Improving Code-Switching and Named Entity Recognition in ASR with Speech Editing based Data Augmentation

Zheng Liang, Zheshu Song, Ziyang Ma, Chenpeng Du, Kai Yu, Xie Chen†
MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{liangzhenglz, songzheshu, zym.22, duchenpeng, kai.yu, chenxie95}@sjtu.edu.cn

Abstract
Recently, end-to-end (E2E) automatic speech recognition (ASR) models have made great strides and exhibit excellent performance in general speech recognition. However, there remain several challenging scenarios that E2E models are not competent in, such as code-switching and named entity recognition (NER). Data augmentation is a common and effective practice for these two scenarios. However, the current data augmentation methods mainly rely on audio splicing and text-to-speech (TTS) models, which might result in discontinuous, unrealistic, and less diversified speech. To mitigate these potential issues, we propose a novel data augmentation method by applying the text-based speech editing model. The augmented speech is coherent and diversified, also more akin to real speech. The experimental results on code-switching and NER tasks show that our proposed method can significantly outperform the audio splicing and neural TTS based data augmentation systems.

Index Terms: speech recognition, speech synthesis, data augmentation, code-switching, named entity recognition

1. Introduction
In recent years, end-to-end (E2E) automatic speech recognition (ASR) models have gained increasing research interest and present superior performance compared to conventional hybrid systems [1–4]. Nevertheless, there remain several challenging scenarios for E2E ASR models, which might limit their potential applications, such as code-switching and named entity recognition tasks.

Code-switching is a common language phenomenon where two or more languages appear in one utterance. The recognition of code-switching is important but challenging for E2E ASR models. There are always insufficient code-switching audios in the training data, while E2E ASR models are data-driven and their performance heavily depends on the diversity and coverage of the content of training data. Named entity recognition faces a similar dilemma for E2E ASR models. It is impractical to collect abundant training data to cover all possible names of persons, organizations, places, and other named entities. Consequently, the recognition performance of named entities and code-switching is far worse than that of common words in standard E2E ASR systems. Both tasks suffer from the problem of data scarcity, as it is impractical to collect sufficient training data to cover all possible cases of code-switching and named entities. Moreover, both tasks involve rare words and long tail [2–4] distributions where high-frequency words contribute to most of the appearance, while low-frequency words are rarely observed, which are hard to model by E2E ASR systems.

There are consistent and active research efforts on E2E models aiming to improve code-switching and named entity recognition performance given the importance of these two application scenarios. For the code-switching, there is a series of previous works in the E2E ASR model from the perspective of specific model design and explicit language information incorporation [5–11]. On the other hand, for named entity recognition, various aspects have been explored [12–17] in order to boost named entity recognition. One simple approach is to apply advanced TTS models to generate audio given the text of interest. One potential drawback of using the TTS model is the artifact and lacking diversity in acoustic and speaker characteristics from the synthesized audio. The other approach is to splice audio segments with small units, such as phone and word, from the existing corpus to form a long audio corresponding to the given text. However, this concatenative operation might introduce discontinuity and inconsistency. Therefore, there still exists clear discrepancies between the audios generated from current data augmentation approaches and real recordings.

In order to mitigate the discrepancy between augmented and real audio, in this paper, we propose a novel data augmentation method by applying the text-based speech editing technique. By “editing” the real audio, the augmented speech with the proposed method is able to retain the consistency and diversity of recording audio, and is expected to match well with the unseen audio in test time. The experimental results demonstrate that our proposed data augmentation yields significant performance improvements over competitive baseline systems, including neural TTS and audio splicing based data augmentation methods, on both code-switching and named entity recognition tasks.

2. Related work

2.1. Audio splicing and neural TTS for data augmentation
In this section, we briefly review two common types of speech data augmentation approaches, which are audio splicing and neural TTS.
Audio splicing is a straightforward and effective method to form new audio by simply concatenating speech segments of small units, such as phonemes and words, from the training corpus. One advantage of this method is that it preserves most of the characteristics of real speech. Audio splicing has been widely adopted for data augmentation to enhance speech recognition tasks, especially for code-switching and named entity recognition scenarios [1, 18, 19]. Moreover, for the target words that are scarce in the training data, the more fine-grained spliced segments or neural TTS can be applied to generate the speech segments of such words. However, one practical issue of audio splicing methods is the inconsistency of the segments from different sources in the augmented audio, which might sacrifice the potential performance gain.

