Report

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Code-Mixed Text Generation

Hindi-English

Introduction to NLP

Spring 23

Team

Aparna Agrawal (2021121007)

Shreya Patil (2021121009)

Abstract

Text generation is a highly active area of research in the computational linguistic community.Code-mixing, the phenomenon of mixing two or more languages in a single utterance, is a common occurrence in multilingual communities worldwide. With the growing interest in developing natural language processing systems that can handle code-mixed text, recent research has focused on generating code-mixed text using neural models. However, generating high-quality code-mixed text is a challenging task due to the complex nature of language mixing.

In this study, we explore the use of curriculum training to fine-tune multilingual Transformers like mT5 and mBert. We also use a dependency-free strategy for producing code-mixed texts from bilingual distributed representations that we utilize for enhancing language model performance due to the dearth of training data for code-mixing. We apply a curriculum learning strategy, in particular, where we first train the language models on synthetic data, then on gold code-mixed data as suggested by the paper https://arxiv.org/pdf/2105.08807.pdf

Keywords: CodeMix, multilingual, fine-tuning, synthetic ,Real,Transformers

Contents

1. Introduction

When a multilingual person speaks to others in two or more languages, this is known as "code mixing." For multilinguals, it is the most natural method of communication.Code mixing is the practise of incorporating phrases, words, and morphemes from one language into an utterance from another. Code mixing is frequently seen on social media platforms like Twitter and Facebook. It is critical to create systems to process such text as social media usage soars and code-mixed data usage grows as a result.Code mixing is the practise of incorporating phrases, words, and morphemes from one language into an utterance from another.To illustrate both directions, we provide Figure 1. The Figure presents sample translation pairs for Hinglish to English as well as English to Hinglish. Code mixing is frequently seen on social media platforms like Twitter and Facebook. It is critical to create systems to process such text as social media usage soars and code-mixed data usage grows as a result.Our task is translation of Hindi-English parallel sentence pairs to Hindi-English code mixed sentences. The challenges for solving this task include: (1) lack of Hindi data in roman script (2)informal writing style, (3) paucity of English-Hinglish parallel data. The English to Hinglish translation direction is a less researched study problem when compared to Hinglish to English translation. Translation from English to Hinglish can be useful in many situations. It can be used, for instance, to develop interesting conversational agents that imitate the code-mixing customs of a human user. Creating training data for some downstream applications, including token-level language detection, is another use of the Hinglish data that results. for this, we adopt a curriculum training method to finetune multilingual transformers as described in https://arxiv.org/pdf/2105.08807.pdf.As suggested in the paper we makes use of code-mixed data produced artificially and a multilingual text-to-text Transformer model More specifically, our system utilizes the state-of-the-art pre-trained multilingual generative model, mT5 (a multilingual variant of "Text-to-Text Transfer Transformer" model as a backnbone.

Hinglish to English translation (Dhar et al. (2018), Srivastava and Singh (2020))		
Hinglish: Hi there! Chat ke liye ready ho?	→	**English**: Hi there! Ready to chat?
Hinglish: isme kids keliye ache message hein, jo respectful hein sabhi keliye	→	**English**: It does show a really good message for kids, to be respectful of everybody
English to Hinglish translation (our task)		
English: Maybe it's to teach kids to challenge themselves	→	**Hinglish**: maybe kida ko teach karna unka challenge ho saktha hein
English: It's seem to do OK on rotten tomatoes I got a 79%	→	**Hinglish**: ROTTEN TOMATOES KA SCORE 79% DIYA HEIN JO OK HEIN KYA

Figure 1: Examples of translation pairs for the machine translation tasks from English to Hinglish and vice versa.

2. Background

Code-mixed text analysis is becoming more and more popular as a research topic, but there is one obstacle that has prevented the development of such works: a shortage of data. Code Mixed text generation is a relatively new problem, and so is its initial stage.

Some of the work uses linguistic theories to syntactically build code-mixed text using parallel monolingual corpora of two languagesThe Equivalence Constraint Theory states that code-mixing can only occur at parts of the text where the surface structures of two languages overlap, and the grammatical rules of both languages are followed. In contrast, the Matrix Language Theory separates the two languages into a base language and a second language. The rules of the base language are followed, and parts of the second language replace corresponding parts of the base language when grammatically feasible. The Matrix Language Theory thus attempts to address the limitations of the Equivalence Constraint Theory.This method is addressed in https://aclanthology.org/2021.eval4nlp-1.20/.
Systems for code mixture generation have also been created using Deep Learning and Neural Networks. The issue of text production in these systems has been framed as a machine translation challenge, where monolingual text is translated to code-mixed language.
The task of translation has gotten considerably easier with the advent of multilingual models like mT5 (Xue et al., 2020), mBART, indicBART (Dabre et al., 2021), etc. as these models understand both languages and have been proved to outperform earlier models in various workshops.

Multilingual T5 (mT5) is a massively multilingual pretrained text-to-text transformer model, trained following a similar recipe as T5 mT5 is pretrained on the mC4 corpus, covering 101 languages:

BART multilingual base model: Pretrained model on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective

we exploits this model by curriculum training. which is first finetuning the model on synthetic data and then again on real dataset. We observe that the mT5 model finetuned on the code-mixed data generated by the method proposed based on bilingual word embeddings followed by finetuning on gold data achieves a BLEU score of 17.43.

