Evaluating and Developing Methods of Generating Code-Switched Data
07-300, Fall 2021

Abhishek Vijayakumar
https://inkyubeytor.github.io/category/code-switching.html

November 11, 2021

1 Project Description

I will be working with Professor Alan W. Black on the evaluation and development of methods of generating synthetic code-switching text.

Code-switching is the act of switching between multiple languages in a single instance of speech or text. This is a common behavior for bilingual speakers. Code-switching has become increasingly commonly used in informal digital communication, such as in texting or on social media. As such, there is a rapidly growing need for language technologies addressing code-switched data for tasks such as translation, information retrieval, and sentiment analysis.

However, existing models, even multilingual models, do not perform well when faced with code-switched input, and training new state-of-the-art models requires large amounts of code-switched data. As code-switching data is less common than monolingual data and often occurs in private (e.g. texting) or noisy (e.g. Twitter) contexts, it is extremely difficult to collect a sufficient amount of real code-switched data. An alternative solution rapidly gaining traction in the research community is the generation of synthetic code-switched data.

My project will focus on the evaluation of existing methods of synthetic code-switched data generation. There are two general types of generation method: non-neural "substitutive" methods, which translate portions of input phrases [8] [7] [4], and neural "generative" methods, which generate sentences entirely from scratch [5] [6] [2]. We will evaluate several techniques in both categories with several levels of available training resources in order to see how each technique's performance depends on the quality and quantity of the data available to it. The evaluation will begin with intrinsic evaluation of distributional properties of the generated text. We will then evaluate the generated text by using it to fine-tune a multilingual BERT (mBERT) model for a suite of downstream tasks in the GLUECoS evaluation [3]. The results of this evaluation will guide future work with code-switched languages by allowing researchers to determine what techniques are most appropriate for working with new code-switched languages based on the resources they have available. If this evaluation is successful and we identify a weakness in an existing generative method (such as not fully taking advantage of a possible resource), the next step in this project is improving that generative method with respect to its evaluations.

A major challenge of this work will be developing a consistent and fair evaluation scheme with respect to the generative methods, as the ways they use resources and generate data are different. For example, substitutive methods translate fragments of input sentences to create output sentences, while neural generative methods create entire sentences. As the goal of this

project is to evaluate method performance relative to available resources, it is imperative that the chosen evaluation methods provide a good representation of downstream task performance, regardless of the way the method operates.

2 Project Goals

What do you hope to accomplish? What metrics will be used to evaluate the success of your project?

2.1 75% Project Goal

- Develop a suite of distributional evaluations for code-switched text corpora.
- Implement or acquire code for at least 6 different methods across both the non-neural and neural categories.
- Conduct distributional evaluations on all available methods.

2.2 100% Project Goal

- Conduct GLUECoS evaluations using all available methods.
- Evaluate the performance of methods that depend on real code-switching data with different qualities and quantities of code-switching data.
- Identify distributional evaluations that correlate highly with GLUECoS evaluations.
- Develop a set of recommendations based on available resources to guide researchers in choosing synthetic data generation methods.

2.3 125% Project Goal

- Extend evaluations to language pairs besides Hindi-English and Spanish-English (the language pairs of the GLUECoS benchmark).
- If possible, improve an evaluated technique with respect to the language data resource costs required to achieve a given level of performance.

3 Project Milestones

3.1 1st Technical Milestone for 07-300

I plan to implement or acquire code for at least 4 generative methods. I also plan to construct the datasets I need for method training.

3.2 February 1st

I plan to select at least 4 different distributional evaluations to implement. I am currently looking at BERTScore, Burstiness, Code-Mixing Index, Span Entropy, and Diversity as possible candidates. Of these, I plan to implement at least one evaluation across all methods.

3.3 February 15th

I plan to finish the implementation of at least 3 more distributional evaluation methods. I plan to run these evaluations on the generative methods I have available.

3.4 March 1st

I plan to implement the GLUECoS evaluation locally, so I do not have to repeatedly submit requests to Microsoft's repository to perform evaluations.

3.5 March 15th

I plan to run the GLUECoS evaluation on the methods I have available. Note that this deliverable is smaller due to spring break.

3.6 March 29th

I plan to implement additional generative methods. As I am doing this, I also plan to re-run GLUECoS evaluations against methods trained with different resource levels.

3.7 April 12th

I plan to continue implementing and evaluating additional generative methods.

3.8 April 26th

I plan to consolidate my evaluations during this period. I will identify any distributional evaluations strongly correlated with GLUECoS score. I will attempt to synthesize evaluation results into a consistent set of guidelines and summaries of method performances.

4 Literature Search

I have read several papers on code-switching generative methods, and I have selected 6 such papers to target for implementations to test [8] [7] [4] [5] [6] [2]. I have also read background materials on code-switching in general [1].

5 Resources Needed

The work will be done primarily in Python. The deep learning frameworks involved are both PyTorch and Tensorflow, which can be installed via Anaconda. I will also be using pretrained BERT models via Hugging Face. As a large portion of the project consists of evaluating existing methods, I have also requested these codebases from the authors of the relevant papers, and have obtained access to some of them.

The hardware required is entirely compute resources. I have been given access to the `k-baees` machine by my mentor, which contains 2 Nvidia GTX 1070s. These will be sufficient compute resources for the project.

The final type of resource needed is language data. I have been given some data from the aforementioned paper authors. In addition, I have begun collecting additional data from code-switching resources online, which I will be processing into a dataset usable for my project and similar projects in the future.

References

[1] Gayatri Bhat, Monojit Choudhury, and Kalika Bali. Grammatical constraints on intra-sentential code-switching: From theories to working models, 2016.

[2] Khyathi Raghavi Chandu and Alan W. Black. Style variation as a vantage point for code-switching. In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 4761–4765. ISCA, 2020.

[3] Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. Gluecos: An evaluation benchmark for code-switched NLP. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3575–3585. Association for Computational Linguistics, 2020.

[4] Garry Kuwanto, Afra Feyza Akyürek, Isidora Chara Tourni, Siyang Li, and Derry Wijaya. Low-resource machine translation for low-resource languages: Leveraging comparable data, code-switching and compute resources. *CoRR*, abs/2103.13272, 2021.

[5] Bidisha Samanta, Sharmila Reddy, Hussain Jagirdar, Niloy Ganguly, and Soumen Chakrabarti. A deep generative model for code switched text. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5175–5181. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[6] Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169, Online, August 2021. Association for Computational Linguistics.

[7] Jitao Xu and François Yvon. Can you traducir this? machine translation for code-switched input. *CoRR*, abs/2105.04846, 2021.

[8] Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. CSP:code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636, Online, November 2020. Association for Computational Linguistics.