Proceedings of the International Conference on Sustainable Computing and Data Communication Systems (ICSCDS-2022)
IEEE Xplore Part Number: CFP22AZ5-ART; ISBN: 978-1-6654-7884-7

Code-Mixed Hinglish to English Language Translation Framework

Ishali Jadhav
Department of Computer Engineering
International Institute of Information Technology
Pune, India
ishalijadhav20@gmail.com

Aditi Kanade
Department of Computer Engineering
International Institute of Information Technology
Pune, India
aditikanade2110@gmail.com

Vishesh Waghmare
Department of Computer Engineering
International Institute of Information Technology
Pune, India
waghmarevishesh810@gmail.com

Sahej Singh Chandok
Department of Computer Engineering
International Institute of Information Technology
Pune, India
chandoksahej@gmail.com

Prof. Ashwini Jarali
Department of Computer Engineering
International Institute of Information Technology
Pune, India
ashwinij@isquareit.edu.in

2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) | 978-1-6654-7884-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSCDS53736.2022.9760834

***Abstract*—Indians, like many other non-English speakers around the world, avoid using single code in their social media conversations. They use transliteration and blend multiple languages to exhibit their linguistic proficiency by randomly merging English words (English-Hindi, English-Spanish, etc.). As a result, a large amount of unstructured text is generated because of the wide use of social media applications. Code-mixing (CM) is a fast-evolving field of study in the domain of text mining. The present situation of various social media posts, blogs, and reviews have a large use of code-mixed messages, due to its modern yet localized way of speaking. Linguistic codes from various languages are used for different purposes. Code-mixed Hindi and English is a typical practice observed in India's day-to-day language usage. Most people have already started to consider this mixing as a new language which has given birth to a brand new language termed "Hinglish". Hinglish is majorly used among the younger generation, as observed in the code-mixed data obtained via social sites and various other platforms. This mixing of languages stands as a new challenge to the concept of machine translation. It is important to recognize the foreign elements in a language and process them appropriately. As a result, a translation mechanism is needed to assist monolingual users, as well as for easier comprehension by language processing models. This paper proposes a pipelined mechanism for machine translation of a bi-lingual language i.e. Hinglish to monolingual English in this paper.**

Keywords—Code-Mixed Language; natural language processing; language identification; transliteration; machine translation; Hinglish;

I. INTRODUCTION

India is a linguistically diverse country with numerous languages spoken across its different regions. Moreover, due to its significantly long history of international acquaintances, English has become an essential part of India's education system and hence a population that is quite comfortable using bilingualism in communication has emerged. Such bilingualism initiates frequent code-mixing in informal conversations on a regular basis. Furthermore, due to the rapid rise of social media, this form of communication has grown even more popular. For instance, while conversing on social-media platforms like Facebook, WhatsApp, or Twitter, we have noticed that people are not very particular about expressing themselves in monolingual Hindi or English; instead, they prefer to mix up the languages. Translating such kind of data manually is burdensome and hence availing machines for the same is more desirable.

Due to vocabulary deficit, context misunderstanding, grammatical mistakes, bias, as well as other issues, the translation of languages is a challenging task. Hence, it becomes even more difficult to handle code-mixed data, as the blended structure introduces plenty of new issues. Therefore, we intend to create a system that will convert code-mixed English and Hindi (Hinglish) language to monolingual English. This will help people make better conversation and help them to increase their understanding of the code-mixed social media posts and also aid in code-mixed corpus analysis in various other domains. To implement this, we propose adding an extra layer of language identification and segmentation in order to improve the performance of pre-existing machine translation systems for the translation of code-mixed language. This is discussed in detail further in the paper.

II. CODE MIXING AND CODE SWITCHING

Language mixing is a natural result of bi-lingual and multilingual language usage. This mixing can occur between sentences or within a single sentence. A code might be thought of as a dialect or a language. It can be difficult to tell the difference between code-switching and code-mixing at times. In a single discourse, code-switching is a technique for switching between two or more languages amidst a conversation. For example, "Ram is a good student, aur wo bohot acha gaata bi hai." Here, we have switched from English to Hindi because the initial portion of the sentence is in English and the second portion is in Hindi. Since this occurs while switching from one sentence to another, it can also be termed as "Inter Sentential" switching.

Code-mixing, on the other hand, is the practice of employing different languages in a single conversational statement. For example, "Ram is eating aam on the terrace". The word 'aam' (mango in English) was adopted from Hindi and adjusted to fit in perfectly within the English

structure. This has occurred within a single sentence and hence is known as “Intra Sentential” switching.

