERT and XLM-R has demonstrated encouraging outcomes, they remain unable to capture the linguistic patterns of code-mixed text in all aspects. In addition, the generalizability of the current methods is limited by the lack of big, heterogeneous, and high-quality datasets.

3. Challenges in Preprocessing Hinglish Text

One of the most difficult issues in Natural Language Processing (NLP) is preprocessing Hinglish, a code-mixed language that is the mixture of Hindi and English [1], [2]. As opposed to monolingual text, Hinglish is informal, very fluctuating, and ever-changing. It often consists of Romanized Hindi, English words with Hindi grammatical constructions, slang, emojis, and unusual spellings.

The following properties complicate the process of preprocessing in terms of basic steps, including tokenization, normalization, and language recognition, to a considerable extent compared with the monolingual case [3]. Figure 1 lists the key issues associated with preprocessing of Hinglish text and explains them below.

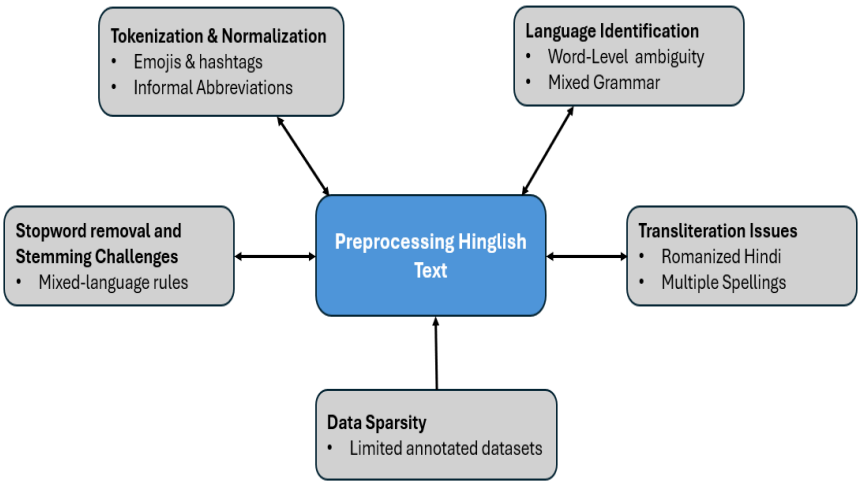

Figure 1. Major challenges in preprocessing code-mixed Hindi–English (Hinglish) text.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

3.1. Language Identification

Proper identification of words or better said token level language recognition is one of the main problems of preprocessing Hinglish [15]. Hindi and English words are often used interchangeably alongside each other and some tokens may be of either language based on use. As an example, the word “is” is given in form of a verb in English and transliteration of the Hindi word “this”. There are also phonetic scripts of the Hindi words in Roman script, which have many forms like “kya”, “kyaa”, and “kia”. There is also a common insertion of English words in Hindi syntax where the word is "movie ka review" thus making it more complex to classify. Any mistake in language recognition may have disastrous consequences on such downstream processes as sentiment analysis, named entity recognition and machine translation [11], [17].

3.2. Transliteration and Spelling Variation

Romanized Hindi is often included in Hinglish text, and this results in a great deal of spelling variation in the same lexical item [7], [13]. As an example, Hindi term “mera” (meaning mine) can be written either as “mera”, “meraa”, “mra” or “maira”. The spellings are also slightly altered intentionally by the user to bring about an emphasis like “haaan” or “haaaaan” rather than “haan”. To overcome such variants, it is necessary to utilize effective transliteration and normalization methods, which can be a blend of phonetic proximity, grammatical principles, and dictionaries as well as machine learning-based methods.

3.3. Tokenization Challenges

The Hinglish text tokenization is significantly more complicated than the standard English [5] or Hindi text tokenization [6]. Emojis can be very expressive in terms of emotional or semantic meaning and cannot be just stripped out in the preprocessing phase. Hashtags like #best movie ever, are concatenated words which need to be broken down into meaningful analysis. The use of repeated characters in words such as “cooool”, “yeesss” etc add more challenges to the boundary of tokens. Irregular use of punctuations, spacing and capitalization (as in informal online messages) may also be a source of confusion to traditional tokenization algorithms.

