Character-level Data Augmentation on Code Mixed Sentences for Low-Resource Settings

Niraj Pahari, Kazutaka Shimada
Kyushu Institute of Technology
nirajpahari@gmail.com
shimada@ai.kyutech.ac.jp

Abstract

Due to the rise in multilingual speakers and the use of social media, the occurrence of code-mixing is increasing. To understand these code-mixed data, different models require a large amount of data to train on. However, the code-mixed data is not readily available. Data augmentation is commonly being used for increasing the performance of the models by generating synthetic data to train on. Most of the existing techniques on data augmentation for code-mixing use large parallel corpus and/or external knowledge like POS taggers. In this paper, we study the combination of four different character-level operations to generate the augmented sentences from the source sentences while preserving the labels of the original sentences. Experiments show that this method can benefit the multilingual pre-trained language models on the low-resource cases. This method can be a strong baseline for future research on this domain.

1 Introduction

Whenever multilingual speakers communicate with each other, they tend to mix the structure or vocabulary from different languages. This phenomenon of mixing multiple languages during one conversation is known as code-mixing or code-switching. Code-mixing can occur within the sentence (intra-sentential) or between multiple sentences (inter-sentential). The rise in the number of multilingual speakers and the rise of social media usage has increased the occurrence of code-mixing. This has increased the need for NLP models to understand the code-mixing data. However, due to the spontaneous nature of code mixing, it is difficult to collect the data, so it is mostly considered a low-resource problem [1]. Code-mixed data involve more than one language. Identifying the token-level language tag is one of the fundamental tasks of this domain. In this paper, we perform a language identification task. It is a sequence labeling task.

Data augmentation is the set of techniques to generate synthetic data. It can help to train better and more robust models in the case that available original data is small. Augmentation techniques are commonly used in computer vision [2] and speech [3]. However, these techniques are more challenging in NLP due to the discrete nature of language [4]. Several researchers explored different techniques for data augmentation in NLP [5], [6], [7]. These studies mostly focus on classification tasks that are different from sequence labeling tasks. Some papers presented different augmentation techniques for sequence labeling tasks [8], [9], [10]. There are previous studies on code-mixed data augmentation [11], [1]. However, most of the work has been done on classification tasks, and the techniques mostly use external knowledge.

In this paper, we try to study the augmentation technique for code-mixed sentences using simple character-level operations. To create the augmented data, we use swapping, substituting, deleting, and inserting random characters from the original sentence. The labels are kept the same for augmented sentences and the original sentences. Since no external knowledge is required for these techniques, they are extremely easy to implement.

2 Related Work

Several studies have been done on data augmentation for NLP. Kobayashi [5] studied the technique to replace the synonym from the sentence to generate the augmented sentence. Easy data augmentation (EDA) proposed by Wei and Zou [6] uses synonym replacement, random insertion,

Table 1: Example of a Nepali-English sentence with augmented sentences. "$l1$", "$l2$", "ne", and "O" denote tags for each token. The detail is explained in Section 4.1. In this example, $l1$ stands for English, and $l2$ stands for Nepali. The modified tokens after character-level operations are shown with bold typeface.

Operation	Sentence
None / Original Sent.	Nepali$_{ne}$ language$_{l1}$ lai$_{l2}$ English$_{ne}$ ma$_{l2}$ kasari$_{l2}$ translate$_{l1}$ garne$_{l2}$?$_{O}$
Swap	**Nepail** language lai **Engilsh** ma **kasrai translaet** garne
Substitute	**Nepadi languaze** lai **Englidh** ma **kasmri** translate garne
Delete	Nepali **languge li nglish** ma kasari **translat** garne
Insert	Nepali **languaBge lzai** English ma kasari **tranDslate gcarne**
All	**Npali languaeg fai** English ma **kasair** translate garne

swap, and deletion of the words from the sentences to generate their augmented counterparts. Sennrich et al. [7] proposed the back-translation method, one of the widely used data augmentation techniques. It is the procedure of translating a sentence into one language and translating it back to the original language. While this technique is common for sentence classification tasks, it is not much applicable for sequence labeling since the position of labels might change during the process.

