



A Novel Approach for Effective Recognition of the Code-Switched Data on Monolingual Language Model

Ganji Sreeram, Rohit Sinha

Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati, Guwahati - 781039, India

{s.ganji, rsinha}@iitg.ernet.in

Abstract

Code-switching refers to the phenomena of mixing of words or phrases from foreign languages while communicating in a native language by the multilingual speakers. Code-switching is a global phenomenon and is widely accepted in multilingual communities. However, for training the language model (LM) for such tasks, a very limited code-switched textual resources are available as yet. In this work, we present an approach to reduce the perplexity (PPL) of Hindi-English code-switched data when tested over the LM trained on purely native Hindi data. For this purpose, we propose a novel textual feature which allows the LM to predict the code-switching instances. The proposed feature is referred to as code-switching factor (CS-factor). Also, we developed a tagger that facilitates the automatic tagging of the code-switching instances. This tagger is trained on a development data and assigns an equivalent class of foreign (English) words to each of the potential native (Hindi) words. For this study, the textual resource has been created by crawling the blogs from a couple of websites educating about the usage of the Internet. In the context of recognition of the code-switching data, the proposed technique is found to yield a substantial improvement in terms of PPL.

Index Terms: code-switching, factored language model, recurrent neural networks

1. Introduction

Code-switching is defined as the fusion of two or more distinct languages within the same utterance by a speaker [1, 2]. In recent past, the code-switching phenomenon has become widely accepted in multilingual communities, such as Cantonese-English [3], Mandarin-English [4], Spanish-English [5], etc. In India, after independence, though the Indian constitution declared Hindi as the primary official language, the usage of English was still continued as a secondary language for its dominance in administration, education and law [6, 7]. Thereby, creating a trend among the urban population to communicate in English for economic and social purposes. Over the years, substantial code-switching to English while speaking Hindi as well as other dominant Indian languages has become a common feature.

The switching between the languages can happen within the utterance or at its boundary. If the switching occurs within the utterance then it is referred to as the intra-sentential code-switching. Whereas, the switching occurring at the utterance boundary is referred to as the inter-sentential code-switching [8]. In this work, we attempt to address the intra-sentential code-switching in Indian language context. We consider the Hindi-English (Hinglish) code-switching phenomenon

Type 1	मुझे मेरा current account balance जानना है Grammarly एक advanced grammar and spell checker है
Type 2	अपने budget के अनुसार investments कर सकते हैं Market में बहुत से paid और free cdn उपलब्ध है

Figure 1: Example sentences showing the varying levels of intra-sentential code-switching. Type 1 and Type 2 refer to those cases where the code-switched English words carry high and low context information, respectively.

where the native language is Hindi and the foreign language is English [9]. In literature, there are works addressing the intra-sentential code-switching problem [10, 11, 12]. But in most of those works, the code-switched foreign language contained sequences of words having some contextual information [13, 14], thus a few works employed the interpolation technique over the LMs trained on the native and the foreign languages separately. In this work, we consider a case where the majority of the Hindi language utterances are code-switched with the English language words without much context information. In such cases, due to insufficient contextual information, the existing techniques become less effective in recognizing the code-switched data. Figure 1 shows a few example sentences of the intra-sentential code-switching phenomenon highlighting the variation in the contextual information present in the code-switched foreign words.

Code-switching is a phenomenon that happens more commonly in spoken form than in written form. Hence, the availability of code-switching text corpora is very scarce. To address these issues, one way is to tediously collect a huge amount of code-switching text corpus and train the language model using it as attempted in [15]. Alternatively, one can also explore augmenting the monolingual LM with the semantic and the syntactic information extracted from limited code-switching text corpus. We hypothesized that, if this code-switching information can be captured while training, the same could enhance the ability of the monolingual LM to recognize the code-switching test data effectively. For validating our hypothesis, we propose a novel textual feature which allows the LM to predict the code-switching instances along with the possible foreign word with which the native word has been code-switched. We also have developed a tagger that facilitates the automatic tagging of the code-switching instances and assigns an equivalent class of foreign (English) words to each of the potential native (Hindi) words.

The remainder of this paper is organized as follows: In Section 2, the motivation behind the proposed code-switching factor (CS-factor) and the procedure followed to develop the code-

switching tagger (CS-tagger) have been discussed in detail. The detailed description of the text corpora collected and the system tuning parameters involved in this study are described in Section 3. The evaluation results of the proposed CS-factor based RNN-LM in contrast with the existing approaches has been presented in Section 4. The paper is concluded in Section 5.