With the recent advance of neural TTS models [25–27], high-fidelity audio can be easily generated. Therefore, neural TTS can be naturally applied for data augmentation to synthesize audios of given text [18–21]. Unlike audio splicing which relies on existing speech data, neural TTS models generate speech with high quality and naturalness by converting text input into speech audio. However, most neural TTS models aim to produce high-quality audio, instead of audio with complicated and diversified acoustic and speaker conditions. The artifact and lacking diversity in synthesized audio might limit the performance improvement for ASR tasks. Hence, in order to achieve better performance, most neural TTS based data augmentation methods need to mix synthesized data with similar amounts of real data to alleviate the impact of acoustic mismatch [20–22]. In this paper, we use the non-autoregressive TTS model, FastSpeech 2 [25], to generate high-quality augmented audios in a fast inference speed, and HiFiGAN [28] is applied as the vocoder for waveform generation based on the predicted Mel-spectrogram from FastSpeech 2.

2.2. Text-based based speech editing

Text-based speech editing is a technique that can "edit" audio by deleting, inserting or replacing the specified words in a flexible way [29–34]. The “edited” audio retains similar speaker, channel and prosody attributes as the original audio. Hence, the speech editing model has the potential to generate audio akin to the recording audio in the training corpus, while easing the aforementioned issues from audio splicing and neural TTS methods. In the next section, we will elaborate on the use of speech editing for data augmentation.

3. Data augmentation with speech editing

To overcome the above-mentioned drawbacks of existing data augmentation methods and generate speech with better authenticity and consistency, we propose a novel data augmentation by applying the text-based speech editing model on the original audios from the training corpus, to generate augmented audio with specified modification.

We extend the FastSpeech 2 with the BERT-style contextual modeling [35] that empowers the "extraneous" speech segment to capture the rich contextual information of the recorded audio, including speaker and prosody features, for text-based speech editing. The speech editing model architecture is illustrated in Figure 1. In the model, a text encoder is applied to capture the content information, a speech encoder is constructed to capture the acoustic attributes, and these two pieces of information are fused in the joint net to yield the Mel-spectrogram for further speech generation. By incorporating acoustic information derived from authentic speech, the speech editing model can generate speech that exhibits a high degree of contextual coherence and diversity in the modified region.

![Figure 1: Overall framework of the text-based speech editing model applied in this paper. The model is trained to generate the speech according to the edited text and the corresponding original audio.](https://liangzheng-zl.github.io/bedit-web)

3.1. Augmentation of code-switching recognition

The steps of generating code-switching speech with the proposed method, taking Mandarin-English for example, are as follows: First, forced alignment is performed on the existing monolingual Mandarin data to obtain the phone-level alignment information between the text and speech, as well as the boundaries of the words. Then, the English word or phrase is inserted or replaced into the Mandarin text following the grammatical rules of both languages to generate the code-switching text. Moreover, there are a series of previous works [36–39] aiming to generate code-switching text data. And then, the speech editing model predicts the duration of the modified region, i.e. the English words. The mask is then applied to the specified region of the Mel-spectrogram with some fixed value, e.g. 0.1. The well-trained model generates the Mel-spectrogram of the masked region based on the text information and contextual acoustic information captured from the edited text and masked Mel-spectrogram, and concatenates the generation with the original Mel-spectrograms to obtain edited Mel-spectrograms. Finally, the edited Mel-spectrogram is directly outputted or inputted into a vocoder to synthesize speech. We illustrate the pipeline of the model in Figure 2(a).

![Figure 2: The pipeline of data augmentation by applying the text-based speech editing model for code-switching and named entity scenarios.](https://liangzheng-zl.github.io/bedit-web)

3.2. Augmentation of named entity recognition

The generation of the speech with named entities is similar to the generation of code-switching speech. Specifically, given the named entities list and the texts with modified positions, the edited text can be easily obtained by insertion and replacement in the specified position. And masks are applied on the recorded audio, where the length of masking can be estimated from the model. Inference can be performed on the speech editing model given the complete text and masked speech, to obtain the edited Mel-spectrogram. The general process is illustrated in Figure 2(b).

---

1The code and samples of the speech editing model are available at https://liangzheng-zl.github.io/bedit-web
4. Experiments
We conduct experiments on two distinct tasks: code-switching and named entity recognition in ASR. For both tasks, all the audio is sampled to 16kHz and extracted 80-dimensional log-Mel filterbanks (Fbank) with 50ms frame size, 12.5 ms frame hop, and the Hann window function. The force alignment and grapheme-to-phoneme are both carried out by the GMM-HMM model implemented on Kaldi toolkit [40] to convert words to phones and acquire alignment information between the audio segments and phones, and the neural TTS and E2E ASR models are built on the ESPNet toolkit [41].

