# Enabling Code-Mixed Translation: Parallel Corpus Creation and MT Augmentation Approach

**Mrinal Dhar**
IIIT Hyderabad
Gachibowli, Hyderabad
Telangana, India
mrinal.dhar@gmail.com

**Vaibhav Kumar**
IIIT Hyderabad
Gachibowli, Hyderabad
Telangana, India
vaibhav4595@gmail.com

**Manish Shrivastava**
IIIT Hyderabad
Gachibowli, Hyderabad
Telangana, India
m.shrivastava@iiit.ac.in

## Abstract

Code-mixing, use of two or more languages in a single sentence, is generated by multi-lingual speakers across the world. The phenomenon presents itself prominently in social media discourse. Consequently, there is a growing need for translating code-mixed hybrid language into standard languages. However, due to the lack of gold parallel data, existing machine translation systems fail to properly translate code-mixed text.

In an effort to initiate the task of machine translation of code-mixed content, we present a newly created parallel corpus of code-mixed English-Hindi and English. We selected previously available English-Hindi code-mixed data as a starting point for our parallel corpus, and 4 human translators, fluent in both English and Hindi, translated the 6,096 code-mixed English-Hindi sentences into English. With the help of the created parallel corpus, we analyzed the structure of English-Hindi code-mixed data and present a technique to augment run-of-the-mill machine translation (MT) approaches that can help achieve superior translations without the need for specially designed translation systems. The augmentation pipeline is presented as a pre-processing step and can be plugged with any existing MT system, which we demonstrate by improving code-mixed translations done by systems like Moses, Google Neural Machine Translation System (NMTS) and Bing Translator.

## 1 Introduction

In the last decade, digital communication mediums like e-mail, Facebook, Twitter, etc. have allowed people to have conversations in a much more informal manner than before. This informal nature of conversations has given rise to a new form of hybrid language, called *code-mixed* language, that lacks a formally defined structure.

Myers-Scotton (1993) defines code-mixing as "the embedding of linguistic units such as phrases, words and morphemes of one language into an utterance of another language". *Code-switching* is a similar concept, except that code-mixing is observed entirely in a single sentence, while code-switching occurs across sentences. For the purposes of this study, however, we will not make a difference between code-mixing and code-switching and treat them both as code-mixing.

Code-mixing of Hindi and English, where sentences follow the syntax of Hindi but borrow some vocabulary from English is very prevalent on social media content in India, where most people are multilingual. An example of a code-mixed English-Hindi sentence is presented below:

- "*Main kal* movie *dekhne jaa rahi thi* and *raaste me* I met Sam."

  **Gloss** : I yesterday [movie] to-see go Continous-marker was [and] way in [I met Sam].

  **English Translation** : I was going to a movie yesterday and I met Sam on the way.

This phenomenon is present in informal communication in almost every multi-lingual society, as studied by Choudhury et al. (2007). They investigated how language used on these social media platforms, which they have called *texting* language, differs from the standard language that is found in more formal texts like books. It is also more common in the areas of the world where people are naturally bi- or multi-lingual. Usually, these are the areas where languages change over short geo-spatial distances and people generally have at least a basic knowledge of the neighbouring languages (Jamatia et al., 2015). A very good example for this is a country like India which has an extensive language diversity and where dialectal changes frequently instigate code-mixing.

In recent times, we have seen an explosion of Computer Mediated Communication (CMC) worldwide (Herring, 2003). In CMC, language use lies somewhere in between spoken and written forms of a language and tend to use simple shorter constructions, contractions and phrasal repetitions typical of speech (Danet and Herring, 2007). Such conversations, especially in social-media are both multi-party and multilingual, with mixing occurring between two or more languages, where the choice of language-use being highly influenced by the speakers and their communicative goals (Crystal, 2011). With multiple languages coming into play, and the variety of factors which influence the usage of those, the task of processing text becomes quite difficult.