For the sequence tagging tasks, Sahin and Steedman [8] proposed the dependency tree morphing technique which was motivated by image cropping and rotation in computer vision. Ding et al. [9] used a generation approach for low-resource tagging tasks. Their approach linearizes the sentences with their labels, and a language model is learned based on the linearized sentences. The learned language model is then used to sample new sentences and is delinearized to obtain the augmented sentences with token-level labels. The study by Dai and Adel [10] modifies the sentence-level tasks into sequence labeling tasks and shows an improvement in the performance for different recurrent models and transformer-based models.

In the case of code-mixing, Pratapa et al. [12] used the Equivalence Constraint Theory to generate grammatically valid augmented code-mixed sentences. Gupta et al. [11] proposed an mBERT-based method to generate code-mixed sentences using the existing parallel data in two languages. Li and Murray [1] introduced a language-agnostic solution for creating synthetic data. They also use parallel corpus and POS taggers. In this paper, we study data augmentation with the combination of four different character-level operations. It is easy to implement and does not require any additional data or knowledge.

3 Method

Data augmentation is a technique to create artificial data from real data to increase the size of the dataset. The data for our task contain code-mixed sentences from social media. Code-mixed data involve more than one language. Moreover, in social media, the occurrence of typos is also high. Therefore, it adds to the complexity of data augmentation. Furthermore, sequence tagging tasks are more susceptible to data augmentation noise since they are token-level tasks as opposed to classification tasks, which are sentence-level tasks [9]. Inspired by the work by Wei and Zou [6], based on four simple word-level operations for simple data augmentation, we perform the simple character-level operations on the available dataset and generate synthetic data. For a given sentence, we randomly perform the following character-level operations:

1. **Swap** the adjacent characters in the word,
2. **Substitute** a character in the word with a random character,
3. **Delete** a random character in the word, and
4. **Insert** a random character at a random position in the word.

Table 1 shows an example of augmented sentences using the character-level operations mentioned above.

The number of tokens in the source sentence and augmented sentences are the same. We assume that the augmented sentences are label-preserving. In other words, the label of each token is the same as the original sentence token.

The architecture for the training is shown in Figure 1. Similar to Li and Murray [1], we employ the gradual fine-tuning process proposed by Xu et al. [13] for training pur-

Table 2: Dataset Statistics.

Dataset	Total Tokens	Train Sentences	Dev. Sentences	Test Sentences
Nepali-English	188,784	8,451	1,332	3,228
Hindi-English	146,722	4,823	744	1,854

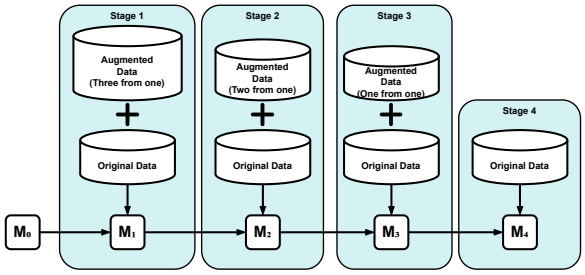


Figure 1: Architecture for gradual fine tuning with augmented data. "Three from one" denotes that three sentences are generated from one sentence in the original data, as the augmented data. M_0 is the initial pre-trained model. M_n is the fine-tuned model after stage n.

poses. The gradual fine-tuning is to apply the augmented data to the model gradually. In fine-tuning, a large size of augmented data is utilized first. Then, the size of additional data decreases step by step. M_0 is the pre-trained language model. In the first stage, we generate three augmented sentences from each sentence in the original dataset. The M_0 model is fine-tuned with these augmented data followed by the original dataset, resulting in the fine-tuned model M_1. In the next stage, we utilize two augmented sentences from each sentence in the original dataset. The M_1 model is again fine-tuned with the augmented sentences followed by the original dataset to obtain the fine-tuned model M_2. Similarly, fine-tuning is done with one augmented sentence per sentence in the original dataset. Finally, in the last stage, the model is fine-tuned with only the original dataset, which allows the model to fit better on the original dataset distribution. The final fine-tuned model M_4 is used for inferencing. Random operations from the four operations are applied to the generation of augmented sentences from each original sentence.

