Language Informed Modeling of Code-Switched Text

Khyathi Raghavi Chandu* **Thomas Manzini*** **Sumeet Singh*** **Alan Black**

Language Technologies Institute, Carnegie Mellon University
{kchandu,tmanzini,sumeets,awb}@cs.cmu.edu

Abstract

Code-switching (CS), the practice of alternating between two or more languages in conversations, is pervasive in most multilingual communities. CS texts have a complex interplay between languages and occur in informal contexts that make them harder to collect and construct NLP tools for. We approach this problem through Language Modeling (LM) on a new Hindi-English mixed corpus containing 59,189 unique sentences collected from blogging websites. We implement and discuss different Language Models derived from a multi-layered LSTM architecture. We hypothesize that encoding language information strengthens a language model by helping to learn code-switching points. We show that our highest performing model achieves a test perplexity of 19.52 on the CS corpus that we collected and processed. On this data we demonstrate that our performance is an improvement over AWD-LSTM LM (a recent state of the art on monolingual English).

1 Introduction

Code-switching (CS) is a widely studied linguistic phenomenon where two different languages are interleaved. This occurs within multilingual communities (Poplack, 1980; Myers-Scotton, 1997; Muysken, 2000; Bullock and Toribio, 2009). Typically one language (the *matrix language*) provides the grammatical structure for CS text and words from another language (the *embedded language*) are inserted. However, CS data is challenging to obtain because this phenomenon is usually observed in informal settings. Data obtained from online sources is often noisy because of spelling, script, morphological, and grammatical variations. These sources of noise make it quite challenging to build robust NLP tools (Çetinoğlu et al., 2016). Our goal is to improve LM for Hindi-English code-mixed data (*Hinglish*) where similar challenges are apparent. The task of language modeling is very important to several downstream applications in NLP including speech recognition, machine translation, etc. This is particularly important in domains that lack annotated data, such as code-switching, where the need to leverage unsupervised techniques is a must. We address the task of language modeling in CS text with a dual objective: (1) predicting the next word, and (2) predicting the language of the next word.

In addition to the techniques used for monolingual language modeling, providing information about the language is a key component in CS domain. Our main goal in this paper is to examine the effect of language information in modeling CS text. We approach this systematically by experimenting with ablations of encoding and decoding language IDs along with the word itself. In this way, the model implicitly learns the switch points between the languages. We achieve the least perplexity score using a combination of a language informed encoder and a language informed decoder.

The current material begins with a review of LM techniques for CS text in section 2. Then we describe our data collection and processing steps in Section 3 and model architecture in Section 4. Section 5 contains a brief quantitative and qualitative discussion of our observations and promising directions for future work. We then conclude in section 6.

2 Related Work

The increased reach of Internet and social media has led to proliferation of noisy CS data where earlier computational frameworks for code-switching, such as Joshi (1982); Goyal et al. (2003); Sinha and Thakur (2005); Solorio and Liu (2008a,b), are not readily applicable. In recent times, the community has focused on develop-

* These authors contributed equally

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 92–97
Melbourne, Australia, July 19, 2018.

ing a variety of NLP tools for CS data such language models by Li and Fung (2013, 2014); Adel et al. (2015, 2013a,b); Garg et al. (2017), POS taggers by Vyas et al. (2014); Jamatia et al. (2015); Çetinoğlu and Çöltekin (2016), automatic language identification by Jurgens et al. (2017); King and Abney (2013); Rijhwani et al. (2017); Jhamtani et al. (2014), prediction of code-switch points by Das and Gambäck (2014), sentiment analysis by Rudra et al. (2016) and also certain meta level studies that include understanding metrics to characterize code-mixing Patro et al. (2017); Guzmán et al. (2017). The idea of including language identifier vectors on the input and/or output side has become fairly common for other tasks as well, e.g. in Johnson et al. (2016) for machine translation, Ammar et al. (2016) for parsing, or Östling and Tiedemann (2016) for language modeling.