Due to globalization and internationalization, there has been a major increase in the number of multilingual, multicultural, and interracial immigrants. However, not everyone finds it easy to completely adapt to the vast vocabulary of the universal language, English. Code-mixing, then, makes it easier for them to converse in a foreign language like English and prevents any kind of hesitance to use English. Also, with the tremendous increase in code-switching and code-mixing today people find it simple and comfortable to traverse and reveal their varied identities in conversations. It allows them to stay connected to their communities and preserve their identity by establishing a meaningful connection with those who speak the same language or have the same dialect.

III. System Purpose

The code-mixing we come across is not just limited to verbal conversations but textual as well. With the increasing usage of data mining and analysis, this code-mixed data needs to be comprehended well. A cultural difference in speaking is not the only thing that may give rise to code-mixing. One of the most common theories for why bilinguals code-mix is to make up for a lack of linguistic fluency. When asked about it, bilinguals are quick to express their "lack of formal knowledge" in one or both languages. Some people, though, may find it challenging to comprehend. However, there is currently no such translation method for code-mixed text. One reason is that, in order to perform well in this area, most machine translation (MT) systems require a large number of "parallel" sentences, which is difficult to synthesize due to the variety of languages. To ease the translation process, a framework is needed to facilitate code-mixing on existing MT systems, without using a parallel corpus. This could be utilized in different ways for developing better language models to build multilingual systems like speech recognition systems and chatbots that can handle code-mixing and create true engagement.

IV. Related Work

Several studies on code-mixed data, particularly including language tagging, have been conducted in the recent past. Bhattu and Ravi (2016) [1] have created a two-stage language classification method. The first level of classification was based on sentence-level character n-grams, while the second level was based on word-level character n-grams. Singh et al. (2018) [2] experimented with different Recurrent Neural Network (RNN) cells, encoder and decoder depths, output activation functions, and the number of RNN cells in each layer in multiple Seq2Seq models. They also presented a unique semi-supervised language identifier that exploits the character-level differences in languages to build their Named Entity Recognition (NER) model. Mandal and Singh (2018) [3] used multichannel neural networks that combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for word-level language identification. Furthermore, with respect to back transliteration and translation of languages, Ravishankar (2017) [4] discusses how to back-transliterate transliterated Marathi words into Roman using a finite-state system. Lepage and Luo (2015) [5] proposed back translation architecture based on the SMT framework.

Dhar et al. (2018) [6] created a parallel corpus of code-mixed English-Hindi and English in order to tackle the problem of code-mixed translation. They have also presented an augmentation pipeline that is used to boost the performance of existing machine translation systems in terms of code-mixed translation. Alternatively, Mahate et al. (2019) [7] used a translation-transliteration framework in order to translate Bengali-English code-mixed sentences to monolingual Bengali without incorporating a parallel corpus. Similarly, Sinha and Thakur (2005) [8] have used morphological analyzers to translate Hindi-English code-mixed to English monolingual. However, they did not conduct an in-depth analysis. Gautam et al. (2021) [9] have used mBART, a multilingual sequence-to-sequence model that has been pre-trained, to create a supervised machine translation system for English to Hinglish. Their system gave a BLEU score of 12.22. Jasim et al. (2020) [10] suggested a partial back-translation approach “PhraseOut”. Rather than translating the entire monolingual target sentence back into the source language, only high confidence phrases were substituted, with the remainder of the words remaining in the target language.

V. Methodology

Previously discussed methodologies consist of partial transliteration systems, parallel corpus approaches, and different combinations of languages with English. Our proposed approach is a pipeline of four stages that do not use any parallel corpus and targets the translation of specifically Hindi-English language combination (i.e. Hinglish). The first stage is language tagging where each word is assigned a language label, either Hindi or English. This helps us in segmenting the given code-mixed sentence into sub-sequences that are of the same language. In the second stage, the romanized Hindi segments are back-transliterated to the Devanagari form. Finally, the third stage translates the English segments of the code-mixed sentence to Hindi using a neural machine translation system.

Further, we join these transliterated and translated segments to form a monolingual Hindi sentence and translate it to English using the usual neural machine translation. We have used Hindi as the matrix language and English as the embedded language in our system. The matrix language provides structure to the code-mixed sentence and blends with another language by embedding tokens from the embedded language. The entire system architecture is portrayed in Figure 1.