4 Experimental Settings

4.1 Tasks

The task in this paper is a language identification task. This is a sequence labeling task where each token should be classified as the particular language label (*lang1, lang2*), named entity label (*ne*), or others label (*O*). Correct language labels should be given to the tokens based on the language to which the token belongs. Since named entities are language-independent, we cannot identify the language label. Therefore, they are labeled as *ne*. If a token is emoticons or special characters, we cannot also identify the language. In a similar way to *ne*, the token is labeled with the "*O*" tag.

4.2 Datasets

In this paper, the Nepali-English [14] code mixed dataset and the Hindi-English [15] code mixed dataset are used. The Nepali-English dataset contains tweets and Facebook from public posts, whereas the Hindi-English dataset contains a corpus from Facebook pages of prominent public figures in India. Although Nepali and Hindi languages use Devanagari script for writing, these datasets only consider sentences written in Romanized form. The train, development, and test splits provided by [16] are used. The statistics of the dataset are shown in Table 2.

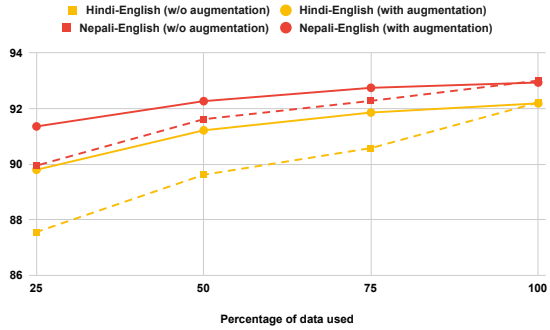
4.3 Experimental Setup

The experiments are conducted using two multilingual transformer-based language models: mBERT [17] and XLM-R[18]. The experimental models are implemented using Pytorch libraries [1)]. Huggingface models [2)] are used to load the pre-trained models. AdamW [19] optimizer with the learning rate of $5e^{-5}$ is used. TextAttack [20] library is used to generate the augmented sentences. For the gradual fine-tuning, 4 stages explained in Section 3 with 10 epochs per stage are run. Due to the imbalance in label distribution, the weighted F1 score is used as the evaluation metric.

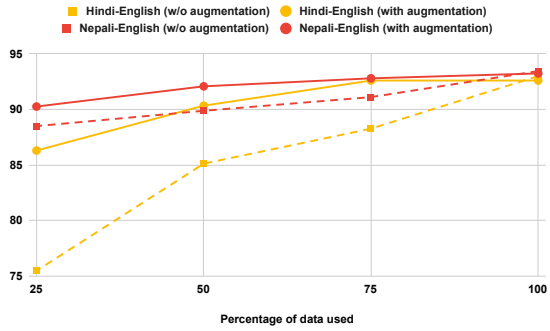
5 Results

We compare four types of original dataset sizes: 25%, 50%, 75%, and 100% used from the original data. We also compare the models with our data augmentation and with-

1) https://pytorch.org/
2) https://huggingface.co/models



(a) Language identification with mBERT



(b) Language identification with XLM-R

Figure 2: Performance of gradual fine-tuning with data augmentation for mBERT and XLM-R.

