PHINC: A Parallel Hinglish Social Media Code-Mixed Corpus for Machine Translation

Vivek Srivastava*
TCS Research and Innovation
Pune, India
srivastava.vivek2@tcs.com

Mayank Singh
Indian Institute of Technology, Gandhinagar
Gujarat, India
singh.mayank@iitgn.ac.in

Abstract

Code-mixing is the phenomenon of using more than one language in a sentence. In the multilingual communities, it is a very frequently observed pattern of communication on social media platforms. Flexibility to use multiple languages in one text message might help to communicate efficiently with the target audience. But, the noisy user-generated code-mixed text adds to the challenge of processing and understanding natural language to a much larger extent. Machine translation from monolingual source to the target language is a well-studied research problem. Here, we demonstrate that widely popular and sophisticated translation systems such as Google Translate fail at times to translate code-mixed text effectively. To address this challenge, we present a parallel corpus of the 13,738 code-mixed Hindi-English sentences and their corresponding human translation in English. In addition, we also propose a translation pipeline build on top of Google Translate. The evaluation of the proposed pipeline on $PHINC$ demonstrates an increase in the performance of the underlying system. With minimal effort, we can extend the dataset and the proposed approach to other code-mixing language pairs.

1 Introduction

Code-mixing is the phenomenon of switching between two or more languages by the speaker in a single sentence of a text or speech. It is a frequently observed pattern of communication in linguistically diverse countries such as India with 23 official languages and 122 major languages. With more than 300 million native speakers each, English and Hindi are among the top five most frequently used languages across the world. With the increase in the number of English speakers in Hindi speaking communities in India, the popularity of *Hinglish* (code-mixing in English-Hindi languages) is seeking a boom. Lambert (2018) first introduced the word Hinglish in 1967. David Crystal (Baldauf, 2004) projected in 2004 that the number of Hinglish speakers may soon outrun the number of native English speakers in the world. Other than Hinglish, multiple other bilingual code-mixed languages are popular in multilingual communities in India, such as Bengali-English, Telugu-English, etc. Lack of a standard for writing code-mixed text presents several challenges (see Section 2 for details) to natural language understanding tasks. Due to the source of origin (social media, online gaming, etc.), the code-mixed text is inherently noisy. We frequently observe code-mixing on social media platforms such as Twitter, Facebook, etc., in contrast to the formal literary sources such as books, poems, and newspapers. We, therefore, use social media platforms like Twitter and Facebook as the primary data source for our purpose.

The recent thrust on user engagement on social media platforms has led to several research directions, particularly in resource-constraint noisy user-generated content. Barman et al. (2014) discussed the language identification task for the code-mixed data involving Bengali-Hindi-English. Das and Gambäck (2014) presented various techniques to identify languages at the token-level for the Bengali-English and Hindi-English code-mixed corpus. Singh et al. (2018) discussed various techniques to identify the named-entities in the code-mixed Hindi-English corpora consisting of 3,638 tweets. Vyas et al. (2014) proposed various experiment to identify POS tags of the 1,062 code-mixed Hindi-English Facebook posts. They collected data from three popular celebrity Facebook public pages of Mr. Amitabh Bachchan (well-known actor), Mr. Shahrukh Khan (well-known actor), and Mr. Narendra Modi (current Indian Prime Minister). Besides, they leverage the BBC Hindi

*Work done during author's stay at IIT Gandhinagar

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 41–49
Online, Nov 19, 2020.

news articles. Sinha and Thakur (2005) presented a rule-based machine translation system to translate the code-mixed Hindi-English sentence to monolingual Hindi and English forms. Khanuja et al. (2020) presented an evaluation benchmark for the two code-mixed language pairs (English-Hindi and English-Spanish). The proposed evaluation benchmark has six NLP tasks, i.e., language identification, POS tagging, named entity recognition, sentiment analysis, question answering, and natural language inference. These tasks have been part of the recently shared tasks co-located with various NLP conferences or the latest research works. Even so, it presents two significant challenges and opportunities. First, most of the datasets available for various tasks are significantly less extensive to build robust standalone systems. Second, the comparatively less studied task for the code-mixed machine translation presents an opportunity to build datasets and translation systems. Dhar et al. (2018) propose a machine translation augmentation pipeline to use on top of the standard machine translation systems. They also create a parallel corpus of 6,096 English-Hindi code-mixed sentences and their corresponding translation in English.

