Adversarial attacks on language models

Alexey Birshert
Higher School of Economics
Faculty of Computer Science
Moscow, Russia
adbirshert@edu.hse.ru

***Abstract*—Code-switching, alternating two or more languages in a sentence or discourse, is a common phenomenon in multilingual societies. The absence of training data with code-switching is one of the main problems in developing multilingual neural network models in natural language processing. To solve data shortages, we suggest a revolutionary approach to generate code-switching texts automatically. This approach utilizes adversarial training of a Transformer based language model. We plan to achieve higher metrics values than existing methods.**

***Index Terms*—code-switching, language models, slot-filling, natural language processing**

I. Introduction

We are researching natural language processing, more precisely the multilingual text corpora analysis and such a phenomenon as code-switching. Code-switching occurs in multilingual communities around the world and ordinary texts on the Internet. It consists of mixing and using two or more languages within one phrase or sentence. In our work, we set ourselves to solve the problem of generating realistic texts with code-switching. Such texts' artificial creation would improve the existing datasets for classic multilingual tasks like recognizing user intent or question-answer systems. It would enhance language models' quality and make them more resistant to code-switching.

For data generation, we would construct a language model based on Transformer architecture [1]. To train such a model, we plan to use adversarial attacks. As the main related work, we use articles [2] and [3]. The [2] article describes an approach to training language models based on Transformer architecture for a similar task - replacing words in a sentence without losing grammatical correctness while deceiving the classifier. The [3] article also describes another approach to generating code-switching texts using LSTM architecture. Our idea is revolutionary, as all previous methods use radically different language model architectures. Our research's primary goal is to study possible variants of training language models based on Transformer architecture. The main expected result is developing a method for training such language models to generate realistic data with code-switching, which is superior in quality to existing solutions. First, we will dive into the literature on similar topics in more detail; then, we will discuss the planned experiments and methods. In the end, we will summarize the intended results of this research.

II. Literature review

Code-switching, the mixing of languages within a single sentence or discourse, is a well-known process in multilingual communities worldwide. With the increasing globalization of the world and growing mixing of peoples, especially on the Internet, more and more code-switching data is generated. In our research, we try to build a model capable of generating realistic code-switching data. The main idea behind producing artificial code-mixed texts is to provide training data for classical natural language processing tasks. To construct and train such a model and to conduct various experiments, we need to understand how to make the following steps:

- We need to find a decent sizeable text corpus to fulfil our needs in training data. We looked especially for question answering datasets to generate code-switching data for question answering systems. One can find the most recent work describing and comparing datasets in the [4] article. Based on this article, we have chosen several intent-prediction datasets: BANKING77, CLINK150 and HWU64. Each of the datasets contains many utterances with hundreds of different intents to classify, which would be extremely useful for our future training algorithm.
- We need to develop an idea of generating code-switching data with a mask-filling language model. As we conduct bleeding-edge research in natural language processing, we have no examples of generating code-switching data with the type of language model we are willing to use. So, we adopt other natural language processing field of knowledge methodology - we base our ideas on the recent article on adversarial attacks on language models. The main idea of [2] is to make the language model select some tokens without changing the meaning of the sentence while keeping it grammatically correct and deceiving a classifier. Authors utilize the Deep Levenstein model for measuring the distance between sentences and a trained intent classifier for intent classification. These two parts combine the loss function.
- We need to acknowledge how to apply semantic constraints to the code-switching data generation task. Authors describe in the recent article [3] how they choose tokens to replace with ones from the other language. They are utilizing a word alignment model for mixing languages between sentences. LSTM model architecture is a base model for their code-switching data generating

framework.

- We need to come up with some methods of evaluating code-switching data corpora. The [5] article describes many different metrics that I can use to estimate trained models' quality. We have chosen to use the main ones in our work: Code-Mixing Index, Average switch-points, and Language Entropy.

III. Methods

We are researching how a slot-filling language model can generate code-mixed data. We plan to conduct several experiments in which we would train and evaluate different language models. In these experiments, we would use different model architectures and training approaches. As a result of these experiments, we aim to develop a method of training a slot-filling language model for generating realistic code-switched texts.

We plan to research several slot-filling language models' architectures in our work - XLM [6], XLM-RoBERTa [7] and BERT [8]. These models are multilingual models based on the Transformer neural network architecture. They have some differences in their training process, but the whole idea is more or less common, so it would be relatively simple to conduct experiments comparing these models. The main problem with these experiments is that all the models have different preparation processes of input sequences and very different vocabularies. We intend to solve this problem during our search.