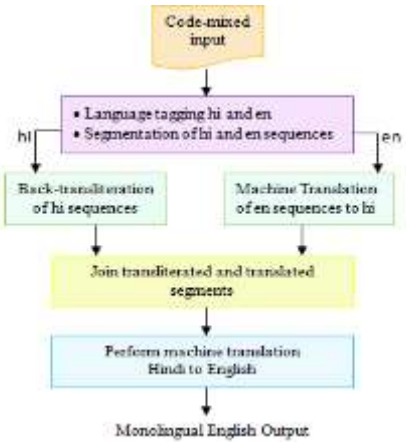

Fig. 1. System Architecture

A. *Language Identification and Segmentation*

In a Natural Language Processing (NLP) scenario, the task of identifying the natural language, in which any document or some part of it is written, is referred to as language identification. For more than fifty years, researchers have been studying language identification. Since text processing algorithms often assume that the input text language is known, language identification is now a necessary aspect of most of the text processing pipelines. Similarly, in order for language translation to be used to translate a document into a target language, it is usually necessary to establish the content's source language, which is the job of language identification.

For language classification, we have used an LSTM based neural method along with a pre-labeled dataset of around 25k most commonly used Hindi and English words obtained from Precog [11]. A part of the dataset comprising of two columns for the word and its corresponding label is shown in Table I. We have implemented two LSTM layers and a dense layer with the relu, sigmoid combination and applied the Adam optimizer which computes the individual learning rates. The score of every word in the given input sentence is calculated, by this model, based on which it is tagged as either English or Hindi.

TABLE I. LANGUAGE TAGGING DATASET

Word	Label
cricket	en
chal	hi
video	en
kabhi	hi

Based on this tagging, we divide the code-mixed sentence into Hindi and English segments, i.e. fragments of the sentence written in the same language. These labeled fragments are then individually sent for transliteration and translation depending on the tag. In the examples below, segments are denoted by brackets.

Example 1: (I just realized)$_{En}$ (ki mera bohut)$_{Hi}$ (time waste)$_{En}$ (hogaya h)$_{Hi}$

Example 2: (I think)$_{En}$ (koi aadmi hum sab ko dhokha de rha h)$_{Hi}$

There exist some limitations, discussed in Section VII, in language identification in a code-mixed corpus which could cause errors in the system. A loss function is used to optimize a machine learning method. Our language prediction model has a validation accuracy of 0.90 and a validation loss of 0.35 which implies a lower error rate in prediction.

Algorithm

Input: Code-mixed sentence S consisting of English (En) and Hindi (Hi) language.
Output: Language tags to each and every word in the given sentence.
Begin:
Pre-process the input sentence and tokenize it into words.
Remove all numeric values and special characters as they do not belong to any language.
`For each word w ∈ S`
Get the vectorized value of that word
Use the predict method to get the probability of the word w being Hindi
If prediction > 0.5
`Word w ∈ Hi`
Tag that word as Hindi
Else
`Word w ∈ En`
Tag that word as English
Print the sentence with words having tags of their respective language.
End

B. *Language Transliteration*

Language transliteration is the task of transforming a character or word from one alphabetical system to another. The morphological structure of words must be retained to the maximum degree feasible while constructing the transliteration model. Rule-based and machine translations are the two most common transliteration methods. Machine translation methods use parallel training data to learn, whereas the rule-based system undergoes character mappings between the two scripts.

For the implementation of back-transliteration in our system, we have used pre-existing transliteration systems. The Hindi segments in the original code-mixed sentence, written in romanized form, are back-transliterated to the Devanagari script. We have experimented with two different transliteration models, i.e. indic-trans [19] and Google transliteration API with the google-transliterate-api package, out of which Google transliteration outperformed indic-trans significantly. Below is an example of the back-transliteration of romanized Hindi.

Example: Mai aaj bahar jaunga

Transliteration: मैं आज बाहर जाऊंगा

C. *Language Translation*

The translation of speech or text by a computer with no human intervention is referred to as machine translation. Neural Machine Translation (NMT) has the ability to tackle many of the limitations of standard machine translation

systems. NMT is generally composed of two RNNs, one for consuming the input text sequence and the other for generating the translated output text, with an additional attention layer for longer inputs. An example of translation is given below.

Example: I think it will rain tomorrow.

Translation: मेरे विचार से कल बारिश होगी

In our system, we translate the tagged English segments of the code-mixed sentence to monolingual Hindi in Devanagari script. Also, the joined transliterated and translated segments in Hindi are again translated to English using the same translation method. We have explored various different pre-built translation models like Google Neural Machine Translation (GNMT) [14] via googletrans api package and Helsinki model [18]. The best outcomes were obtained with GNMT which also consists of an encoder-decoder network along with an additional attention module.