In this paper, we present a good quality large-scale parallel corpus[1] for code-mixed English-Hindi noisy social media text messages. The main contributions are:

- We present a parallel corpus of 13,738 Hindi-English code-mixed sentences and their corresponding English translations by the human annotators.
- We discuss various challenges faced by machine translation systems in translating code-mixed sentences. Translation systems addressing these challenges could help mitigate the limitations of these systems.
- As a baseline, we propose a translation pipeline and compare the results with two widely popular translation systems (Google Translate and Bing Translate) on various evaluation metrics.
- We also discuss various limitations of the corpus and the research opportunities.

2 Code-Mixing and Challenges in Machine Translation

Code-mixing is the informal style of communication where words from two (in general) or more languages are part of the same utterance of a text or speech. An example code-mixed Hinglish sentence is, *"Hamare paas fully autonomous vaahan hai"*. This style of writing presents several challenges to almost all monolingual natural language processing tasks such as sentiment analysis, POS tagging, dependency parsing, etc. The widely-used machine translation systems, e.g., Google Translate, Bing Translate, etc., perform reasonably well on the monolingual translation task, but they fail to perform well on the code-mixed data (see Section 4 for details). We identify six potential causes for the failure of the standard machine translations systems on the code-mixed text are:

- **C1 (Ambiguity in language identification)**: Hindi words written in the Roman script present some significant challenges to identify the language of the text at the token level. Words like *'is', 'me', 'to'*, exists in both Hindi and English, leading to ambiguity in classification as English and Hindi without proper knowledge of context. Similarly, hashtags are often used on social media platforms, and code-mixed hashtags make it challenging to identify the boundaries of code-switching.
- **C2 (Spelling variations)**: Romanized Hindi also presents a challenge with no standard spelling of the words. Various spellings for the same word is used based on the user's pronunciation of the word, emotions, etc. For example, *'jaldi', 'jldi'*, and *"jldiii',* are some of variations for the word *'hurry'* in English. At times, people use repeated instances of some particular character to emphasize emotion, such as in *'jaldiii'*.
- **C3 (Named entity recognition)**: Recognition of named entities in the code-mixed data is also a challenging task. E.g., *'Bhartiya Janta Party'* is a code-mixed named entity (name of a political party in India). In translation, the unrecognized code-mixed entity might make the translation semantically incorrect.
- **C4 (Informal style of writing)**: We largely witness an informal style of writing on social media platforms. At times, we do not follow the standard rules of sentence structure on these platforms. This presents a challenge to translate the sentence in a monolingual style where we need the formal sentence structure for semantic correctness. For example, *'Sad kabhi dekha h usko.. me never'*, when translates to English becomes, *'Have you ever seen her sad? I have never seen her sad'*.
- **C5 (Misplaced/skipped punctuation)**: In

[1] https://bit.ly/36qkfkn

the informal writing style on social media platforms, punctuations are usually skipped, misplaced, or repeatedly used to express an opinion, and that makes it difficult for the machine translation system to translate such sentences. For example, *'Aap kb se cricket khelne lage..never saw u bfr'* misses a question mark(?) apart from other necessary modifications to make the structure of the sentence correct.
- **C6 (Missing context)**: Lack of knowledge of the context makes the machine translation task significantly difficult and challenging. Hidden sarcasm might get unnoticed while translating the sentence with missing context. For example, *'Note kr lijiye.. Bandi chal rahi h'* is a code-mixed sentence, and demonetization (*'notebandi'*) is the hidden context.