3.4. Normalization and Stopword Handling

Such normalization methods as lowercasing, removal of stopwords, stemming, and lemmatization are especially difficult with Hinglish text [7], [8]. Hindi or English stopword lists do not suffice and many words like “hai” or “is” need to be understood in the context. Lowercase can change the correct nouns and even delete stylistic effect employed by the user. Moreover, without any loss of valuable semantics, normalization needs to be able to deal with transliterated words, emojis, and informal symbols.

3.5. Data Sparsity and Limited Resources

One of the gaps in Hinglish NLP studies is the lack of large and quality, and diverse annotated datasets [5], [18]. Most of the existing corpora are either small, domain-specific or obsolete. Datasets with language labels at the token level, transliteration mappings or sentiment annotations are especially scarce. This scarcity does not allow building powerful models, and it does not allow them to generalize in other areas and styles of Hinglish text.

3.6. Informal and Dynamic Language Use

Hinglish is very dynamic particularly within social media setting [1]. New slanging words, shortenings, memes, and phrases are created very often, which results in an ever-growing vocabulary. Users tend to confuse languages in an unpredictable manner leading to many out of vocabulary words. In order to easily manage this dynamic nature, the preprocessing systems should be dynamic, and usually semi-supervised, unsupervised or continual learning methods are used [14].

3.7. Contextual and Multi-Sentence Challenges

Some preprocessing activities involve having context knowledge outside of individual tokens [9], [12]. Meaning or language of a word can easily vary due to the usage of other words as in the case school ka function where an English word is incorporated in the Hindi structure. There is also a possibility of code-mixing between sentences, which complicates activities like sentiment analysis, discourse understanding and machine translation.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

All in all, preprocessing Hinglish text is not an easy task because of its informal character, variability in spelling and transliteration, vocabulary development, scarce resources and language use that varies depending on the context [3],[6]. Studies have shown that hybrid systems that are a combination of rule-based, machine learning, deep learning, and multilingual embeddings are needed in the construction of robust, and versatile preprocessing pipelines. These challenges are important to effectively address so that proper and accurate downstream NLP applications can be done on Hinglish text.

4. Advances in Preprocessing Techniques

Recent Hinglish text preprocessing has gradually transitioned away from rule-based and handcrafted techniques toward less ad hoc and more systematic data-driven techniques, which can handle the dynamic and code-mixed character of Hindi-English text. Though there is a variety of techniques, the majority of preprocessing systems follow the typical pipeline, which includes collecting and cleaning data, language identification, transliteration, and tokenization, further feature representation, and embedding generation. High level strategies are then implemented including hybrid pipelines, synthetic data generation and self-supervised learning and this is then evaluated against benchmarks. Figure 2 illustrates a generalized preprocessing pipeline for Hinglish text, highlighting the key stages addressed by recent studies. The following discussion elaborates on the major advances corresponding to each stage.

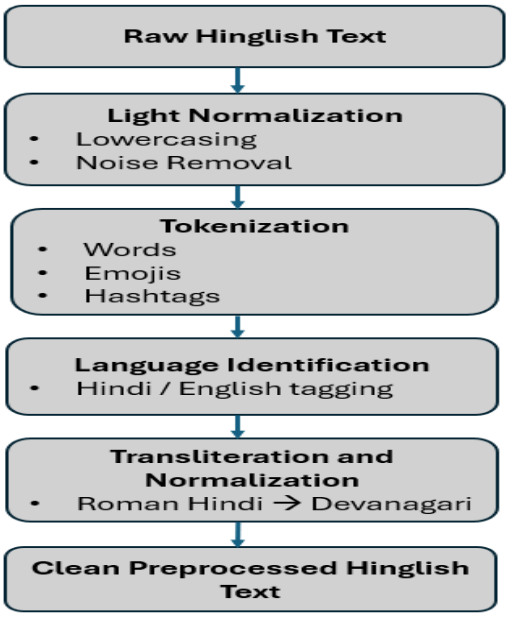

Figure 2. Generalized preprocessing pipeline for Hinglish text

The subsequent discussion reviews recent advances corresponding to different stages of the preprocessing pipeline.