TABLE II. TRANSLATION OUTPUT COMPARISON

Input	GNMT Output	LID [a] + GNMT Output
mjhe to lgta hai they hate hindus too tabhi unki hindu college walo se banti nahi hai	I think they are Hindus, then only their Hindus are not made with college people.	I think they also hate Hindus, that's why they don't get along with Hindu college people.
yeh goggles laga kar bowling karne kaa funda aaj tak samajh mein nahi aayaa	The fund of bowling with these goggles has not been understood till date.	The noose of bowling with this glasses on is still not understood.
and koi bhi match nhi hua with the graphics shown in the photo	and koi bhi match nhi hua with the graphics shown in the photo	and none matched with the graphics shown in the photo
dont know how but at these moments natural acting comes out and important is that har baar naye bahane dimag me hote the	dont know how but at these moments natural acting comes out and important is that har baar naye bahane dimag me hote the	Don't know how, but in these moments the natural acting comes out and the important thing is that every time new excuses are in mind.

[a] LID: Language Identification

D. Evaluation Metrics

The segment-level similarity-based method is used by the majority of automated metrics; it compares an MT system's output to a human-generated reference translation and calculates how close the machine-translated sentence is to that reference translation. The comparison unit can be a word, but the metrics also compute precision scores using n-grams (a sequence of n elements from a text or speech sample).

a) BLEU score: The Bi-Lingual Evaluation Understudy (BLEU) metric assesses the similarity of machine-translated text to a set of reference translations and is used to automatically evaluate machine-translated texts.

b) TER score: The Translation Error Rate (TER) is a machine translation error metric used to measure the number of corrections required to modify the output of a system to one of the samples.

c) WER score: The Word Error Rate (WER) is calculated by dividing the number of faults by the total number of words.

VI. RESULTS & FINDINGS

Since our system adds an additional layer for the improvement of the existing NMT, we have evaluated our system in comparison with the existing Google NMT. The output comparisons for a code-mixed input are presented in Table III. We chose the BLEU score, Word Error Rate (WER), and Translation Error Rate (TER) as to accuracy metrics since they are optimal for machine translation. The scores for both systems are shown in Table II. It can be observed that our pipelined approach surpasses the pre-existing system of GNMT by a fair margin i.e. 0.241 BLEU, which tells us that a strong language identification layer is essential for performing translation of code-mixed corpus. As the TER and WER scores for our system are less, this shows the less error rate which is advantageous for any system. Therefore, we can speculate that our additional layer boosts the existing system's performance. It is also advantageous in terms of the data and the training time required since this system does not incorporate any parallel corpus.

TABLE III. EVALUATION RESULTS

Model	Scores [b]		
	BLEU	***TER***	***WER***
GNMT	0.496	0.437	0.456
LID + GNMT pipeline	0.737	0.256	0.238

[b] Note: Bleu score – higher the better; TER and WER – lower the better.

VII. LIMITATIONS

Even after using a matrix language as an intermediate language, i.e. Hindi in our case, for a smooth translation process, there were many ambiguity errors. Some examples of the inputs which showed the ambiguity due to the similarity of words in both languages are presented below.

Example 1: "uss" is recognized as an abbreviation

Code-mixed Input: tune {uss} ladke ko dekha kya

System Output: तूने {यूएस} लड़के को देखा क्या

Example 2: "are" is recognized as an English word

Code-mixed Input: {are} yaar office ka time khatam ho gaya

System Output: {हैं} यार कार्यालय का समय ख़तम हो गया

Example 3: "to" is recognized as an English word

Code-mixed Input: I went {to} movie aur mujhe badaa mazaa ayaa

System Output: मैं गया {तो} फ़िल्म और मुझे बड़ा मज़ा आया

Example 4: verb changes according to the gender of the noun in Hindi

Code-mixed Input: tune uss student ki dress dekhi kitni choti thi

System Output: तूने यूएस छात्र की पहनावा देखि कितनी छोटी थी

Example 4: verb changes according to the gender of the noun in Hindi

Code-mixed Input: Aapne apna commitment nahi nibhayaa

System Output: आपने अपना प्रतिबद्धता नहीं निभाया

VIII. CONCLUSION & FUTURE WORK

We conclude that previous studies have shown the implementation of code-mixed translation for different languages using a parallel corpus. In our system, we have designed a pipelined architecture that can be utilized to improve the translation of Hindi-English code-mixed data with the additional language identification layer and have later compared it with the existing system of Google neural machine translation and our system was able to give better results. We believe that our system will aid monolingual speakers to understand and analyze the context and meaning being conveyed by the text consisting of multiple languages (mainly Hindi and English) across various social platforms. Some more work has to be done to solve the ambiguity problems that we have mentioned because of the similarity of words with different meanings in Hindi and English language and this will definitely increase the system performance much more. Also, this system could be extended to different variety of languages in the future.