out the augmentation method. Figure 2 shows the performance of our data augmentation method compared with the case where data augmentation is not used [3]. The data augmentation method improved the results for both mBERT and XLM-R, especially in low-resource settings such as 25%. For both datasets, the tendency of the performance is similar. The difference in the performance was significant when the data use was very less. For the Nepali-English dataset, mBERT and XLM-R obtained performance gains of 1.41 and 1.26, respectively, in the 25% data. Similarly, for the Hindi-English dataset, mBERT and XLM-R obtained performance gains of 2.24 and 10.82, respectively, in the 25% data. It can also be seen that mBERT performs better than XLM-R in a very low-resource case. For both mBERT and XLM-R, the difference in the F1-scores "with" and "without" augmentation tends to be larger in Hindi-English than in Nepali-English. For example, in the 25% settings of XLM-R, while the F1-scores are 91.36 and 89.95 (the red circle and square) for Nepali-English, those are 86.3 and 75.48 (the yellow circle and square) for Hindi-English. It is caused by the number of instances of the datasets. For the 25% setting, while the number of instances for Hindi-English was 1206, that for Nepali-English was 2113. This result shows that our augmentation is effective in the case that the size of the dataset is small.

3) The vertical scale for two figures are different.

6 Conclusion

In this work, we study the character-level operations for generating augmented sentences for a code-mixing domain. We experimented with the augmentation techniques on different low-resource settings with mBERT and XLM-R pre-trained models. From the results of the experiments, we show that this easy method can improve the low-resource cases for code-mixing sentences. For the extremely low-resource case for XLM-R, it can boost the performance by 10.82 for the Hindi-English code-mixing dataset in the case that only 1206 sentences were used. This technique becomes a strong baseline for future studies on different augmentation techniques in this domain.

References

[1] Shuyue Stella Li and Kenton Murray. Language agnostic code-mixing data augmentation by predicting linguistic patterns. **arXiv preprint arXiv:2211.07628**, 2022.

[2] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. **Journal of Big Data**, Vol. 6, No. 1, pp. 1–48, 2019.

[3] Aggelina Chatziagapi, Georgios Paraskevopoulos, Dimitris Sgouropoulos, Georgios Pantazopoulos, Malvina Nikandrou, Theodoros Giannakopoulos, Athanasios Katsamanis, Alexandros Potamianos, and Shrikanth Narayanan. Data augmentation using gans for speech emotion recognition. In **Interspeech**, pp. 171–175, 2019.

[4] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. In **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**, pp. 968–988, 2021.

[5] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, pp. 452–457, 2018.

[6] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 6382–6388, 2019.

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In **Proceedings of the 54th Annual Meet-**

ing of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 86–96, 2016.

[8] Gözde Gül Şahin and Mark Steedman. Data augmentation via dependency tree morphing for low-resource languages. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 5004–5009, 2018.

[9] Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. Daga: Data augmentation with a generation approach for low-resource tagging tasks. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6045–6057, 2020.

[10] Xiang Dai and Heike Adel. An analysis of simple data augmentation for named entity recognition. In **Proceedings of the 28th International Conference on Computational Linguistics**, pp. 3861–3867, 2020.

[11] Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. Training data augmentation for code-mixed translation. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 5760–5766, 2021.

[12] Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1543–1553, 2018.

[13] Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. Gradual fine-tuning for low-resource domain adaptation. In **Proceedings of the Second Workshop on Domain Adaptation for NLP**, pp. 214–221, 2021.

[14] Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. Overview for the first shared task on language identification in code-switched data. In **Proceedings of the First Workshop on Computational Approaches to Code Switching**, pp. 62–72, Doha, Qatar, October 2014. Association for Computational Linguistics.

[15] Deepthi Mave, Suraj Maharjan, and Thamar Solorio. Language identification and analysis of code-switched social media text. In **Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching**, pp. 51–61, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[16] Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In **Proceedings of The 12th Language Resources and Evaluation Conference**, pp. 1803–1813, Marseille, France, May 2020. European Language Resources Association.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[18] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 8440–8451, 2020.

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **International Conference on Learning Representations**, 2018.

[20] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 119–126, 2020.