2.1 Code-Switched Language Models

There has been some recent focus on adapting existing language models for CS text. Li and Fung (2013, 2014) use a translation model together with the language model of the matrix language to model the mixed language. The search space within the translation model is reduced by linguistic features in CS texts like inversion constraint and functional head constraint (Sankoff and Poplack, 1981).

In another approach Adel et al. (2015), use a Factored Language Model (FLM) that includes syntactic and semantic features found in CS text that are indicative of a switch e.g. trigger words, trigger POS tags, brown cluster of function and content words that result in significant reduction in perplexity.

Another recent method called Dual Language Model (DLM) (Garg et al., 2017), combines two monolingual language models by introducing a '*switch*' token common to both languages. Predicting this word in either languages acts a proxy to the probability of a switch and the next word is then predicted using the LM of the language that was switched to.

Among neural methods, Adel et al. (2013a) use a Recurrent Neural Network based LM to predict the language of the next word along with the actual word to model CS text. Following on these intuitions, our models are built on top of the AWD-LSTM LM (Merity et al., 2017) that was chosen due to its accessibility and high performance (recently State of the Art) on the Penn-Tree Bank and Wikitext-2 dataset (Merity et al., 2016). Extensive work has been done on this model through investigation on relative importance of hyper-parameters (Merity et al., 2018).

Criteria	Train	Dev	Test
# Sentences	35513	11839	11837
Avg Length of Sentences	18.90	17.58	18.22
Multilingual Index	0.8892	0.8905	0.8914
Language Entropy	0.6635	0.6639	0.6641
Integration Index	0.3304	0.3314	0.3312
Unique Unigrams	35,769	18,053	19,330
Unique Bigrams	276,552	125,108	130,947
Unique Trigrams	553,866	219,098	229,967

Table 1: Hinglish Data Statistics

3 Data Analysis

Curating a reasonable dataset for CS text is an important challenge for researchers in this domain. To the knowledge of the authors, there is no benchmark CS corpus for language modeling as there is for English (Merity et al., 2016; Marcus et al., 1994). The two potential source choices to gather data include social media (such as Twitter and Facebook) and blogging websites. We decided to go with the latter due to comparatively lesser noise and availability of more descriptive text. Our CS LM data was collected after having crawled eight Hinglish blogging websites[1], that were returned by popular search engines (such as Google and Bing) with simple code-switched queries in the domains of health and technology. The topics covered in these CS texts include technical reviews of electronic and general e-commerce products as well as several health related articles.

These texts were tokenized at the sentence level over which we ran a language identifier. Language detection is performed both at the word level and also at the sentence level by treating the entire sentence as a sequence labeling problem. Naive Bayes and Hidden Markov Models with Viterbi Decoding were used respectively that gave an accuracy of around 97% on a subset of our data. Moreover, all the sentences that did not have at least one word each from both languages were discarded to channel our problem towards tackling intra-sentential code-switching. This resulted in a total of 59189 unique sentences. To estimate the quality and extent of mixing and frequency of switching in our data, we measured Multilingual index (M-Index), Language Entropy and Integration index (I-index) that were introduced in the domain of CS by Guzmán et al. (2017). These metrics along with other n-gram statistics over our data are presented in Table 1. A multilingual index of 1 indicates that there is equal extent of mix-

[1]Some Hinglish websites:
www.hinglishpedia.com,
www.hindimehelp.com,
www.pakkasolutionhindi.com