2. Motivation and Proposed Approach

The issue of code-switching has become a common feature in several multilingual communities. Hence, there is a rising demand for an automatic speech recognition (ASR) system that can handle the code-switched speech data. Code-switching cannot be characterized as the random mixing of the words or phrases from two or more languages. In fact, the switching between the languages appears to follow some broad syntactic rules. We can capture the information about the native words that are being code-switched with the foreign words. It is hypothesized that a monolingual LM that incorporates this information would be able to yield better recognition performance for the code-switching test data. Motivated by that, we have proposed a novel textual tagger that includes the code-switching information to the training data and the same is described in the following subsection.

2.1. Code-Switch Tagger (CS-tagger)

During code-switching, the foreign words are inserted into the native language sentences mostly without affecting its syntax as well as the semantics. To exploit this fact, we have proposed a textual feature that identifies the locations where the code-switching can potentially occur. It is referred to as *CS-factor* in this paper. When a monolingual LM is trained by employing the CS-factor along with the words and tested using the code-switched data, a significant improvement in terms of the perplexity is noted.

To introduce this CS-factor in the LM training data, a tagger has been developed. The steps followed for its creation are given below:

1. Create a Hinglish development (dev) set and their Hindi translated versions that cover the Hindi LM training set. While translating, the proper-nouns and the abbreviations are kept unchanged. An example of the same is shown below.

Hinglish	booch methodology दो development process को suggest करता है मेरा atm card खो गया है तो मैं अपने payment को कैसे रोक सकता हूँ क्या आप मुझे चालुक्य express का arrival time बता सकते हैं Class और object के बीच relationship क्या है
Hindi	booch पद्धति दो विकास प्रक्रियाओं का सुझाव करता है मेरा atm पत्रक खो गया है तो मैं अपने भुगतान को कैसे रोक सकता हूँ क्या आप मुझे चालुक्य द्रुतगामी का आगमन समय बता सकते हैं वर्ग और वस्तु के बीच रिश्ता क्या है

2. More than 70% of the Hinglish development sentences are found to have word-to-word correspondence with their translated Hindi versions. From the alignment of Hinglish-Hindi sentences, those words which undergo code-switching are marked as 'Yes' while the remaining are marked as 'No' as shown in the example below.

Hinglish	class	और	object	के	बीच	relationship	क्या	है
Hindi	वर्ग	और	वस्तु	के	बीच	रिश्ता	क्या	है
CS tag	Yes	No	Yes	No	No	Yes	No	No

3. Using the above code-switching information, a map has been developed that assigns an equivalent class of foreign (English) word(s) to each of the potential native (Hindi) words as shown below. It is worth highlighting

Hindi words	English words	CS Tag
वर्ग	class	Yes
वस्तु	object	Yes
रिश्ता	relationship	Yes
वस्तु	thing	Yes
atm	atm	No

that the developed map involves only 1640 words out of a total of 5651 words in the development set. This observation supports our hypothesis that the code-switching does not take place randomly rather follows syntactic rules.

4. Using the created map, the Hindi training data is tagged with the CS-factors and the resultant output is used for training the LM. The below example shows the output of the proposed CS-tagger.

Hindi training sentence: वर्ग और वस्तु के बीच रिश्ता क्या है
Output of CS-Tagger: W-वर्ग:S-Class:M-Yes W-और:S-और:M-No W-वस्तु:S-object:M-Yes W-के:S-के:M-No W-बीच:S-बीच:M-No W-रिश्ता:S-relationship:M-Yes W-क्या:S-क्या:M-No W-है:S-है:M-No

2.2. Factored Language Model

Factored language modeling techniques are used to incorporate morphological and linguistic information while training the LM [16, 17, 18]. In this technique, each word w_t in the vocabulary V is represented as a group of k factors denoted as: $f_t^1, f_t^2, \dots, f_t^k$. The factors can be either morphological features like stems, roots etc., or any other linguistic feature of that respective word. Compared to the n -gram scheme, the recurrent neural networks (RNNs) are found to be more efficient in modeling the long-term dependencies and the semantic information in the context of LMs [19, 20, 21]. In recent days, the RNNs are also used in training the factored LMs and have shown significant improvements in terms of recognition performances [22, 23].