3. Data

- OPUS100
  we used 34147 of the training set of OPUS100 to generate synthetic data. out of which 21853 was for training ,5464 validation, 6830 test dataset.
- PHNC(A Parallel Hinglish Social Media Code-Mixed Corpus for Machine Translation) Tweeter data.
  we used the PHNC dataset to finetune over the real dataset. we divide PHNC dataset into 8791 training data,2198 validation data, and 2748 test data.

Data	Size_max
Synthetic(Train)	21853
Synthetic(val)	5464
human Generated (Train)	8791
human Generated (val)	2198
human Generated (test)	2748

Human Generated is PHNC data and Synthetic data is generated by us using OPUS parallel dataset.

4. Approach

4.1 Curriculum Learning

Curriculum learning is a machine learning technique that involves designing a curriculum or a sequence of tasks that the model is trained on, where each task is carefully chosen to gradually increase the difficulty of the learning problem. The goal of curriculum learning is to improve the speed and quality of the learning process by providing the model with a structured learning experience. In case of code-mix generation, it is particularly helpful due to the lack of reliable data. We implement curriculum learning using the method proposed teh paper referenced above. The model is first trained on a synthetic dataset and then fine-tuned on a dataset of real codemixed data.

4.2 Generating Synthetic Data

To generate a synthetic dataset, we begin with a substantial amount of bitext from a particular pair of languages (LG1 and LG2), such as English and Hindi in this case, for which we want to produce code-mixed data. Let $B_i = \{x_i, y_i\}$ represent the bitext data, where x_i and y_i are sentences in LG1 and LG2, respectively. Let $ngrams(n, x_i, y_i)$ denote the collection of distinct n-grams in x_i and y_i. We define $cumulative\_ngrams(n, x_i, y_i) = \bigcup_{j=1}^{n} ngrams(j, x_i, y_i)$ as the cumulative set of unique n-grams in the pair of sentences x_i and y_i.

Next, we randomize the n-grams in the cumulative set and create a shuffled code-mixed sentence by concatenating the randomized set with n-grams that are separated by spaces. For example, suppose we designate LG1 as English and LG2 as Hindi. A sample bitext instance could be "I've never seen it" (x_i) and "mai kaise jantna hu" (y_i). The set of unique 1-grams would be *['how', 'do', 'i', 'know', "]*, which is represented as $ngrams(1, x_i, y_i)$ (we use different tokenization in implementation). Then, $cumulative\_ngrams(2, x_i, y_i)$ would correspond to *['i_know', 'janta', 'kaise_janta', 'how_do', 'do_i', 'mai', 'how', 'janta_hu', 'hu', 'know', 'i', 'kaise', 'mai_kaise', 'do']*. We can create a shuffled code-mixed sentence such as "*i_know mai do_i mai_kaise kaise_janta how_do how hu jaanta janta_hu know i do kaise*". We produce one shuffled code-mixed sentence for each bitext instance, which results in a shuffled code-mixed corpus. To generate embeddings for n-grams in both languages, we use a word2vec model trained on the shuffled code-mixed corpus. The embeddings are aligned across

languages, allowing us to translate n-grams from one language to another. For instance, the nearest neighbor of a English 1-gram "I" could be the Hindi "mai".
Once the embeddings are created, we can create a code-mixed sentence for LG1 and LG2. To do this, we sort the n-grams in LG1 based on their cosine similarity with the most similar n-gram in LG2. We then replace each n-gram in LG1 with its top corresponding n-gram in LG2, based on the word embeddings, one at a time. We repeat this substitution process until we've used the specified number of substitutions.

4.3 Model

we choose a two text-to-text Transformer models from available models namely mT5 and mBART.Multilingual encoder-decoder models such as mT5 and mBART are suited to the MT task, and already cover both English and Hindi.we first explore the potential of these two models on the code-mixed translation task.
we finetune the mBART and mT5 models on the PHNC dataset which is the codemix Twitter dataset for baseline performance.

Then for curriculum training, we first trained both the MT5 model and mBART model on our Generated Synthetic data which was generated using OPUS corpus. And then again finetuned it with the real PHNC data.

5. Results

model	Traing	BLEU	rouge	rouge2	rougeL	rougeLsum
mt5	PHNC(Real data)	15.685	0.472	0.299	0.455	0.455
mt5	Synthetic data	4.894	0.292	0.136	0.274	0.273
mt5	curriculum	17.43	0.498	0.315	0.474	0.476
mbart	PHNC(Real data)	12.11	0.428	0.216	0.42	0.42
mbart	Synthetic data	4.106	0.28	0.11	0.262	0.262
mbart	curriculum	13.08	0.453	0.255	0.453	0.442

Figure 2: Results on mBart and mT5 (All tested on PHNC test dataset)

we used evaluate.load("sacrebleu") and rouge$_metric = evaluate.load("rouge")tocalculaterougear$

5.1 Qualitative Analysis

- Example of nearest neighbours:
  - aur, and: similarity=0.9570194482803345
  - of, ki: similarity=0.7434549927711487
- Eample of synthetic sentence:
- **en**: "Okay just calm down, we'll get to the bottom of this.",
  cm: "Okay just shAMta ho ham 'll get to the bottom of this.