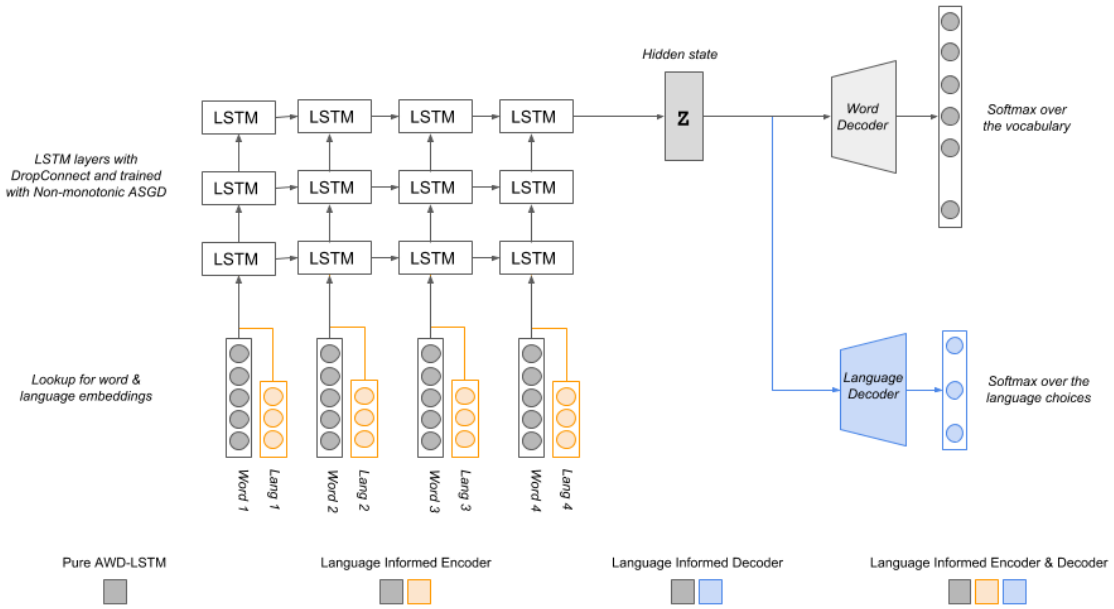

Figure 1: Various CS LM models that we explored in this work

ing from both the participating languages. As we can observe, the mixing is close to 0.8 which indicates that both Hindi and English are participating in the ratio 4:5. The metric itself does not reveal about which is the embedded language and which is the matrix language. Note that the CS metrics for each of the train, validate and test splits of the data are almost the same, indicating a similar extent of mixing in them.

4 Models and Experiments

There are a number of ways to frame the desire for humans to switch between languages (Skiba, 1997; Moreno et al., 2002), however, we view the human desire as out of scope for this work. Instead, our focus is on how we can incorporate linguistic information while training a statistical model for code-switched text. We discuss two main choices as to where we can introduce this information: either at the encoding stage or at the decoding stage of an RNN language model.

Given a CS sentence $\boldsymbol{X_{cs}} = (\boldsymbol{x^1}, \boldsymbol{x^2} \ldots, \boldsymbol{x^n})$ which has lexical level language sequence $\boldsymbol{L_{cs}} = (\boldsymbol{l^1}, \boldsymbol{l^2} \ldots, \boldsymbol{l^n})$, our model has to predict the word at the next time step. Note that this vector $\boldsymbol{l^i}$ is the language of the $\boldsymbol{ith}$ lexical item trained in concert with the model. This allows our model to encode the distributional properties of the language switching. We experimented with encoding and decoding the word and language embeddings for this task. $\boldsymbol{\theta_{E_X}}$, $\boldsymbol{\theta_{E_L}}$, $\boldsymbol{\theta_{D_X}}$ and $\boldsymbol{\theta_{D_L}}$ are the parameters for the word encoder, language encoder, word decoder and language decoder respectively.

We identify four different model architectures (Figure 1) that could be useful in training code-switched language models. In the first model, our baseline, we have a sequence of words and we are trying to predict the following word. This model is identical to running a traditional RNN language model on CS text.