Figure 1 shows three example code-mixed Hinglish sentences and the corresponding translations by Google Translate and the human annotator. In all the examples, we observe various associated challenges (C1 through C6) with an effective translation by Google Translate.

We posit that the above challenges can be addressed to a large extent with the higher availability of a good-quality, manually annotated parallel corpus. However, as discussed in the previous section, the only available code-mixed Hinglish dataset (Dhar et al., 2018) is significantly small and less topically diverse. Some of the major differences with the previous work (Dhar et al., 2018) (*'PW'*, hereafter) are:

- **Spelling variations:** The annotation policy in our experiment (see Section 3 for details) explicitly ask the annotators to use the correct spellings in the translated sentences. E.g., the annotators provide the correct spelling for the words *'u'*, *'coz'*, and *'plz'* as *'you'*, *'because'*, and *'please'*, respectively. In PW, we observe large traces of incorrect spellings of the words in the translated sentence, such as 78 instances of the word *u*, 37 instances of the word *pls*, and 83 instances of the word *plz*.
- **Short sentences:** We remove the sentences that are less than five tokens. It helps to remove the monolingual sentences or sentences with less code-mixing. In PW, we find 747 (12.25%) sentences with length less than or equal to 3 tokens and 1,537 (25.21%) sentences with length less than or equal to 5 tokens.
- **Ambiguous sentences:** We refrain annotators to provide translation for the ambiguous sentences. In PW, we observe a few ambiguous code-mixed sentences and their corresponding translations. E.g., *"Tamil teri yadda Nai .. Har pal Teri yadda yadda wich h tu Tamil kaha aya game me ?"* is a code-mixed sentence with the English translation *"you don't rememeber Tamil .. every moment your memory memory which you in tamil where is it in game?"*.
- **Abusive sentences:** We prefilter the abusive sentences as well as refrain the annotators to translate them. In PW, we observe multiple sentences with abusive words.
- **English sentences:** We refrain the annotators to translate the sentences already in the English language. In PW, we find multiple instances of sentences in the code-mixed data which are already in English. For eg., *"my salman khan"*, *"luv u salman khan"*, *"Hallo salman sir"*, etc.

Example I

SENTENCE: Phone ka wallpaper dekhte dekhte zindagi kat rahi hai.
GOOGLE TRANSLATION: Life is cut off while watching the wallpaper of the phone.
HUMAN TRANSLATION: I'm spending my life seeing my phones wallpaper
ASSOCIATED CHALLENGES: C4 and C6

Example II

SENTENCE: Is shaher ko ye Hua kya hai.. Kahi rakh hai to kahi dhua dhua.. Play interrupted due to bad weather
GOOGLE TRANSLATION: What has happened to this city .. If there is smoke somewhere, then smoke somewhere .. Play interrupted payable then bad weather
HUMAN TRANSLATION: What has happened to this city. there is ash and smoke everywhere. play interrupted due to bad weather
ASSOCIATED CHALLENGES: C1, C4, and C5

Example III

SENTENCE: Bhai IIT wale hai pehle relationship toh bane laundon ki, break up par nacha rahe ho.
GOOGLE TRANSLATION: Brother-in-law is the first relationship to be made of laundries, you are dancing on the brake sub.
HUMAN TRANSLATION: Brother, you are an IITian. First get in to a relation. Then you can worry about break up.
ASSOCIATED CHALLENGES: C1, C2, C3, and C4

Figure 1: Comparison of translation of code-mixed sentences by Google translate and human annotators. The ineffective translation by Google Translate has various associated challenges.