4.1. Larger and Richer Datasets

The first step to better preprocessing is a good data way to start. Recently, big bodies of code-mixed Hindi-English text on the social media, forums, and online comments became accessible [5]. Such datasets may have labels like token-level language labels, transliteration mappings and normalized forms. The availability of richer and more realistic data enables models of normalization, transliteration, language identification, tokenization and segmentation to acquire authentic usage patterns in lieu of focusing on straightforward rules alone. These datasets are an excellent basis of any future improvement of preprocessing.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

4.2. Improved Language Identification

Correct word level language identification is also one of the most important preprocessing stages [15]. Transformer models consider the context around the tokens and not the tokens alone. Synthetic variations, such as alternative romanized spellings of Hindi words and fake code switching, are added to the training data to enhance model robustness to real social media text, e.g. mera, meraa, or mra. The classifications of language switches in the sentence can be done with precision by labeling the sequences at the token level instead of the sentence level [17], [18].

4.3. Transliteration and Spelling Normalization Using Neural Models

Previous methods were extremely dependent on rule-based transliteration and normalization. Existing systems use sequence-to-sequence context-sensitive models that handle sentences as a whole, and resolve ambiguous romanizations depending on contextual words [7], [13]. Large language models that are fine-tuned have been shown to be able to normalize variant spellings into canonical ones. Generative approaches suggest several possible normalizations of candidates and pick the most optimal one. These strategies deal better with misspellings, slang and long forms as compared to traditional rule systems.

4.4. Tokenization and Subword Modeling for Mixed-Script Text

Hinglish text is not easy to tokenize because of the presence of Romanized Hindi words, English words, hashtags, emojis, repeated letters, and odd punctuations. These improvements have led to training subword tokenizers on code-mixed corpora in order to include common Hindi affixes and English words. The tokenizers are either at a character level or a byte level to improve resistance to abnormal spellings and concatenated tokens. The hashtag scanning guidelines divide hashtags (e.g., Best Movie Ever #BestMovieEver), retain emojis, and standardize repeated letters or punctuations [9], [12].

4.5. Code-Mixed Specific Embeddings and Models

Instead of working with models only based on multilingual models such as mBERT, researchers are now training embeddings and models with large Hinglish corpora [5]. The mixed-language contexts, such as Romanized Hindi, are intuitively modeled in these models as opposed to generic multilingual models, which usually model these languages as either unknown or noisy. They form a strong basis of downstream activities like normalization, language identification and named entity recognition [14].

4.6. Hybrid Pipelines Combining Rules and Machine Learning

With advanced deep learning models, all-end-to-end methods can tend to fail in rare slang, memes, or highly noisy text [3]. A variety of pipelines today learn models on the rare cases combined with fast rule-based algorithms on common ones. Reliability is further enhanced by having fallback mechanisms like lexicon look up or manual inspection when the confidence is low. Such a tradeoff between interpretability and flexibility makes such a combination of rules and learning well suited to the real world.

4.7. Synthetic Data Augmentation and Dynamic Vocabulary Generation

Hinglish is colloquial and in a state of constant development. To solve this, preprocessing pipelines usually feature data augmentation strategies [6]. These strategies produce Romanized forms of Hindi words, imitate code switching in monolingual sentences, or employ the back-translation to produce more sentences with mixed languages. Augmentation assists models to extrapolate to the unseen or infrequent patterns and mitigate overfitting.

4.8. Self-Supervised, Multi-Task, and Unsupervised Learning

Newer methods are based on self-supervised, multi-task, and unsupervised learning because there is a lack of labeled data [7]. Models can be used to predict both masked tokens and language labels and normalized forms. Contrastive learning aligns the different Romanized spellings of the same word in embedding space, which creates a stronger representation. Unsupervised transliteration induction maps variant forms to canonical forms enhancing preprocessing pipelines despite a small amount of annotated data [14].

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

4.9. Better Evaluation Frameworks and Benchmarks

Assessment methods have also become stricter and are token-level accuracy-based language identification and F1 as well as normalization metrics founded on the edit-distance [3]. Contemporary data include various areas and multi-sentence code mixing, which allows one to compare models fairly. The instruments that quantify the extent of code-mixing assist in evaluating the performance of the model on sentences of different degrees in language mixture [11].

4.10. Real-World Preprocessing Pipelines

In real-world applications, pipelines can include multiple preprocessing steps, usually executed according to the needs of the final task or task group that is undertaken in data processing. The existing pipelines are a combination of light normalization, subword tokenization, token-level language tagging, transliteration of Romanized Hindi tokens, task-specific adapters and fallback mechanisms [5], [6]. Modular designs make preprocessing enhance itself on the basis of tasks down the line and be robust to the variety of types of social media content.