4.1. Code-switching in ASR

4.1.1. Datasets

The code-switching speech recognition experiments are conducted on the ASRU 2019 Mandarin-English Code-Switching data [42], which contains 200 hours of code-switching speech by 566 speakers and 500 hours of Mandarin speech by 1880 speakers. Additionally, 460 hours of mono-English data by 1172 speakers from the LibriSpeech dataset [43] are also included. The test set for the experiments comprises 20 hours of code-switching speech. To compare the augmented speech with the real speech, we randomly sample 100h data from the 200h code-switching data as Augreal set to serve as the simulation of the real collected code-switching data for data augmentation. The other half of 100h code-switching data combined with two kinds of monolingual data are used as Train set.

4.1.2. Models and tasks

We apply the multilingual ASR model LAE [44] without any language model for experiments. To compare the effectiveness of the proposed method with other methods, besides the Augreal set, five other augmented sets are prepared with the same texts in the Augreal set for data augmentation as follows. 1) Augspice set: The speech is generated with the audio splicing method. The English word speech segments in the sentences of the Augreal set are cut according to the alignment information. Then, FastSpeech 2 and HiFiGAN synthesize the English word speech segments with the inputs of the words. Finally, the synthesized speech segments of the words are inserted into the corresponding positions in the original speech to generate the speech data. 2) Augtxt set: The speech data is generated by the FastSpeech 2 and HiFiGAN with the text data from the Augreal set. 3) Augdit set: The speech data is generated by the text-based speech editing model and HiFiGAN. The regions of the English words in the original speech from the Augreal set are masked and then fed into the speech editing model with the corresponding texts to generate the speech. 4) Augdit−feats set: The process of generating this set is similar to that of Augdit. However, we keep only output Fbank feats generated by the speech editing model instead of converting them to waveform with HiFiGAN. 5) Augdit−feats+tts set: Half of the augmented data is from Augdit−feats, the other half is from Augdit.

The ASR baseline system is trained on the Train set. Different combinations of data augmentation sets (Augreal, Augspice, Augtxt, Augdit, Augdit−feats, Augdit−feats+tts) and Train are used to train ASR model separately. We evaluate these models using mixed error rate (MER), word error rate (WER) for English and character error rate (CER) for Mandarin. To keep the consistency of the speaker, FastSpeech 2, HiFiGAN and the speech editing model are all trained on the 100h code-switching speech in Train set.

4.1.3. Results

The results of the code-switching test set are summarized in Table 1. It can be observed that all the data augmentation methods can help to improve the performance of the ASR model on all the metrics, and the best scores on all metrics are achieved by training with Train+Augreal. In particular, training with Train+Augreal, Train+Augdit−feats and Train+Augdit−feats+tts yields similar results to training with Train+Augreal, and the CER and MER of training with Train+Augreal achieve the same scores with Train+Augreal. This indicates that our proposed method can generate speech with consistency and diversity analogous to real speech. Moreover, our proposed method can bring a higher boost in data augmentation results than the other two methods. The impact of the vocoder is shown by comparing the results of training with Train+Augdit and Train+Augdit−feats. It can be seen that the performance in terms of all metrics is consistently degraded by the artifacts introduced by the vocoder. Furthermore, by comparing the results of training with Train+Augdit−feats and Train+Augdit−feats+tts, it is revealed that a large proportion of unrealistic synthetic speech may adversely affect the model’s performance.

Table 1: MER, Man(CER) and Eng(WER) results on Code-Switching test set. The model is trained with Train set and augmented sets by different methods (real collection, audio splicing, neural TTS, and text-based speech editing).