3 Dataset

In India, Hinglish is a commonly observed pattern of communication on various platforms such as social media, online gaming, product reviews, discussion forums, etc. As outlined in the previous sections, multiple works have explored the various nuances of the code-mixed Hinglish text, such as language identification, sentiment analysis, etc. However, curating the code-mixed Hinglish dataset for these tasks requires a significant amount of human efforts due to the identification and filtering of noise from the useful content. In this work, we initially curated Hinglish sentences from six already existing works ((Singh et al., 2018), (Swami et al., 2018), (Joshi et al., 2016), (Barman et al., 2014), (Vrishank Shete and Mittal, 2016), and (Khandelwal, 2018)). One major advantage of using these datasets is the availability of high-quality code-mixed sentences without considerable manual filtering. Also, it offers diversity in terms of the source of the data collection as the major social networking platforms (Twitter and Facebook) are present. Additionally, the proposed curation process mitigates the topical bias, as we consider multiple topics in social-media discussions. Table 1 shows the statistics of the previous code-mixed datasets and $PHINC$. We select these datasets across various tasks, platforms, and topics/focus areas.

3.1 Description, Collection, and Pre-processing

We collect a total of 52,234 Hinglish sentences from multiple sources, as described above. We then shuffle, pre-process, and share these sentences with the annotators to provide the corresponding English translation. The script used in writing each sentence in the corpus is Roman. Pre-processing of the dataset involves the following steps:

- We remove sentences with less than five or more than 40 tokens. We introduce the upper limit on the sentence length to speed up the annotation process.
- We remove sentences having a percentage of out of vocabulary (OOV) words less than 50% or more than 90%. Lower limit (i.e., 50%) helps to filter out the sentences with the majority of English words whereas the upper limit (i.e., 90%) filter out the sentences containing a high percentage of Hindi words. We consider alphanumeric tokens as part of the vocabulary. We are using the English dictionary of the Natural Language Toolkit (NLTK) to identify OOV.
- We filter the sentences containing abusive words in English or the Romanized Hindi.

After pre-processing, we obtain a total of 25,346 code-mixed sentences.

3.2 Annotation

The objective of the annotation process is to produce the English translation of the corresponding code-mixed Hinglish sentence. We employ 54 annotators in the annotation task. Each annotator has expert level proficiency in writing, speaking, and understanding English and Hindi languages. We assign randomly selected 400 unique samples to each annotator, and the annotator has to provide the translation of each sentence in English. Each sentence in the final dataset is annotated by a single annotator. We provide a set of guidelines for each annotator for the annotation task. The annotation guidelines are listed below:

- **Special characters and emoticons:** Use the best understanding to include or skip these symbols and characters in the translated English sentences.
- **URLs, mentions, and hashtags:** Keep the same URLs, mentions, and hashtags in the translated sentence.
- **Incorrect spellings (u, hm, pls, coz, etc.)**: Translated sentence should have the correct spelling for each word.
- **Lower casing:** Write the translated sentence in lowercase.
- **Proper English sentence:** If the input sentence is already in English and also grammatically correct with no spelling mistakes, then its translation will only be "&" (without quotes). E.g., "I can translate the sentence quickly", do not require any modification.
- **Ambiguous sentence:** Do not translate an ambiguous sentence. If the sentence is unclear to translate in English, mark it as "#" (without quotes).
- **Abusive words:** Do not translate sentences containing abusive/cuss words. Mark it as "#" (without quotes).

We provide the same label to ambiguous and abusive sentences because, at times, the annotator is unaware of the abusive word used in the sentence, and the sentence appears ambiguous. Post annotation, we obtain 21,597 sentences. It also includes sentences that are refrained from the translation (i.e., proper English sentence, ambiguous sentence, and sen-

Dataset Source	Task	Platform	Dataset Size	Topics/Focus areas
Singh et al. (2018)	Named-entity recognition	Twitter	3,638	Politics, social events, sports, etc.
Swami et al. (2018)	Sarcasm detection	Twitter	5,250	Bollywood, cricket, and politics
Joshi et al. (2016)	Sentiment analysis	Facebook	3,879	Bollywood and politics
Barman et al. (2014)	Language identification	Facebook	771	Not available
Vrishank Shete and Mittal (2016)	Sentiment analysis	Facebook	7,663	Politics, news articles, etc.
Khandelwal (2018)	Humor detection	Twitter	31,033	Not available
$PHINC$	Machine translation	Twitter & Facebook	13,738	Sports, politics, Bollywood, etc.