Motivated by those abilities of RNNs, in this work, we have employed the RNN-based factored LM (F-RNNLM) to evaluate the proposed CS-factors in training the native (Hindi) LM. The network architecture of the F-RNNLM is given in Figure 2, where x , s , and y represents the input, hidden and the output layers respectively. The F-RNNLM predicts the posterior probability of the current word as

$$P(w_t|F(w_{t-1}), s_{t-1}) = P(w_t|F(w_{t-1}), s_{t-1}, c(w_t)) \times P(c(w_t)|F(w_{t-1}), s_{t-1}) \quad (1)$$

where $F(w_{t-1}) = [f_{(t-1)}^1, f_{(t-1)}^2, \dots, f_{(t-1)}^k]$ is the k -dimensional feature vector corresponding to the word w_{t-1} , $c(w_t)$ represents the class to which the word w_t belongs to and s_{t-1} refers to the previous context obtained from the hidden layer.

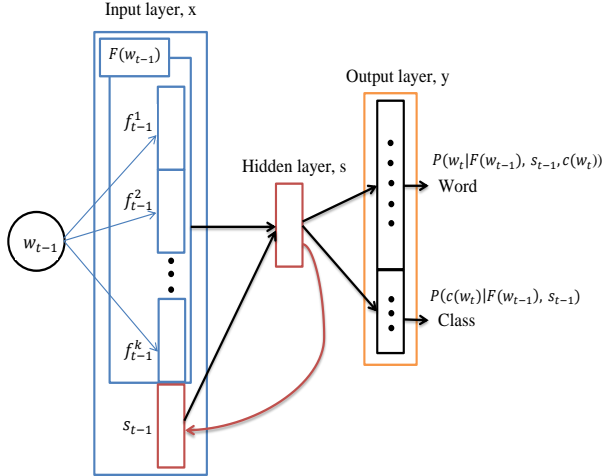


Figure 2: Architecture of the factored RNN-LM. The classes are derived by partitioning the vocabulary into groups based on the word counts and it helps reduce the search complexity.

3. Experimental Setup

This section describes the creation of Hinglish code-switched test, development and Hindi train text datasets and the LM tuning parameters used in experimentation.

3.1. Database preparation

The code-switched Hinglish text data is created by crawling few websites educating about the usage of Internet in day-to-day life [24, 25]. The crawled text data has been further processed and normalized into meaningful sentences by removing the emoticons, special characters, spaces, etc. Later, the normalized Hinglish sentences are translated to Hindi sentences with the help of *Google translate* [26], an online translation tool. In few cases, the Google translate has failed to generate correct Hindi translation. This might be due to insufficient Hindi vocabulary in the translator. In such cases, the Hindi translation has been manually done with the help of a few online Hindi vocabulary resources [27, 28]. It is to note that the abbreviations and the proper-nouns in the Hinglish sentences are left unchanged while translating them into Hindi sentences. The obtained Hindi data is partitioned into training, test and development datasets consisting of 1050, 105 and 435 sentences, respectively. Note that, the Hinglish sentences corresponding to 435 Hindi development sentences are used for training the CS-tagger while maintaining word-to-word correspondence with those of Hindi data set. To highlight the difference in the contextual information between the train and development Hindi datasets, the top ten most likely 3-grams are given in Figure 3. Also, the Hinglish sentences corresponding to 105 Hindi test sentences are used for evaluating the proposed approach. The salient details of the Hinglish and Hindi datasets created are summarized in Table 1.

The proposed and the contrast LMs for the code-switching task are trained using the Hindi training dataset. The Hindi test data is used for the parameters tuning while training the language models. The recognition performances of the trained LMs are evaluated on the Hinglish test data. Further, for the contrast purpose, a class-based RNN-LM and 5-gram LM are trained on the Hindi training data.

Table 1: Details of the vocabulary size and the word count of the train, test and dev sets involved in this study. The proper-nouns and abbreviations are left unchanged while translating Hinglish sentences to Hindi. Hence, we find few English words in the translated Hindi sentences.

Dataset	Data type	# Sentences	# Words		# Unique words	
			Hindi	Eng.	Hindi	Eng.
Train	Hindi	1050	15604	1036	1938	381
	Hinglish	105	1563	59	508	29
Test	Hindi	105	1088	533	284	263
	Hinglish	435	5498	153	2323	62
Dev	Hindi	435	3773	1874	1381	1004
	Hinglish	435	5498	153	2323	62

Training data 3-grams		Development data 3-grams	
Log likelihood	Word sequences	Log likelihood	Word sequences
-0.015	के बारे में	-0.032	सकते है </s>
-0.021	प्रदान करता है	-0.041	सकता है </s>
-0.032	किया जाता है	-0.051	किया जाता है
-0.037	जा सकता है	-0.061	देता है </s>
-0.041	<s> खोज यन्त्र	-0.079	जाता है </s>
-0.045	नीचे दिए गए	-0.079	<s> खोज यन्त्र
-0.045	घुंड़ी पर टक	-0.079	के बारे में
-0.045	<s> सहबद्ध विपणन	-0.096	कर सकते हैं
-0.051	के घुंड़ी पर	-0.096	करते हैं </s>
-0.051	<s> दलाली संयोजन	-0.124	करने के लिए

Figure 3: Differences in the contextual information present in the training and the development datasets. For highlighting the same, the top ten most likely 3-grams in both cases are listed.