REFERENCES

[1] S. N. Bhattu and V. Ravi, "Language Identification in Mixed Script Social Media Text." Fire workshops, 2015, pp. 37-39.

[2] K. Singh, I. Sen and P. Kumaraguru, "Language identification and named entity recognition in hinglish code mixed tweets", Proceedings of ACL, Student Research Workshop, 2018, pp. 52-58.

[3] S. Mandal and A. K. Singh, "Language identification in code-mixed data using multichannel neural networks and context capture", arXiv preprint arXiv:1808.07118, 2018.

[4] V. Ravishankar, "Finite-state back-transliteration for Marathi, The Prague Bulletin of Mathematical Linguistics", De Gruyter Poland, Vol. 108(1). 2017.

[5] J. Luo, Y. Lepage, "Handling of Out-of-vocabulary Words in Japanese-English Machine Translation by Exploiting Parallel Corpus", Int. J. Asian Lang. Process, Vol. 23(1), 2015, pp 1-20.

[6] M. Dhar, V. Kumar, M. Shrivastava, "Enabling code-mixed translation: Parallel corpus creation and mt augmentation approach", Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing, 2018, pp131-140.

[7] S. K. Mahata, S. Mandal, D. Das, S.Bandyopadhyay, "Code-mixed to monolingual translation framework", Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019, pp30-35.

[8] R. M. K. Sinha, A. Thakur, "Machine translation of bi-lingual hindi-english (hinglish) text", Proceedings of Machine Translation Summit X: Papers, 2005, pp149-156.

[9] D. Gautam, P. Kodali, K. Gupta, A. Goel, M. Shrivastava, P. Kumaraguru, "Comet: Towards code-mixed translation using parallel monolingual sentences", InProceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching, 2021, pp. 47-55.

[10] B. Jasim, V. Namboodiri, C. V. Jawahar, "PhraseOut: A Code Mixed Data Augmentation Method for MultilingualNeural Machine Tranlsation", Proceedings of the 17th International Conference on Natural Language Processing (ICON), 2020, pp. 470-474.

[11] K. Singh, I. Sen, P. Kumaraguru, "Language Identification and Named Entity Recognition in Hinglish Code Mixed Tweets.", Proceedings of ACL 2018, Student Research Workshop, 2018, pp 52—58.

[12] H. Jhamtani, S. K. Bhogi, and V. Raychoudhury, "Word-level language identification in bi-lingual code-switched texts", Proceedings of the 28th Pacific Asia Conference on language, information and computing, 2014, pp348—357.

[13] J. Emond, B. Ramabhadran, B. Roark, Moreno, M. Pedro , Min, "Transliteration based approaches to improve code-switched speech recognition performance", 2018 IEEE Spoken Language Technology Workshop (SLT), IEEE,2018, pp448—455.

[14] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", arXiv preprint arXiv:1609.08144, 2016.

[15] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, M. Shrivastava, "IIIT-H System Submission for FIRE2014 Shared Task on Transliterated Search," in FIRE, 2015, pp.48–53.

[16] R. Heredia and J. Altarriba, "Bilingual language mixing: Why do bilinguals code-switch?", Current Directions in Psychological Science, SAGE Publications Sage CA: Los Angeles, CA, 2001, pp 164-168.

[17] G. Medvedev, "Google translate in teaching English", Journal of teaching English for specific and academic purposes, 2016, pp 181-193.

[18] R. Ostling, Y. Scherrer, J. Tiedemannand Tang, N. Gongbo , Tommi, "The Helsinki neural machine translation system", arXiv preprint arXiv:1708.05942, 2017.

[19] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, and M. Shrivastava, "IIIT-H System Submission for FIRE2014 Shared Task on Transliterated Search," in FIRE, 2015, pp. 48–53.

[20] S. R. Laskar, A. Dutta, P. Pakray, S. Bandyopadhyay, "Neural machine translation: English to hindi", conference on information and communication technology, IEEE, 2019, pp. 1-6.

[21] C. Verma, A. Singh, S. Seal, V. Singh, I. Mathur, "Hindi-English neural machine translation using attention model", Int. J. Sci. Technol. Res, 2019, Vol. 8(11).