As of now, according to the data of internetworldstats.com, the Internet has four-hundred fifty million English speaking users out of one billion five hundred million total users. This means that the market for English language is slightly less than one third of the total market. In other terms, most current approaches to information extraction exploiting social media and user-generated content (UGC), that are predominantly developed for English, are working with a mere third of the total data available.

The huge part of UGC that is not in English is currently being neglected mostly due to the relatively ephemeral character of UGC in general. Such content remains usually untranslated unless the users themselves so choose, because (i) it expresses opinions, and most users are much more likely to express or look for other opinions in the same language as their own rather than translated from a different one; (ii) it is generated and updated at an extremely fast pace and has a very short lifespan, which rules out in practice the possibility of translation by human subjects; and (iii) it is also produced in immense quantities, which, together with the previous point, ends up rendering translation by human subjects effectively impossible (both in terms of time and cost). All this leads to an enormous body of information being constantly generated which is also being constantly lost behind language barriers: the consolidation of Web 2.0 has caused an unprecedented increase in the amount of data and each individual user is currently being deprived of most of it (Carrera et al., 2009).

These language barriers are intensified by the fact that a huge number of people don't use just one language, but multiple languages simultaneously, in the form of code-mixing. Even if we were able to create a system capable of processing information in every language, or perhaps a system capable of translating text from any given language to another, we would still not be able to break down the language barrier completely, due to the phenomenon of code-mixing. This is where code-mixed translation comes into play.

While translation of code-mixed text has been a requirement for some time, there is a noticeable lack of resources for this task. To tackle this problem, we have created a set of 6,096 English-Hindi code-mixed and monolingual English gold standard parallel sentences as an initial attempt to promote generation of data resources for this domain.

However, most MT systems require a significant number of "parallel" sentences to perform well. While we wait for large parallel corpora for code-mixed text to be developed, we could make do by equipping the existing state-of-the-art MT systems to handle code-mixed content. This necessity has motivated us to develop an augmentation pipeline to support code-mixing on existing MT systems.

To summarize, the main contributions of this work are as follows:

- We are releasing [1] a gold standard parallel corpus consisting of 6,096 English-Hindi code-mixed and monolingual English that we created.
- We have developed an augmentation pipeline for existing machine translation systems that can boost their translation performance on code-mixed content.
- We carry out experiments involving various machine translation systems like Moses, Google NMTS and Bing Translator to compare their translation performance on code-mixed text, and demonstrate how our augmentation pipeline improves their translation results.

The rest of our paper is divided into the following sections: We begin with a study of research conducted in this domain in Section 2. We discuss the process of creation of the corpus and its features in Section 3. In Section 4, we introduce our augmentation pipeline for machine translation systems and describe the approach in detail. Then in Section 5, we perform experiments with existing machine translation systems, and describe the impact of our augmentation pipeline on their translation accuracy for code-mixed data, proceeded by a discussion of our results.

## 2 Related Work

In recent times, there has been a lot of interest from the Computational Linguistics community to support code-mixing in language systems and models.

Due to a massive growth of social media content, the usage of noisy non-standard tokens online has also increased. Hence, text normalization systems have become necessary that can convert these non-standard tokens to their standard form. Language identification at the word level was attempted by Nguyen and Doğruöz (2013) on Turkish-Dutch posts collected from an online chat forum. They performed comparisons between different approaches such as using a dictionary and statistical models. They were able to develop a system which had an accuracy of 97.6%, and concluded that language models prove to be better than a dictionary based approach. Barman et al. (2014) explored the same task on social media text in code-mixed Bengali-Hindi-English languages. They annotated a corpus with over 180,000 tokens, and used statistical models with monolingual dictionaries to achieve an accuracy of 95.76%.

Vyas et al. (2014) deal with POS tagging of English-Hindi code-mixed data that they extracted from Twitter and Facebook. Social media is a good source for obtaining code-mixed data as it is the preferred choice of the urban youth as informal platforms for communication. Bali et al. (2014) have done a study of English-Hindi code-mixing on Facebook, and their investigation demonstrates the extent of code-mixing in the digital world, and the need for systems that can automatically process this data. Sharma et al. (2016) have addressed shallow parsing of English-Hindi code-mixed data obtained from online social media. They were the first to attempt shallow parsing of code-mixed data, to the best of our knowledge.