3.2. Parameter tuning

The RNN-based language models used in the experiments are trained using the RNNLM toolkit [29]. These RNN-LMs are trained with a single hidden layer having 300 nodes and *sigmoid* as the non-linearity function. By conducting tuning experiments on Hindi test data, the parameter corresponding to the number of classes is set to be 50 and the variable corresponding to backpropagation through time (BPTT) is set as 5. Also, the 5-gram LM is trained using the SRILM toolkit [30] by setting the discount parameter to *kndiscount*.

4. Results and Discussion

The evaluation of the proposed CS-factors has been done in the context of Hinglish code-switching task. For this purpose, a 5-gram LM and both classes and factor-based RNN-LMs are developed using Hindi training data. The recognition performances of these LMs, in terms of perplexity (PPL), for Hinglish and Hindi test sets are evaluated. In each case, the k -fold cross-validation with $k = 3$ has been performed and their average is reported in Table 2. When the Hinglish test set is evaluated over the 5-gram LM and the normal class-based RNN-LM, a huge degradation in terms of PPL has been observed in comparison to that of the Hindi test set. It is attributed to the fact that whenever a foreign (English) words occur while testing, there is no contextual information associated with them. Note that, the OOV issue is addressed by adding all the English words present in the development data into the vocabulary of Hindi training data while developing the 5-gram and the normal RNN-LM.

Table 2: The recognition performances (in terms of perplexity) of the proposed CS-factors in the context of Hinglish code-switching task. Since, the Hindi test set is used for tuning the parameters while training Hindi LMs, those performances are only for reported reference purpose.

LM	Factor	Test data	PPL
5-gram	Word	Hinglish	334.20
		Hindi	80.11
RNNLM	Word	Hinglish	318.23
		Hindi	78.23
F-RNNLM	Word + CS	Hinglish	48.15
		Hindi	48.80

Whereas, when the Hinglish test set is evaluated over the factored RNN-LM trained using the code-switching information as factors, a significant reduction in PPL has been achieved. This is because the CS-factors capture the contextual information while tagging a list of native (Hindi) words with their respective code-switching instances along with their corresponding switched foreign (English) words. Thus, by employing the CS-factors as a feature along with the words while training the RNN-LM, the contextual information will be helpful for predicting the Hinglish word sequences.

For all kinds of LMs, the parameters are tuned to the Hindi test set, hence the Hindi test set performances given in Table 2 are only for reference purpose. From Table 2 we can see that the proposed CS-factor based RNN-LM has resulted in significant improvement in the recognition performance over the conventional RNN-LM. Apart from addressing the code-switching, the proposed approach also help in the modeling of Hindi words having the similar context. This is the reason behind the improvement in the PPL observed for the Hindi test data. For contrast purpose, Table 2 also lists the performances of the Hinglish and the Hindi test sets evaluated on the traditional 5-gram LM created using Hindi training data.

5. Conclusion

In this work, we have proposed a novel code-switching tagger (CS-tagger) for improving the recognition of Hinglish code-switched data. On using the CS-tags, the factored RNN-LM trained on Hindi text data has shown a significant reduction in terms of perplexity for Hinglish code-switched data. The proposed approach is found to be quite effective in modeling the contextual information and thus, handling the code-switched data. The significance of this work lies in two aspects. First, the approach attempts to enhance the ability of native language LM without the need to incorporate the code-switched data. Second, the proposed CS-tagger does not require a large amount of target code-switched data for the effective modeling.

In this study, the data involved in the creation of the native LM is relatively small. So, the findings of this study are required to be validated on a larger native dataset. Also, we aim to address the code-switching tasks corresponding to other Indian languages in future.