5. Evaluation Metrics and Datasets

To determine the reliability and capability of NLP models to generalize between the different types of data, it is necessary to evaluate preprocessing techniques of Hinglish, a blend of Hindi and English. Hinglish poses special consideration, such as code switching, different spelling, transliteration and informal grammar that makes conventional monolingual evaluation techniques to be inadequate. Researchers have over the years come up with special datasets and measures of evaluation that are indicative of the complexity of code-mixed text especially on activities like language identification, part-of-speech tagging, sentiment analysis and machine translation [1], [2].

5.1. Datasets for Hinglish NLP

High quality datasets have been a major motivation to the development of preprocessing methods and subsequent NLP applications with Hinglish. Table 1 provides key Hinglish code-mixed datasets between 2021 and 2025, with each applied in studies to perform certain tasks including code-mix generation, language identification, multi-task NLP, and machine translation.

Table 1. Key Verified Hinglish / Code-Mixed Datasets (2021–2025)

Dataset Name	Year	Domain / Source	Task Type	Size / Instances	Key Features / Notes
HinGE	2021	Web / general text	Generation & Evaluation	Varies	Human and algorithm-generated Hinglish text for generation & evaluation research [6].
L3Cube-HingCorpus	2022	Twitter	LID, POS, Sentiment, NER	~52.9 M sentences	Large real code-mixed Hinglish corpus used for multiple NLP tasks [5].
MUTANT	2023	Political & news articles	Multi-sentential code-mixed NLP	~84.9 K	Multi-sentential Hinglish dataset for long-sequence tasks [12].
Hinglish Language Corpus	2024	Mixed sources (synthetic + manually written)	Corpus for diverse NLP models	Varies	Blends synthetically generated and manually written Hinglish for general NLP research [8], [9].

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

COMI-LINGUA	2025	Mixed social + web	Multi-task NLP: LID, POS, NER, MT	~125 K+ annotated instances	Expert-annotated large multi-task code-mixed dataset in Roman & Devanagari scripts [19].

5.2. Evaluation Metrics for Hinglish Preprocessing

In the evaluation of preprocessing on code-mixed text, it is important to select the right evaluation metrics. The conventional measures like the total accuracy or simple F1-score cannot be used properly because of the fluctuating length of the tokens, unclear language labels, and irregular spellings [13],[15]. The measures that researchers have used are usually task-specific and are adapted to specific preprocessing tasks:

- Token-Level Metrics: Precision, Recall and F1-score to measure the effectiveness of the token classification as Hindi, English or mixed [15], [16].
- Sequence-Level Metrics: BLEU and Word Error Rate (WER) assess the proximity of normalized or transliterated output to human resources [8].
- Semantic / Contextual Metrics: Semantic preservation post-preprocessing is verified by Cosine similarity or sentence embedding alignment (e.g., with Sentence-BERT) [14].
- Task-Oriented Metrics: In downstream tasks, e.g. sentiment analysis, offensive content detection, or NER, Macro F1, Weighted Accuracy, and confusion matrices can be used to identify fine vehicle workings and class imbalances [6],[18].

5.3. Discussion

Hinglish preprocessing has gone beyond basic token-based evaluation to contextual and multi-task evaluation which is a combination of linguistic and semantic measurements [1], [2]. The growing availability of multilingual benchmark datasets, such as large corpora and multi-task annotated resources has contributed to the comparability and the reproducibility of the difference in preprocessing methods.
Nevertheless, there are still difficulties such as:

- Lack of uniformity in annotation of datasets.
- Absence of correspondence between datasets, cross-dataset evaluation is not easily achievable.

The next steps that can be taken in future research are to develop standardized evaluation systems that combine automatic measures with human judgment that produce reliable and generalizable preprocessing pipelines of Hinglish text [18].

6. Applications of Pre-processed Hinglish Text

Pre-processing Hinglish, the mixture of Hindi and English is an important step towards crafting prosperous NLP systems within the multilingual Indian digital ecosystem. The mixture of languages on social media, messaging applications, and online platforms often result in noisy and undefined text, as the user mixes languages. This text is standardized through preprocessing, such that downstream applications can use such code-mixed inputs successfully [1], [2].

6.1. Sentiment Analysis and Opinion Mining

Pretrained Hinglish enhances sentiment analysis on social networking sites such as twitter, YouTube, and Facebook by understanding transliterated words, emojis, and language mixed. Such context-aware embeddings as BERT-based Hinglish models can allow us to gain a subtle insight into words such as “mast” or “bakwaas” [14].