Language identification in code-mixed data is an important step because it might determine how to further process the data in tasks like POS tagging, for example. Chittaranjan et al. (2014) explore a CRF based system for word level identification of languages.

Apart from this, Raghavi et al. (2015) have explored the problem of classification of code-mixed questions. By translating words to English before extracting features, they were able to achieve a greater accuracy in classification. WebShodh, developed by Chandu et al. (2017), is an online web based question answering system that is based on this work.

Gupta et al. (2016) created a dataset of code-mixed English-Hindi sentences along with the associated language and normalization. This was the first attempt to create such a linguistic resource for the language pair. They also presented an empirical study detailing the construction of language identification and normalization system designed for the language pair.

Code-mixed translation has been attempted previously by Sinha and Thakur (2005) from a linguistics perspective. However, we primarily draw our inspiration from the skeleton model presented by Rijhwani et al. (2016). Though they present the broad idea similar to ours, we find that many of the assumptions

[1] https://github.com/mrinaldhar/en-hi-codemixed-corpus

made were very simplistic and no evaluation or results were provided and no systems were released. We provide a deeper look into the code-mixed translation process and demonstrate the impact along with benchmark dataset.

## 3 Corpus creation

### 3.1 Analysis

We started with the dataset created and released by Gupta et al. (2016). They collected 1,446 sentences from social media, and performed language identification and word normalization on these sentences. We also obtained 771 sentences from the dataset released as part of the ICON 2017 tool contest on POS-tagging for code-mixed social media text, created by collection of Whatsapp chat messages. Additionally, we use the dataset released by Joshi et al. (2016) for the task of sentiment analysis of code-mixed content, which contains 3,879 code-mixed sentences.

For our study, we removed annotations such as sentiment labels, POS tags, etc. from the obtained datasets, and only used raw sentences for the task of corpus creation. For the augmentation pipeline, we also make use of the language identifiers, wherever available in the dataset samples.

The 6,096 code-mixed sentences contain a total of 63,913 tokens. Of these tokens, 37,673 are Hindi words and 16,182 are English words. The rest of the tokens were marked as "Rest". "Rest" would mean that these tokens could be abbreviations, named entities, etc. The tokens in the data were already normalized.

It is very crucial to find the level of code-mixing in data, since if the extent of code mixing is too less, then it is as good as a monolingual corpus and would not provide us much benefit in the task of code-mixed translation. Hence, in order to find the level of code-mixing present in the data, we use the Code-Mixing Index (CMI) (Das and Gambäck, 2014) defined as:

$$CMI = \begin{cases} 100 \times [1 - \frac{max\{w_i\}}{n-u}] & n > u \\ 0 & n = u \end{cases} \quad (1)$$

where, $w_i$ is the number of words tagged with a particular language tag, $max\{w_i\}$ represents the number of words of the most prominent language, $n$ is the total number of tokens, $u$ represents the number of language independent tokens (such as named entities, abbreviations), in our case these would be the words marked as "Rest".

Summarized statistics of the data can be found in Table 1. From the table we can see that the code-mixing index is around 30.5, hence, we can safely assume that the data has a decent variety of code-mixed sentences which could be effectively used for the creation of translation systems.

### 3.2 Methodology for Annotation

For the task of translation, we selected 4 annotators who were fluent in both English and Hindi. Before starting the process of annotation, we randomly sampled 100 sentences from the corpus. We then asked one of the annotators to translate the given sentences into English. The other three annotators judged the translated sentences into two categories, *Totally Correct (TC)* and *Requires Changes (RC)*. Finally, we use the Fleiss' Kappa measure in order to calculate agreement.