Table 1: Statistics of the previous Hinglish code-mixed datasets and $PHINC$. Dataset size shows the number of sentences in the dataset. We select the topics/focus area of the dataset as mentioned in the corresponding dataset source.

tences containing abusive words). We then filter sentences with no human translation. Finally, we obtain 13,738 code-mixed sentences with the corresponding English translation.

Figure 2 shows three examples of the sentences that come under the refrain category of sentences for translation. Example I is a proper English sentence and requires no translation. The sentence in example II contains the abusive word, whereas the sentence in example III is ambiguous to translate. Figure 3 shows two code-mixed sentences and their corresponding translation in the corpus. Example I show a high-quality translation by the annotator that does not require any changes, whereas the translation in the example II is of poor quality, as it is semantically incorrect and requires modification. Note that we are not making any changes to the poor quality translation of the code-mixed sentences. We discuss the quality of translations in detail in Section 3.3.

Example I

CODE-MIXED SENTENCE: RT: Today is the birth anniversary of Maharana Pratap, whose bravery & indomitable spirit doesn't fail to inspire even today.
LABEL: &
REASON FOR NO TRANSLATION: Sentence already in English

Example II

CODE-MIXED SENTENCE: sach bolu ? Aap Cuss hai
LABEL: #
REASON FOR NO TRANSLATION: Presence of abusive/cuss word in sentence.

Example III

CODE-MIXED SENTENCE: yuhi kat jaayega safar sath tweetne se , ki manzil aayegi nazar sath tweetne se . Hum raahi Twitter ke
LABEL: #
REASON FOR NO TRANSLATION: Ambiguous sentence.

Figure 2: Example of the code-mixed sentences with no translation by the annotators. We replace the cuss word in Example II with the word "Cuss".

3.3 Exploratory Analysis

In this section, we conduct the exploratory analysis of the sentence pairs in the corpus.

1. **Out of vocabulary (OOV) words**: Figure 4 shows the distribution of the OOV words in the code-mixed sentences. We are using the NLTK English dictionary for this study. Apart from the Romanized Hindi words, hashtags and mentions also fall into the category of OOV words. We consider alphanumeric tokens as part of the vocabulary. The code-mixed dataset contains sentences with the percentage of OOV words greater than 50% and less than 90%. A large number of sentences comprise a higher proportion of OOV words, illustrate the non-standard writing style of the users while using code-mixed languages on various platforms. Also, on manual inspection, we observe that while writing Hinglish, people often use Hindi as the matrix language and embed the words from the English language. We posit that usage of a high percentage of OOV words makes the text noisy and challenging to perform various natural language processing tasks such as named-entity recognition, machine translation, sentiment analysis, etc.
2. **Degree of Code-mixing**: To evaluate the degree of code-mixing in the corpus, we use Code-Mixing Index (CMI) (Das and Gambäck, 2014). CMI value range from 0 to 100. A value close to 0 suggests monolingualism in the corpus, whereas high CMI values indicate a high degree of code-mixing. To calculate the value of CMI, we randomly sample 100 code-mixed sentences from the corpus and annotate them at the token level with three language tags English, Hindi, and others. The CMI cal-

Example I

CODE-MIXED SENTENCE: Thnks buds! Kabhi kabhi aajate hai achhe photos
ENGLISH TRANSLATION: Thank you buddy, sometime good photos are captured.
REQUIRE CHANGES IN THE ENGLISH TRANSLATION?: No

Example II

CODE-MIXED SENTENCE: Australia ke saath abhi jeete nahi hai, magar NZ ke saath final kaise jeetenge iss soch mein bhartiya yuvak on twitter.
ENGLISH TRANSLATION: Indian youth on twitter thinking that - We have not won against Australia yet, but how would we win final with NZ?
REQUIRE CHANGES IN THE ENGLISH TRANSLATION?: Yes

Figure 3: Example translation of the code-mixed sentences in the corpus. The annotators provide translations to the code-mixed sentences. A change in the translation is required if the translation is semantically incorrect.