- Sample
  - **Input's Text**
    @hurdangi haan.. @sagarikaghose sister will eat green mango today @the_hindu
  - **codemix (True Value)**
    @hurdangi haan.. @sagarikaghose Didi aaj hare rang ke aam khaengi @the_hindu
  - **mt5 on PHNC dataset**
    @hurdangi haan.@sagarikaghose bhai green mango peene ke saath kharab kar jaao the_hindu
  - **mt5 on synthetic data**
    agararikaghose bhU.Nge will eat green mAta Aja the_hindu
  - **mt5(curriculum training)**
    @hurdangi haan.@sagarikaghose behen aaj hare mango pee jaate hai the_hindu

  we can clearly see the improvement in curriculum training.

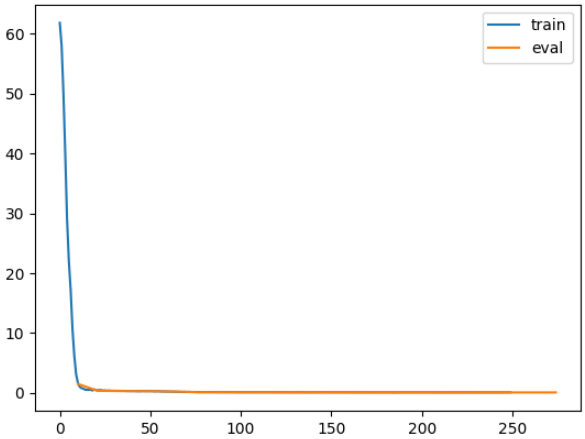

Figure 3: mt5 trained on PHNC

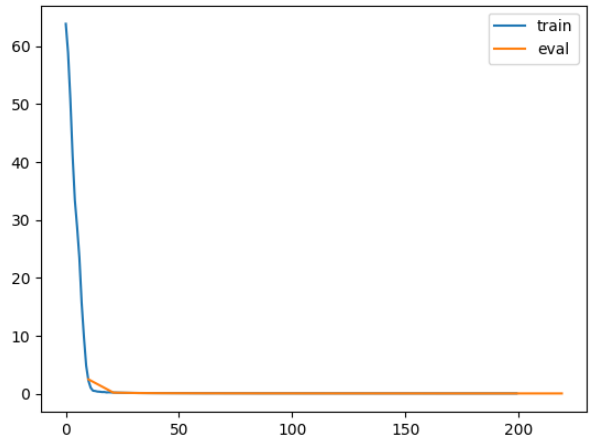

Figure 4: mt5 trained on Synthetic data

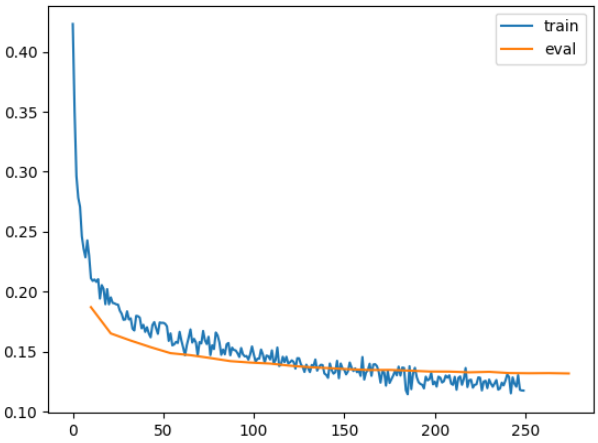

Figure 5: mt5 curriculum training

6. Discussion

6.1 Analysis

We trained MT5 and MBART using the curriculum training and on the synthetic dataset generated using OPUS corpus and on the PHNC dataset. As a control, we also train these models only on the PHNC dataset. We see that both the models trined using curriculum learning perform significantly better on all metrics. This is in agreement with the results of the paper. Among the 2 models (MT5, MBART), MT5 performs slightly better.
We also observe that the scores are marginally better than the scores mentioned in the paper. This may be because of the larger dataset being used.

6.2 Limitations and Challenges

In generating the dataset, we were limited by the time required to generate the synthetic dataset. With he expected time for creating the synthetic dataset being roughly 24 days. As such, we chose to use a subset of PHINC to generate the data, reducing the time to 4 days. With multi-threading and other parallelization techniques, we managed to reduce the time to 12 hours. This, combined with the significant training time limited the amount of data we used.
We observe, however that having a larger synthetic data produced better results. We would have liked to train on a larger dataset.

6.3 Conclusion

We generated a code-mix synthetic dataset and trained a model on it along with a real code-mixed dataset using curriculum learning. We compare it to the baseline model trained only on the real data and observe a significant improvement in performance. We have also used a larger synthetic dataset as compared to the paper, and notice a slightly better performance on most evaluation metrics.