For our baseline model we adapt the state-of-the-art language model, the AWD-LSTM, for this domain. This model is a 3 layered stacked LSTM trained via Averaged SGD with tied weights between the embedding and the softmax layer. There are several other important elements of this model, all of which are detailed in (Merity et al., 2017). The next word in this model is given by:

$$z = Encoder(\boldsymbol{X_{cs}}, \boldsymbol{\theta_E})$$

In our second model we extend our baseline such that we have a sequence of words and their language IDs and we try to predict the following word. In this and all the subsequent models, language ID is represented as a vector of length sixteen. This model can be seen as a factored language model operating with code-switched data. So, the next word in this model is given by:

$$Decoder(Encoder(\boldsymbol{X_{cs}}, \boldsymbol{\theta_{E_X}}), \boldsymbol{\theta_{D_X}})$$

In our third model we take a sequence of words as an input and attempt to predict both the language and the value of the following word. The next word in this model is given by:

$$Decoder(Encoder(\boldsymbol{X_{cs}}, \boldsymbol{\theta_{E_X}}) \bigoplus Encoder(\boldsymbol{L_{cs}}, \boldsymbol{\theta_{E_X}})), \boldsymbol{\theta_{D_X}})$$

Model/Data	Train	Dev	Test
Base AWD-LSTM Model	10.08	19.73	20.92
Language Aware Encoder AWD-LSTM	10.07	19.00	20.18
Language Aware Decoder AWD-LSTM	11.60	20.72	22.01
Language Aware Encoder & Decoder AWD-LSTM	9.47	18.51	19.52

Table 2: Perplexity scores of different models

In our fourth model we take a sequence of words and their corresponding language IDs as input and attempt to predict both the language and value of the subsequent word. In our third and fourth models we operate with two loss values being calculated for (one for the word error, and one for the language error multiplied by 0.1) and gradients for both losses are propagated through the network and are used to update the weights.

5 Results and Discussion

We trained 4 different models based on the description in Section 4. The results of these experiments are presented in Table 2. We observe that the Language Aware Encoding and Decoding with the AWD-LSTM gives the least perplexity. This aligns with our hypothesis that providing language information of the current word at encoding and enabling the model to decode the language of the next word allows the model to learn a higher level context of switch points between the languages.

5.1 Challenges and Future Work

Robustness of the language model also depends on the diversity of context in which the words co-occur. Since most of the articles belong to the topics of e-commerce, latest technology and health, this may be affected. Hence, we plan to use pre-trained word embeddings based on large monolingual corpora after aligning the embedding spaces of both the participating languages such as MUSE embeddings (Conneau et al., 2017). However, due to the non-standardized spellings in the *romanized* Hinglish text, most words that are incorrectly transliterated will not be found in the MUSE embeddings and such errors from transliteration will propagated through the subsequent parts of model. To avoid this, we plan to extend this work by using character encodings in future. Incorporating factors beyond language such as parts of speech, and sentence level features like root words or code-switching metrics could be another direction for future work. Incidentally, the hyper-parameters for our model were tuned on the Wikitext-2 dataset and it would be interesting to tune them on the Hinglish data itself. Lastly, and arguably most importantly, the accumulation and release of additional CS data would be a significant contribution to this field. Much of the work involved in this project was to properly clean, parse, and represent the CS data that was scraped from the online sources discussed above that could not be released because of copyright concerns. These sources remain limited in topic and variation and additional sources of CS data would be the best way to improve how well our model can generalize.

6 Conclusion

We hypothesize that incorporating the information of language aids in building more robust language models for code-switched text. This is substantiated by experimenting with different combinations of providing the language of the current word as input and decoding the language of the next word along with the word itself. We conclude that we are able to improve the State-of-The-Art language model for monolingual text by both explicitly providing the language information and decoding the language of the next word to perform this task for CS domain. We treat this problem as a multi-task learning problem where the same embedding and LSTM layers are shared. These two comparable tasks are predicting the next word and predicting the language of the next word. So far, our best test perplexity is 18.51 on development and 19.52 on test sets. This is in comparison to the baseline model which is 19.73 and 20.92 on development and test sets respectively.

We believe that further research can be done to not only improve perplexity, but to also improve the quality of the training and testing dataset. Language models are a core element in multiple tasks, from speech recognition to machine translation and we hope that this work will support future research into the development of such NLP tools for CS domain.

Acknowledgments

We would like to thank our reviewers for their insightful comments. We would also like to thank Graham Neubig at our institute who gave valuable feedback throughout the course of this work.