Fleiss' Kappa is a statistical measure for assessing the reliability of agreement between a fixed number of raters when assigning categorical ratings to a number of items or classifying items in order to calculate the agreement. The measure is defined as follows:

Let $N$ denote the number of subjects, $n$ denote the number of ratings per subject, and $k$ be the number of categories into which assignments are to be made. In our case $N$=100, $n$=3, $k$=2. Let the subjects be indexed by $i$ = 1, ... $N$, categories be indexed by $j$ = 1, ... $k$ and $n_{ij}$ be the number of raters who assigned the $j^{th}$ category to the $i^{th}$ subject. Then,

$$P_i = \frac{1}{n(n-1)}[(\sum_{j=1}^{k} {n_{ij}}^2) - (n)] \quad (2)$$

here, $P_i$ denotes the extent to which raters agree to the $i^{th}$ subject,

$$\overline{P} = \frac{1}{N} \sum_{i=1}^{N} P_i \tag{3}$$

$$p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij} \tag{4}$$

$$\overline{P_e} = \sum_{j=1}^{k} {p_j}^2 \tag{5}$$

and finally,

$$\kappa = \frac{\overline{P} - \overline{P_e}}{1 - \overline{P_e}} \tag{6}$$

where $\kappa$ denotes the Fleiss' Kappa score.

Using this, we found out that the $\kappa$ score for code-mixed to English translation was close to 0.88. We can consider this score to be in correspondence with almost complete agreement. We speculate that such a high agreement could arise from the fact that the annotators had the same cultural background and shared similar communities.

In Table 2 we provide a few examples of code-mixed English-Hindi translated to English with the help of our pipeline. From the examples we can clearly see that the dataset sentences consist of transliterated Hindi words. Transliterated words usually pose a problem because one needs to come up with a standard form of those words before proceeding to the task of translation. However, Gupta et al. (2016) had already normalized the transliterated words. Hence, we were able to translate sentences without having to normalize the words.

| **Type** | **Value** |
|---|---|
| Total no. of code-mixed sentences | 6,096 |
| Total no. of tokens | 63,913 |
| Total no. of Hindi tokens | 37,673 |
| Total no. of English tokens | 16,182 |
| Total no. of 'Rest' | 10,094 |
| Code Mixing Index | 30.5 |

Table 1: Corpus Statistics

Some examples from the dataset which illustrate code-mixing:

1. I was really trying *ki aajayun*

   **Gloss:** [I was really trying] I come

   **Translation:** I was really trying to come

2. Sorrrry, *aaj subah tak pata nhi tha* that I wudnt be able to come today

   **Gloss:** [Sorrrry], today morning till know not did [that I wudnt be able to come today]

   **Translation:** Sorrrry, I didn't know until this morning that I wudnt be able to come today.

3. *tu udhar ka* permanent *intezaam karke aa* !

   **Gloss:** you there is [permanent] arrangement do come !

   **Translation:** Come over after making a permanent arrangement there !

4. btw i was thinking *mai pehle ghar chale jaungi*, and *sham ko* venue *pe aa jaungi*