6. References

- [1] C. M. Scotton, "Comparing codeswitching and borrowing," *Journal of Multilingual & Multicultural Development*, vol. 13, no. 1-2, pp. 19–39, 1992.
- [2] J. J. Gumperz, *Discourse Strategies*. Cambridge University Press, 1982.
- [3] D. Li, "Cantonese-English code-switching research in Hong kong: A Y2K review," *World Englishes*, vol. 19, no. 3, pp. 305–322, 2000.
- [4] D.-C. Lyu, T.-P. Tan, E.-S. Chng, and H. Li, "An analysis of a Mandarin-English code-switching speech corpus: SEAME," *Age*, vol. 21, pp. 25–8, 2010.
- [5] A. Ardila, "Spanglish: an anglicized Spanish dialect," *Hispanic Journal of Behavioral Sciences*, vol. 27, no. 1, pp. 60–81, 2005.
- [6] A. Dey and P. Fung, "A Hindi-English Code-Switching Corpus." in *Proc. of Language Resources and Evaluation Conference (LREC)*, 2014, pp. 2410–2413.
- [7] S. Malhotra, "Hindi-English, Code Switching and Language choice in urban, uppermiddle-class Indian Families," *Kansas Working Papers in Linguistics*, 1980.
- [8] F. Grosjean, *Life with Two Languages: An Introduction to Bilingualism*. Harvard University Press, 1982.
- [9] K. Bhuvanagirir and S. K. Kopparapu, "Mixed language speech recognition without explicit identification of language," *American Journal of Signal Processing*, vol. 2, no. 5, pp. 92–97, 2012.
- [10] A. K. Joshi, "Processing of sentences with intra-sentential code-switching," in *Proc. of 9th conference on Computational Linguistics*, vol. 1, 1982, pp. 145–150.
- [11] B. H. Ahmed and T.-P. Tan, "Automatic speech recognition of code switching speech using 1-best rescoring," in *Proc. of International Conference on Asian Language Processing (IALP)*. IEEE, 2012, pp. 137–140.
- [12] H. Cao, P. Ching, T. Lee, and Y. T. Yeung, "Semantics-based language modeling for Cantonese-English code-mixing speech recognition," in *Proc. of 7th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2010, pp. 246–250.
- [13] D. C. Lyu, R. Y. Lyu, Y. C. Chiang, and C. N. Hsu, "Speech recognition on code-switching among the Chinese dialects," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 2006.
- [14] C. F. Yeh, C. Y. Huang, L. C. Sun, C. Liang, and L. S. Lee, "An integrated framework for transcribing Mandarin-English code-mixed lectures with improved acoustic and language modeling," in *Proc. of 7th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2010, pp. 214–219.
- [15] I. Hamed, M. Elmahdy, and S. Abdennadher, "Building a First Language Model for Code-switch Arabic-English," *Procedia Computer Science*, vol. 117, pp. 208–216, 2017.
- [16] K. Kirchhoff, J. Bilmes, and K. Duh, "Factored Language Models Tutorial," Dept of EE, University of Washington, Tech. Rep. UWEEETR-2007-0003, 2007.
- [17] J. A. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proc. of Conference on Computational Linguistics on Human Language Technology*, 2003, pp. 4–6.
- [18] J. Gebhardt, "Speech recognition on English-Mandarin code-switching data using factored language models," Master's thesis, Department of Informatics, Karlsruhe Institute of Technology, 2011.
- [19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

- [20] I. Oparin, M. Sundermeyer, H. Ney, and J. L. Gauvain, "Performance analysis of neural networks in combination with n-gram language models," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5005–5008.
- [21] X. Chen, X. Liu, Y. Qian, M. Gales, and P. C. Woodland, "CUED-RNNLM—An open-source toolkit for efficient training and evaluation of recurrent neural network language models," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6000–6004.
- [22] Y. Wu, X. Lu, H. Yamamoto, S. Matsuda, C. Hori, and H. Kashioaka, "Factored language model based on recurrent neural network," in *Proc. of International Conference on Computational Linguistics*, 2012, pp. 2835–2850.
- [23] H. Adel, N. T. Vu, F. Kraus, T. Schlippe, H. Li, and T. Schultz, "Recurrent neural network language modeling for code switching conversational speech," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8411–8415.
- [24] "ShoutMeHindi," [Online] <https://shoutmehindi.com>, accessed: 2017-09-30.
- [25] "Computing Notes in Hinglish," [Online] <https://notesinhinglish.blogspot.in>, accessed: 2017-09-30.
- [26] "Google Translate," [Online] <https://translate.google.com>, accessed: 2017-09-30.
- [27] "Department of official language," [Online] <http://www.rajbhasha.nic.in/hi/hindi-vocabulary>, accessed: 2017-09-30.
- [28] "Wiktionary," [Online] <https://hi.wiktionary.org/wiki>, accessed: 2017-09-30.
- [29] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "RNNLM—Recurrent neural network language modeling toolkit," in *Proc. of the ASRU Workshop*, 2011, pp. 196–201.
- [30] A. Stolcke *et al.*, "SRILM – An extensible language modeling toolkit," in *Interspeech*, vol. 2002, 2002, p. 2002.