References

Heike Adel, Ngoc Thang Vu, Katrin Kirchhoff, Dominic Telaar, and Tanja Schultz. 2015. Syntactic and semantic features for code-switching factored language models. *IEEE Transactions on Audio, Speech, and Language Processing* 23(3):431–440.

Heike Adel, Ngoc Thang Vu, Franziska Kraus, Tim Schlippe, Haizhou Li, and Tanja Schultz. 2013a. Recurrent neural network language modeling for code switching conversational speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pages 8411–8415.

Heike Adel, Ngoc Thang Vu, and Tanja Schultz. 2013b. Combination of recurrent neural networks and factored language models for code-switching language modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 206–211.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Many languages, one parser. *arXiv preprint arXiv:1602.01595* .

Barbara E Bullock and Almeida Jacqueline Ed Toribio. 2009. *The Cambridge handbook of linguistic code-switching.*. Cambridge University Press.

Özlem Çetinoğlu and Çağrı Çöltekin. 2016. Part of speech annotation of a turkish-german code-switching corpus. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*. pages 120–130.

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. *arXiv preprint arXiv:1610.02213* .

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087* .

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text .

Saurabh Garg, Tanmay Parekh, and Preethi Jyothi. 2017. Dual language models for code mixed speech recognition. *arXiv preprint arXiv:1711.01048* .

P Goyal, Manav R Mital, A Mukerjee, Achla M Raina, D Sharma, P Shukla, and K Vikram. 2003. A bilingual parser for hindi, english and code-switching structures. In *10th Conference of The European Chapter*. page 15.

Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. *Proc. Interspeech 2017* pages 67–71.

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 the International Conference Recent Advances in Natural Language Processing*. pages 239–248.

Harsh Jhamtani, Suleep Kumar Bhogi, and Vaskar Raychoudhury. 2014. Word-level language identification in bi-lingual code-switched texts. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558* .

Aravind K Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Proceedings of the 9th conference on Computational linguistics-Volume 1*. Academia Praha, pages 145–150.

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 51–57.

Ben King and Steven Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1110–1119.

Ying Li and Pascale Fung. 2013. Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pages 7368–7372.

Ying Li and Pascale Fung. 2014. Code switch language modeling with functional head constraint. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, pages 4913–4917.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '94, pages 114–119. https://doi.org/10.3115/1075812.1075835.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182* .

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. An analysis of neural language modeling at multiple scales. *arXiv preprint arXiv:1803.08240* .

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR* abs/1609.07843. http://arxiv.org/abs/1609.07843.

Eva M Moreno, Kara D Federmeier, and Marta Kutas. 2002. Switching languages, switching palabras (words): An electrophysiological study of code switching. *Brain and language* 80(2):188–207.

Pieter Muysken. 2000. *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Robert Östling and Jörg Tiedemann. 2016. Continuous multilinguality with language vectors. *arXiv preprint arXiv:1612.07486* .

Jasabanta Patro, Bidisha Samanta, Saurabh Singh, Abhipsa Basu, Prithwish Mukherjee, Monojit Choudhury, and Animesh Mukherjee. 2017. All that is english may be hindi: Enhancing language identification through automatic ranking of the likeliness of word borrowing in social media. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 2264–2274.

Shana Poplack. 1980. Sometimes ill start a sentence in spanish y termino en espanol: toward a typology of code-switching1. *Linguistics* 18(7-8):581–618.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1971–1982.

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding language preference for expression of opinion and sentiment: What do hindi-english speakers do on twitter? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1131–1141.

David Sankoff and Shana Poplack. 1981. A formal grammar for code-switching. *Research on Language & Social Interaction* 14(1):3–45.

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.

Richard Skiba. 1997. Code switching as a countenance of language interference. *The internet TESL journal* 3(10):1–6.

Thamar Solorio and Yang Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 973–981.

Thamar Solorio and Yang Liu. 2008b. Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1051–1060.

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.