<table>
<thead>
<tr>
<th>Training data</th>
<th>Man(CER)</th>
<th>Eng(WER)</th>
<th>MER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train+Augreal</td>
<td>9.6</td>
<td>33.2</td>
<td>11.8</td>
</tr>
<tr>
<td>+Augspice</td>
<td>8.6</td>
<td>30.0</td>
<td>10.5</td>
</tr>
<tr>
<td>+Augtxt</td>
<td>9.2</td>
<td>32.2</td>
<td>11.1</td>
</tr>
<tr>
<td>+Augdit</td>
<td>9.3</td>
<td>31.2</td>
<td>11.1</td>
</tr>
<tr>
<td>+Augdit−feats</td>
<td>8.8</td>
<td>30.7</td>
<td>10.7</td>
</tr>
<tr>
<td>+Augdit−feats+tts</td>
<td>8.6</td>
<td>30.2</td>
<td>10.5</td>
</tr>
<tr>
<td>+Augdit−feats+tts</td>
<td>8.7</td>
<td>30.4</td>
<td>10.6</td>
</tr>
</tbody>
</table>

4.2. Named entity recognition in ASR

4.2.1. Datasets

In the task of named entity recognition in ASR, we use personal names as the target named entities. The ASR model is trained on the GigaSpeech-M set, which consists of 1,000 hours of speech audio from different sources including audiobooks, podcasts and YouTube [45]. And two test sets are prepared as follows. The first test set, referred to as the Giga-name-test set, is generated by processing the standard GigaSpeech dev and test sets through an open-source NER toolkit. Specifically, we randomly sample 1,888 sentences that contain person names, resulting in a total of 1,434 unique names and a percentage of 8.3% of names among all words. This test set serves as an in-domain evaluation of the proposed method’s performance in recognizing person names. The second test set, referred to as the NER-name-test, is extracted from a publicly available dataset released in [46], which consists of 150 hours of speech audio with 38,891 unique named entity tokens of person, location, and organization. For our evaluation purposes, we only use the subset of the test set that contains personal names, which comprises a total of 1,650 unique names and a percentage of 5.0% of names among all words and consists of about 5 hours of audio. It serves as an out-of-domain evaluation to test the generalization ability of our proposed method.

The FastSpeech 2, HiFiGAN and speech editing model are all trained on the HiFiTTS dataset [47], which consists of 292

---

2 https://huggingface.co/dslim/bert-base-NER-uncased
hours of speech audio by 10 speakers.

4.2.2. Models and tasks

In this section, we follow the recipe for GigaSpeech in the ESN-Net to build the ASR model with 12 conformer blocks of the encoder and 6 transformer blocks of the decoder.

The target name list is formed based on all names appearing in the test sets. To generate the texts containing the target names, 800 sentences containing person names are randomly sampled from the Gigaspeech-XL set to build the database of the templates. The sentence templates with name tags and names from the name list can be flexibly combined to form complete sentences, which are then used as input for the three methods of speech generation. Similar to the experiments of code-switching, but only five augmented sets (Augsplice, Augtts, Augedit, Augedit−feats, Augedit−feats+tts) are prepared with the database of the templates and the name list. Finally, the generated speech is used to fine-tune the baseline system. Additionally, about 4 hours of synthesized speech data are used for in-domain testing and about 5 hours for out-domain testing. In order to balance the ASR performance between the general and target domains, the synthesized speech is mixed with an equal amount of real speech data, referred to as real-data set and containing no names, randomly sampled from the training corpus. The augmented audio contains better real acoustic features of the unmodified regions of the sentence and the augmented speech from the existing data augmentation, such as audio splicing and the neural TTS, text-based speech editing, and mixed data with real data (without names) sampled from the training set of ASR model.

The recognition of code-switching and named entity is challenging but important for E2E ASR models. Data augmentation provides a feasible and effective approach for these two scenarios. However, there exist several mismatches between the real audio and the augmented speech from the existing data augmentation methods. In this paper, we proposed a novel data augmentation method by applying the text-based speech editing technique, to directly modify the content of original speech from the training corpus. The augmented audio contains better diversity and is more realistic and coherent than that of traditional data augmentation methods, such as audio splicing and the neural TTS methods. Significant performance improvements can be achieved on both code-switching and named entity recognition tasks by applying the proposed data augmentation method.

5. Conclusions

The recognition of code-switching and named entity is challenging but important for E2E ASR models. Data augmentation provides a feasible and effective approach for these two scenarios. However, there exist several mismatches between the real audio and the augmented speech from the existing data augmentation methods. In this paper, we proposed a novel data augmentation method by applying the text-based speech editing technique, to directly modify the content of original speech from the training corpus. The augmented audio contains better diversity and is more realistic and coherent than that of traditional data augmentation, such as audio splicing and the neural TTS methods. Significant performance improvements can be achieved on both code-switching and named entity recognition tasks by applying the proposed data augmentation method.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62206171), and Alibaba Group through Alibaba Innovative Research Program.
7. References