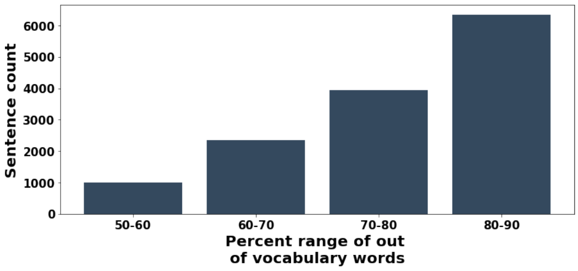

Figure 4: Distribution of out of vocabulary words in the code-mixed sentences. Large number of sentences comprise higher proportion of OOV words.

culated for this set of sentences is 75.76, which indicates a significantly higher usage of code-mixing in the text.

3. **Frequent words**: Figure 5 shows the word clouds of the code-mixed and English translated sentences. It is evident from the word cloud that words from multiple domains such as politics, entertainment, sports, etc., are very frequently used. The list of top-15 most occurring words having character length greater than six[2] are *salman, chahiye, alllahdin, krishna, meetuunnglee, atheist, kejriwal, tomorrow, mahashivratri, pakistan, narendramodi, tumhare, shaadi, gandhi,* and *indvspak*. This list contain words from multiple domains such as politics (*kejriwal, gandhi,* and *narendramodi*), entertainment (*salman* and *allahdin*), social events/festivals (*mahashivratri* and *shaadi*), sports (*indvspak*), etc.
4. **Message Length**: Figure 6 shows the distribution of the message length for the code-mixed and the translated sentences. Distribution of message length for code-mixed and the translated sentences follows a similar trend.
5. **Quality of Translations (QT)**: To evaluate the quality of the translations by the annotators, we randomly sample 1000 sentences from the corpus. We provide two labels to each of the translation *correct translation* and *require change*. The correct translation should be syntactically and semantically correct. We calculate the quality of translation as follows

$$QT = \frac{Count\ of\ correct\ translations}{Sample\ size}$$

822 samples out of 1000 do not require any changes. Thus, the quality of translation is *0.822*. The ambiguity and the noise in the code-mixed text make the text challenging to translate even for the highly familiar and expert code-mixed language speakers.

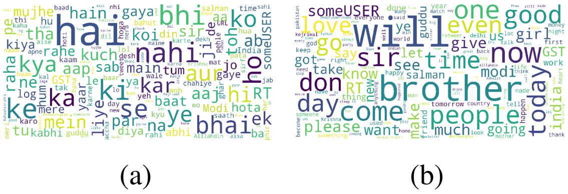

Figure 5: Word cloud of the (a) code-mixed and (b) translated sentences.

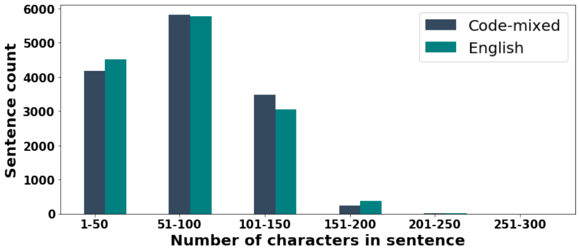

Figure 6: Distribution of message length for the code-mixed and English messages.

4 Evaluation of Machine Translation Systems

Here, we demonstrate the performance of the widely used machine translation systems on the code-mixed text. We experiment with two popular machine translation systems (Google

[2] We set the threshold to length six to remove the Romanized Hindi stopwords.

Translate and Bing Translate) and evaluate their performance on our proposed corpus. We use three different metrics to evaluate system performance. Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002), Word Error Rate (WER), and Translation Error Rate (TER). The values of these three metrics lie between 0 and 1.

To the best of our knowledge, we do not find translation systems build especially for the code-mixed sentences. The majority of the machine translation systems perform well for the monolingual translation tasks. However, these systems demonstrate severe limitations in translating code-mixed text. For the code-mixed text, the current machine translation systems assume input text to be in a single source language. Next, we describe the two translation systems and our proposed approach.