6.2. Conversational Agents and Chatbots

Preprocessing is also an advantage of chatbots and virtual assistants since it aids in language boundary detection, token normalization, and user intent in queries such as “Cab book kar do please”. This facilitates natural, bilingual customer support, e-commerce and voice-based applications [4], [18].

6.3. Machine Translation and Multilingual Processing

Not only does Preprocessing normalize Romanized Hindi and tags languages but it also structures sentences and enhances the quality of machine translations on mixed inputs like: “Kal office me meeting hai na”. It is also used to improve multilingual transformers such as mBERT and XLM-R [5].

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

6.4. Named Entity Recognition and Information Extraction

NER systems are also based on preprocessing to allow correct recognition of names, places, and organizations in Hinglish to deal with both code-switches and transliteration differences, such as between “Rama” and “Ramaa” [6].

6.5. Text Normalization, Spell Correction, and Search Optimization

Preprocessing normalizes a variant of the spelling, as well as the token forms which are required in search engines, predictive keyboards, and content search. Other queries such as, “mera phone kho gaya hai” can give relevant answers in both languages [10], [14].

6.6. Speech and Voice-Based Applications

Preprocessing is an advantage of ASR and TTS systems, in that the mixed-language tokens are always mapped and language alternations are detected to better recognize and provide a natural result to utterances such as “Alexa, AC band kar do please” [17].

6.7. Socio-Cultural and Linguistic Research

Hinglish Preprocessed Hinglish facilitates research on bilingual communication, code-switching behavior, and hybrid forms, as well as inclusive AI and culturally sensitive NLP research [19].

To conclude, the preprocessing of Hinglish is the foundation of almost every bilingual NLP application in India and provides effective, precise, and culturally aware treatment of code-mixed text.

7. Open Challenges and Future Directions

Although there have been remarkable advances in the preprocessing of Hinglish text several issues still restrict the functionality of NLP systems. Hinglish is also very dynamic, socially and structurally heterogeneous and thus the old preprocessing pipelines can hardly generalize across domains and time [1], [2]. Another significant problem is annotating data and the quality. The current datasets tend to be platform-specific and not annotated in various standards, which limits cross-dataset testing and model resilience. Production of large and multi-domain corpora that have common guidelines is still an open requirement.

The preprocessing is also complicated by the fast development of code-mixing. New slang, transliteration patterns, and informal expressions are created on a regular basis and make models that are trained on fixed data obsolete [2], [14]. The adaptive and self-supervised learning of the future systems must be based on the necessity of keeping up with linguistic changes. Scription variability and ambiguity of transliteration - in particular, between Roman and Devanagari scripts - continues to be a problem in tokenization and normalization, which in turn impact subsequent applications, including sentiment analysis and translations. Bidirectional and context-aware transliteration models provide the way forward.

Furthermore, the majority of preprocessing methods do not work on online text, and cannot be used in real-time tasks like chatbots and virtual assistants [10],[12]. Conversational AI requires lightweight and low-latency preprocessing pipelines. Lastly, there is less exploration of issues of evaluation and fairness. Corruption Standard automatic measures tend to be insensitive to semantic retention and appropriate use of cultural values, and current models can be biased to promote English dominant or urban forms of Hinglish [13], [18]. Human evaluation and fairness-conscious modeling practices need to be incorporated in future research. Altogether, the development of Hinglish preprocessing needs adaptive, context-sensitive, and accommodative systems, backed up by standardized data and sound assessment guidelines to provide stable results in real-life multi lingual situations.

8. Conclusion

The preprocessing of code-mixed Hindi-English (Hinglish) text is a complicated but necessary step towards successful natural language processing within the multilingual digital world of India. More recent studies have seen significant advances in normalization, transliteration, and token-level language recognition and the development of benchmark datasets, including GLUECoS and LinCE, which allows considerations of code switching, variation in scripts and informal language. Nonetheless, it is still subject to certain endemic problems such as the fast-changing slang, regional differences, uneven transliteration, and the lack of standardized testing procedures. All NLP applications, such as sentiment analysis, machine translators, chatbots, and speech systems, all rely on accurate preprocessing. Poor management of code-mixed text may result in misunderstanding the user intent, biased results, and contextual, cultural losses.