   **Gloss:** [btw i was thinking] i first home walk go, [and] evening [venue] on come go

   **Translation:** btw I was thinking that I'll first go home and then come to the venue in the evening.

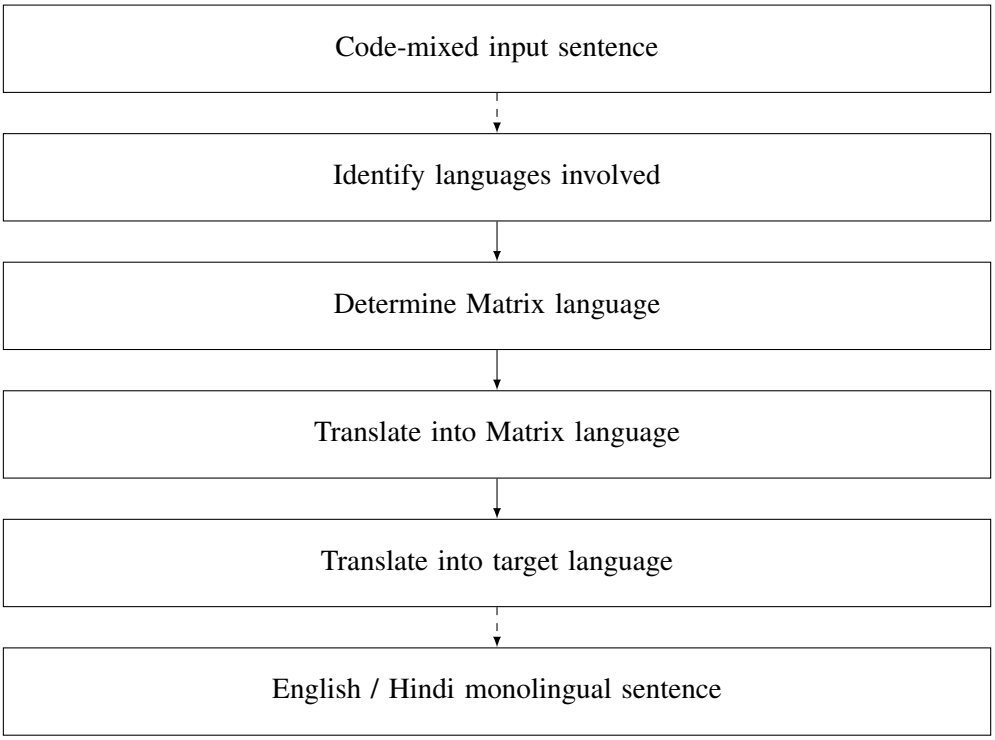


Figure 1: Translation augmentation pipeline

## 4 MT Augmentation Pipeline

Instead of attempting to create a new machine translation system specifically for code-mixed text, which is not feasible due to lack of abundant gold standard data, we now introduce an approach that will augment existing systems for use with code-mixed languages in the form of the pipeline described in Figure 1.

In order to improve the results of the MT system for code-mixing, we attempt to change the input code-mixed sentence to more closely resemble the kind of input these systems work best for - a monolingual sentence. As the code-mixed sentence follows the structure of the Matrix language, we translate words of the code-mixed sentence that belong to the Embedded language (Em) to the Matrix (Ma) language using a machine translation system (Em-Ma MT). The resulting sentence would be translated to the desired target language (Tgt) using another existing MT system (Ma-Tgt MT), which may or may not be the same as the Em-Ma MT system (depending on what the matrix and target languages are). For our experiments, we have fixed the target (Tgt) to be English, as we attempt to translate English-Hindi code-mixed language into English. The pipeline also recognizes when the target language is the same as the matrix language, and therefore there is no need to perform a translation of respective parts or the whole of the sentence.

### 4.1 Identification of languages involved

As a pre-processing step, we identify the languages that are involved in creating the hybrid code-mixed language in the text. This step is crucial in determining the pipeline to follow for the rest of the translation.

Gupta et al. (2016) released their dataset with manually annotated language identifiers associated with every word of the sentences in the data. For the other datasets, we make use of the Language Identification (LID) system by Bhat et al. (2015) to identify the corresponding language for each word in a code-mixed sentence.

### 4.2 Determination of Matrix Language

In the Matrix Language-Frame model proposed by Myers-Scotton (1997), a code-mixed sentence is formed by a Matrix language and an Embedded language. The overall morpho-syntactic structure of the sentence is that of the Matrix language, however, words from the Embedded language are also present in the sentence.

In other words, the Matrix language lends its syntactical structure and the Embedded language lends its vocabulary to the code-mixed sentence. The Matrix language is determined by employing the following heuristics in decreasing order of preference:

1. **The number of words in one language in the sentence**
   The language which has more words in the sentence is likely to be the matrix language.

2. **Determination of the syntactic structure of the text by detecting the language of the verb**
   For example, if the two languages involved in code-mixing are English and Hindi, and a sentence follows SVO structure, in that case English is likely to be the Matrix language of the sentence.

3. **Usage of function words of a particular language in the text**
   The language whose function words exist in the sentence might be the matrix language, since function words are associated with syntax.