1. **Bing Translate (BT)**: BT is a translation service provided by Microsoft. It supports translation in 60 different languages[3] with neural machine translation capability in almost all the most frequently used languages. For translation, we set the language of the code-mixed input sentence as Hindi.
2. **Google Translate (GT)**: Next, we evaluate the performance of the GT on the code-mixed corpus. GT is a translation service provided by Google with the translation capability in 109 languages. It is the most widely used translation service with over 500 million total users, with more than 100 billion words translated daily[4]. We set GT to auto-detect the language of the code-mixed input sentence.
3. **Proposed Pipeline + Google Translate (PPGT)**: In addition to BT and GT, we propose a simple pipeline to use translation capabilities of already existing machine translation systems. In this paper, we specifically use GT. However, we can perform similar experiments with any machine translation system. The pipeline fragments the input sentence into multiple chunks before feeding it to GT. The steps of $PPGT$-based translation pipeline are:
   - We provide a label for each token of the code-mixed sentence based on the language (*English, Hindi, and other*).
   - We create chunks of Type-I using Hindi tokens with at most two English/other token allowed to be part of any chunk. A chunk of Type-I starts with a Hindi token.
   - We create chunks of Type-II using the tokens that are labeled as English/others and not part of any Type-I chunk.
   - We only translate the Type-I chunks using GT. We keep the chunks of Type-II as it is.

Figure 7 shows example translations of code-mixed sentences from two machine translation systems namely, BT and GT, and our proposed approach $PPGT$. In $PPGT$, we maintain the original order of the chunks as that of the code-mixed sentence while translating. For instance, the order of the chunks in Example II in Figure 7 is *[[par if its], [possible and any other guest needs a room ,], [mera room de de kisi ko bhi]]*.

Additionally, we randomly sample 100 code-mixed sentences from the corpus. We use human translated sentences as reference. Table 2 shows the performance evaluation of all the three systems. $PPGT$ outperforms both the other systems on all three evaluation metrics.

	BLEU-1	WER	TER
BT	0.146	0.751	0.885
GT	0.151	0.600	0.718
PPGT	**0.153**	**0.566**	**0.685**

Table 2: Evaluation of machine translation systems on various metrics. We prefer the high value of the BLEU-1 score and the low values of WER and TER.

As most of the machine translation systems do not perform well on the code-mixed data, we can build augmentation pipelines, similar to $PPGT$, on top of these systems that can preprocess and enhance the quality of the input to these systems. We posit that these pipelines can significantly address the challenges to code-mixed machine translation, as outlined in Section 2.

5 Limitations and Opportunities

The data collection, preprocessing, annotation, and resource expansion of $PHINC$ presents several limitations and opportunities. Some of the major insights and the future research prospects of the proposed dataset are:

- Human annotation of the code-mixed parallel corpus is a challenging task which demands significant effort and time. Building a large scale code-mixed parallel corpus solely with human annotators is infeasible. We can extend the proposed dataset using

[3] https://www.microsoft.com/en-us/translator/business/languages/

[4] https://www.blog.google/products/translate/ten-years-of-google-translate/

Example I

Code-mixed Sentence: @Prankoholic tumko matlab kya time hai din ka, kuch samaj nahi aata na
Type-I chunks: [tumko matlab kya time hai din ka, kuch samaj nahi aata na]
Type-II chunks: [@Prankoholic]
English Translation using BT: @prankoholic what time do you mean of the day, some society does not come.
English Translation using GT: @Prankoholic you mean what is the time of day, don't understand anything
English Translation using PPGT: @Prankoholic Do you mean what is the time of day, no sense

Example II

Code-mixed Sentence: par if its possible and any other guest needs a room , mera room de de kisi ko bhi
Type-I chunks: [par], [mera room de de kisi ko bhi]
Type-II chunks: [if its possible and any other guest needs a room ,]
English Translation using BT: On if its possible egg any other guest needs coming room , my room day to anyone
English Translation using GT: par if its possible and any other guest needs a room , mera room de de kisi ko bhi
English Translation using PPGT: par if its possible and any other guest needs a room , Give my room to anyone