Future studies are advised to focus on the adaptive and inclusive designs, including the construction of large, multi-domain Hinglish corpora, incorporating multimodal feedback, e.g. emojis and speech, and creating low-latency models to be used in real-time. Additional support of preprocessing frameworks will be achieved by

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.04 (2025)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.44005

incorporating fairness-aware training, bias assessment, and human-in-the-loop assessment. On the whole, Hinglish preprocessing is a potential field of innovation, though much significant progress has been made so far, which can be used to serve culturally sensitive and powerful multilingual NLP systems.

References

[1]. Winata, Genta Indra, Alham Fikri Aji, Zheng-Xin Yong, and Thamar Solorio. "The Decades Progress on Code-Switching Research in NLP: A Systematic Survey on Trends and Challenges." *Findings of ACL*, 2023.
[2]. Doğruöz, A. Seçil, et al. "A Survey of Code-Switching: Linguistic and Social Perspectives for Language Technologies." *ACL Long*, 2021.
[3]. Khanuja, Simran, et al. "GLUECoS: An Evaluation Benchmark for Code-Switched NLP." *Proceedings of ACL*, 2020.
[4]. Aguilar, Gustavo, Sudipta Kar, and Thamar Solorio. "LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation." *LREC*, 2020.
[5]. Nayak, Ravindra, and Raviraj Joshi. "L3Cube-HingCorpus and HingBERT: A Code-Mixed Hindi-English Dataset and BERT Language Models." *arXiv preprint arXiv:2204.08398*, 2022.
[6]. Srivastava, Vivek, and Mayank Singh. "HinGE: A Dataset for Generation and Evaluation of Code-Mixed Hinglish Text." *Eval4NLP Workshop*, 2021.
[7]. Makhija, Piyush, Ankit Kumar, and Anuj Gupta. "hinglishNorm — A Corpus of Hindi-English Code-Mixed Sentences for Text Normalization." *arXiv preprint arXiv:2010.08974*, 2020.
[8]. Parikh, Dwija, and Thamar Solorio. "Normalization and Back-Transliteration for Code-Switched Data." *CALCS@NAACL*, 2021.
[9]. Gautam, Devansh, et al. "CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences." *CALCS*, 2021.
[10]. Gupta, Deepak, Asif Ekbal, and Pushpak Bhattacharyya. "A Semi-supervised Approach to Generate the Code-Mixed Text using Pre-trained Encoder and Transfer Learning." *Findings of EMNLP*, 2020.
[11]. Patwa, Parth, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. "SemEval-2020 Task 9: Overview of Sentiment Analysis of Code-Mixed Tweets (SentiMix 2020)." *SemEval Proceedings*, 2020.
[12]. Laskar, Sahinur Rahman, Rahul Singh, Shyambabu Pandey, Riyanka Manna, Partha Pakray, and Sivaji Bandyopadhyay. "CNLP-NITS-PP at MixMT 2022: Hinglish–English Code-Mixed Machine Translation." *WMT22*, 2022.
[13]. Yadav, K., et al. "Normalization of Spelling Variations in Code-Mixed Data." *ICON 2022*, 2022.
[14]. Zhang, Ruochen, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Indra Winata, and Alham Fikri Aji. "Multilingual Large Language Models Are Not (Yet) Code-Switchers." *EMNLP*, 2023.
[15]. Vyas, Yogarshi, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. "POS Tagging of English-Hindi Code-Mixed Social Media Content." *EMNLP*, 2014.
[16]. Solorio, Thamar, and Yang Liu. "Learning to Predict Code-Switching Points." *EMNLP*, 2008.
[17]. Burchell, Laurie, Alexandra Birch, Robert P. Thompson, and Kenneth Heafield. "Code-switched Language Identification Is Harder Than You Think." *University of Edinburgh Research Report*, 2024.
[18]. Guha, Prantik, Rudra Dhar, and Dipankar Das. "JU_NLP at HinglishEval: Quality Evaluation of the Low-Resource Code-Mixed Hinglish Text." *INLG 2022 Generation Challenge*, 2022.
[19]. Sheth, Rajvee, Himanshu Beniwal, and Mayank Singh. "COMI-LINGUA: Expert Annotated Large-Scale Dataset for Multitask NLP in Hindi-English Code-Mixing." *arXiv preprint arXiv:2503.21670*, 2025.