### 4.3 Translation into Matrix Language

While most translation systems are unable to translate foreign (including code-mixed) words due to lack of training, borrowed and commonly used words are often found in parallel corpora. Commercial systems, like Google Translate, can translate frequent short phrases as well. Errors creep in when they have to deal with longer phrases. Keeping this in mind and to reduce the translation cost per sentence, we selected **the longest string of words belonging to the Embedded language** for translation to the Matrix language. This is to ensure the maximal meaningful translation with a single call to the Em-Ma MT engine. This results in a code-mixed sentence that follows the syntax of the Matrix language and also has the majority of its words in this Matrix language.

### 4.4 Translation into Target Language

Now that the code-mixed sentence has been translated into the matrix language, it can be directly translated into the Target language using the Ma-Tgt MT engine.,

## 5 Evaluation and Results

In order to evaluate our methods of augmentation, we consider the following existing machine translation systems: **Moses** (Koehn et al., 2007), **Google's Neural Machine Translation System (NMTS)** (Wu et al., 2016), **Bing Translator**.

For training a translation model for Moses, we used the English-Hindi parallel corpus released by Kunchukuttan et al. (2017). This dataset consists of 1,492,827 parallel monolingual English and monolingual Hindi sentences. The Hindi sentences were in the Devanagari script, and required pre-processing for use with our code-mixed dataset, which is entirely in Roman script.

We compare the output translations of these MT systems for code-mixed data, with and without the augmentation by our system. For accuracy metrics, we chose BLEU score, Word Error Rate (WER) and Translation Error Rate (TER) as they are ideal for use with machine translation.

As can be observed from Table 3, our augmentation pipeline significantly improves the translation accuracy of existing machine translation systems. Note that among the systems themselves, Google NMTS performs much better on code-mixed English-Hindi data as compared to traditional phrase based systems like Moses and even neural systems like Bing Translator. Even though Moses does not perform as well as the other systems described here for code-mixed data, our pipeline is still able to boost its performance significantly.

| Original sentence | Without Augmentation | With Augmentation |
|---|---|---|
| room *mei shayad kal bhi nahi* stay *karungi* , cancel *ho sakti hai uski* booking *abhi* ? | I will not stay in the room tomorrow, can I cancel her booking now? | I will not stay in the room tomorrow, can I cancel her booking now? |
| Sorrrry , *aaj subah tak pata nhi tha* that I wudnt be able to come today | Sorry , *aaj subah tak pata nahi tha* that I wouldn't be able to come today | Sorry , Did not know until this morning that I wudnt be able to come today |
| I was really trying *ki aajayun* | I was really trying *ki aajayun* | I was really trying I come |
| *par* if its possible and any other guest needs a room , *mera* room *de de kisi ko bhi* | *par* if its possible and any other guest needs a room , *mera* room *de de kisi ko bhi* | *par* if its possible and any other guest needs a room , Give my room to anyone |
| *toh hum aaj train ki ticket karwa lenge* . | So we will get a train ticket today. | So we will get a train ticket today. |
| *tu udhar ka* permanent *intezaam karke aa* ! | You come here by arranging Permanent! | You come here with a permanent arrangement! |

Table 2: Augmenting Google Translate with our pipeline

| | Without Augmentation | | | With Augmentation | | |
|---|---|---|---|---|---|---|
| | **BLEU** | **WER** | **TER** | **BLEU** | **WER** | **TER** |
| Moses | 14.9 | 10.671 | 2.403 | 16.9 | 9.505 | 2.295 |
| Google NMTS | 28.4 | 5.882 | 0.692 | 37.8 | 4.030 | 0.537 |
| Bing Translator | 18.9 | 8.940 | 1.108 | 25.0 | 8.054 | 0.917 |

Table 3: Comparison of performance with and without using our augmentation pipeline. (Note: BLEU - higher is better, (WER,TER) - lower is better.)

## 6 Conclusions

In this paper, we have created a set of 6,096 English-Hindi code-mixed and monolingual English gold standard parallel sentences for promoting the task of machine translation of code-mixed data and generation of data resources for this domain.