Example III

Code-mixed Sentence: ab voh bola jisne kisi bhi party ko support karne se mana kardiya tha . . a flop show annaji
Type-I chunks: [ab voh bola jisne kisi bhi party ko support karne se mana kardiya tha]
Type-II chunks: [. . a flop show annaji]
English Translation using BT: Now Woh spoke , which was considered to support any party . . Come Flop Show Annaji
English Translation using GT: Now say that he had a desire to support any party. . A flop show Anna
English Translation using PPGT: Now speak that who had refused to support any party . . a flop show annaji

Figure 7: Example translation of code-mixed sentences using BT, GT, and PPGT.

various learning paradigms such as semi-supervised learning, active learning, etc.

- As machine translation systems require a large amount of data to build efficient systems, the dataset presented here alone will not be sufficient for traditional supervised methods. But, we can improve the performance of current SOTA machine translation systems by leveraging the proposed dataset. We can also develop systems with other techniques such as meta-learning, transfer learning, etc., which shows exciting results (Gu et al., 2018; Dabre et al., 2019) with other low resource languages.
- India is a highly diverse country with 23 official languages, and we observe multiple code-mixing pairs (Bengali-English, Telugu-English, etc.) very frequently on various platforms. We can extend the proposed technique for data collection and the translation pipeline to other code-mixed language pairs.
- As the syntactic and semantic structure of the code-mixed sentences is different from the monolingual sentences, the evaluation of the quality of code-mixed data for various tasks such as text summarization, neural machine translation, text generation, etc., requires advanced metrics. $PHINC$ can help in developing such evaluation metrics.
- We observe gender and racial bias in the code-mixed text. We can use the good-quality Hinglish sentences in $PHINC$ to identify and mitigate such biases.

6 Conclusion and Future Work

In this paper, we present a parallel corpus for the English-Hindi code-mixed machine translation task. We discuss various challenges in understanding and processing code-mixed text for various natural language understanding tasks. We also show limitations of the widely popular machine translation system build for monolingual corpus in dealing with code-mixed corpora. We evaluate the performance of the various translation systems on our parallel corpus. We present a translation pipeline that outperforms the various translation systems on our proposed code-mixed $PHINC$ dataset, demonstrating the opportunities in building efficient translation systems.

In the future, we plan to explore other code-mixed languages, especially those that are low-resource and endangered. We also plan to extend the corpus for various other code-mixing tasks such as word-embedding, language identification, named-entity recognition, etc. In addition, we can extend the dataset with more annotation using semi-supervised techniques. As the dataset size is significantly small to train a traditional supervised neural machine translation system, we can build the translation systems using few-shots learning techniques.

References

Scott Baldauf. 2004. A hindi-english jumble, spoken by 350 million. [Online; accessed 23-May-2020].

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23, Doha, Qatar. Association for Computational Linguistics.

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387.

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling code-mixed translation: Parallel corpus creation and mt augmentation approach. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 131–140.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491.

Ankush Khandelwal. 2018. Humor detection corpus. [Online; accessed 08-Jan-2020].

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. Gluecos: An evaluation benchmark for code-switched nlp. *arXiv preprint arXiv:2004.12376*.

James Lambert. 2018. A multitude of “lishes”: The nomenclature of hybridity. *English World-wide*, 39(1):1–33.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35.

R Mahesh K Sinha and Anil Thakur. 2005. Machine translation of bi-lingual hindi-english (hinglish) text. *10th Machine Translation summit (MT Summit X), Phuket, Thailand*, pages 149–156.

Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A corpus of english-hindi code-mixed tweets for sarcasm detection. *arXiv preprint arXiv:1805.11869*.

GaganDeep Singh Chhabra Vrishank Shete and Lokesh Mittal. 2016. Sentiment analysis on hindi-english code mixed data using svm. [Online; accessed 08-Jan-2020].

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979.