We also purpose to research whether one can utilize the [2] approach in adversarial attacking a language model to train it to predict wanted tokens. As the primary model archetype, we have a neural network that fills "slots" in sentences. As an input, it takes a sentence with one word changed to the special mask token. As an output, it generates a probability distribution over tokens' vocabulary. Models are trained on the data collected on the Internet, such as Wikipedia and books. These sources are mostly monolingual. The probability distribution over tokens that are not from the main sentence's language is very close to uniform, and every token probability is very close to zero. We want to train the model in such a way that it would result in increasing other languages probabilities. We also want the model to generate tokens close to the original sentence's masked token in terms of meaning. It would make this new sentence code-switched and keep its intent.

To train a language model in the way described in the previous paragraph, we would use a complex loss function. It would consist of two components. First - we want to keep the meaning of the original sentence. One can calculate the distance between sentences in the following way - calculate embeddings of the original sentence and the generated one, and calculate the distance between the embeddings. We will refer to it as the "distance between sentences" loss compound - $Distance(S_o, S_g)$. For calculating sentence embeddings, we would utilize an embedding model based on XLM-RoBERTa. Second - we want to keep the classifier score for the intent. For this, we would process the original and the generated sentences through the classifier and calculate the difference. We will refer to it as the "classifier" loss compound - $|C(S_o) - C(S_g)|$. We would utilize a simple recurrent neural network for sequence classification.

$$Loss = Distance(S_o, S_g) + \alpha\,|C(S_o) - C(S_g)|$$

We project to experiment with changing the training process. In some experiments, we would try to determine how both loss function parts influence the overall performance. We would train models with only one part of the loss function and change the scale (α) between parts. In other experiments, we would try to determine how to choose tokens to mask. Mainly all current approaches base on linguistic semantic constraints - the goal is to generate realistic code-switching data. The insertions could be words or more significant constituents, and they would comply with the grammatical frame of the language. However, random word insertions could lead to the formation of unnatural code-mixed sentences, which are very rare in practice.

We plan to utilize the following pipeline for experiments - a neural network is trained on a question-answering dataset with some loss function and then evaluated with a set of different metrics. This approach would allow us to determine how each component of training affects the final result.

IV. Results anticipated

The research's main expected result is a training algorithm for the transformer type neural networks to generate realistic code-switching data. A trained model should be capable of generating multilingual code-mixed training data for the question-answering task. It also should achieve higher metrics values than existing methods.

V. Conclusion

Summing up, this study would help understand whether it makes sense to train a Transformer based slot-filling language model to generate code-switching texts. In the course of the research, we would carry out a series of experiments that would show the influence of individual parts of the learning process on the final result. The model should surpass the existing approaches in terms of the quality of the generated text in various metrics.

This research should help classical language models become more robust and work better with multilingual texts. Further research on the topic of code-switching can use the resulting model for conducting various experiments.

In the future, we plan to experiment with classical language models fine-tuned on the generated data and improve their quality on various question-answering benchmarks.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. "Attention Is All You Need." arXiv preprint arXiv:1706.03762v5.

[2] I. Fursov, A. Zaytsev, N. Kluchnikov, A. Kravchenko. 2020. "Differentiable Language Model Adversarial Attacks on Categorical Sequence Classifiers." arXiv preprint arXiv:2006.11078.

[3] Deepak Gupta, Asif Ekbal, Pushpak Bhattacharyya. 2020. “A Semi-supervised Approach to Generate the Code-Mixed Text using Pre-trained Encoder and Transfer Learning.“ Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2267–2280

[4] Shikib Mehri, Mihail Eric, Dilek Hakkani-Tur. 2020. “DialoGLUE: A Natural Language Understanding Benchmark for Task-Oriented Dialogue.“ arXiv preprint arXiv:2009.13570v2.

[5] Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, Monojit Choudhury. 2020. “GLUECoS : An Evaluation Benchmark for Code-Switched NLP.“ arXiv preprint arXiv:2004.12376v2.

[6] Guillaume Lample, Alexis Conneau. 2019. “Cross-lingual Language Model Pretraining.“ arXiv preprint arXiv:1901.07291.

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov. 2020. “Unsupervised Cross-lingual Representation Learning at Scale.“ arXiv preprint arXiv:1911.02116v2.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.“ arXiv preprint arXiv:1810.04805v2.

WORD COUNT: 1511