We have also developed an augmentation pipeline, that can be used to augment existing machine translation systems such that translation of code-mixed data can be improved without training an MT system specifically for code-mixed text. Using the evaluation metrics selected, it is shown that there is a quantifiable improvement in the accuracy of translations with the augmentation proposed.

As part of our study, we have observed that long distance re-ordering of words is still an issue with code-mixed MT. Also, since code-mixed language does not have a standard form, it is difficult to establish a correct version of spelling for a particular word. Language identification is the most critical module for translation augmentation, because the same orthographic form can lead to valid words in multiple languages.

To take this work further, we intend to develop an end-to-end code-mixed MT system which can jointly perform the normalization, language identification, matrix language identification and two-step translation tasks. A hybrid model, partially trained on gold parallel corpus, may also be attempted.

## Acknowledgements

A special thanks to Aashna Jena, Abhinav Gupta, Arjun Nemani, Freya Mehta, Manan Goel, Saraansh Tandon, Shweta Sahoo, Ujwal Narayan and Yash Mathne for their help with this study. Thanks to Vatika

Harlalka for proof-reading this paper and ensuring there were as few errors as possible.

## References

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. i am borrowing ya mixing? an analysis of english-hindi code mixing in facebook. *In Proceedings of the First Workshop on Computational Approaches to Code Switching, EMNLP.*

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.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Jordi Carrera, Olga Beregovaya, and Alex Yanishevsky. 2009. Machine translation for cross-language social media. *PROMT Americas Inc.*

Khyathi Raghavi Chandu, Manoj Chinnakotla, Alan W. Black, and Manish Shrivastava. 2017. Webshodh: A code mixed factoid question answering system for web. In Gareth J.F. Jones, Séamus Lawless, Julio Gonzalo, Liadh Kelly, Lorraine Goeuriot, Thomas Mandl, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 104–111, Cham. Springer International Publishing.

Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using crf: Code-switching shared task report of msr india system. page 73–79. Association for Computational Linguistics.

Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(3):157–174.

David Crystal. 2011. *Internet linguistics: A Student Guide (1st ed.).* Routledge, New York, NY.

Brenda Danet and Susan C Herring. 2007. *The multilingual Internet: Language, culture, and communication online*. Oxford University Press on Demand.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text. *Proceedings of the 11th International Conference on Natural Language Processing*. pages 169–178.

Sakshi Gupta, Piyush Bansal, and Radhika Mamidi. 2016. Resource creation for hindi-english code mixed social media text. *The 4th International Workshop on Natural Language Processing for Social Media in the 25th International Joint Conference on Artificial Intelligence*.

Susan C Herring. 2003. Media and language change: Introduction. *Journal of Historical Pragmatics*, 4(1):1–17.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. *In Proceedings of Recent Advances in Natural Language Processing 2015 Organising Committee, Association for Computational Linguistics*. pages 239–248.

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. The COLING 2016 Organizing Committee.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The IIT bombay english-hindi parallel corpus. *Under review at LREC 2018, CoRR*, abs/1710.02855.

Carol Myers-Scotton. 1993. Common and uncommon ground: Social and structural factors in codeswitching. *Language in society*, 22(4):475–503.

Carol Myers-Scotton. 1997. *Duelling Languages: Grammatical Structure in Codeswitching*. Clarendon Press.

Dong Nguyen and A Seza Doğruöz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 857–862.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. "answer ka type kya he?": Learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*. pages 853–858. ACM.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury Choudhury, and Kalika Bali. 2016. Translating code-mixed tweets: A language detection based system. In *3rd Workshop on Indian Language Data Resource and Evaluation - WILDRE-3*.

Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Srivastava, Radhika Mamidi, and Dipti M Sharma. 2016. Shallow parsing pipeline for hindi-english code-mixed social media text. *HLT-NAACL The Association for Computational Linguistics*. pages 1340–1345

Rai Mahesh Kumar 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.

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. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.