VCTUBE : A Library for Automatic Speech Data Annotation

Seong Choi[1*], *Seunghoon Jeong*[2*], *Jeewoo Yoon*[1], *Migyeong Yang*[1], *Minsam Ko*[2], *Eunil Park*[1], *Jinyoung Han*[1†], *Munyoung Lee*[3], *Seonghee Lee*[3]

[1]Sungkyunkwan University, [2]Hanyang University,
[3]Electronics and Telecommunications Research Institute

{choiseong, yoonjeewoo, mgyang}@g.skku.edu, {eunilpark, jinyounghan}@skku.edu,
{zldzmfoq12, minsam}@hanyang.ac.kr, {munyounglee, slee0003}@etri.re.kr

Abstract

We introduce an open-source Python library, VCTUBE, which can automatically generate <audio, text> pair of speech data from a given Youtube URL. We believe VCTUBE is useful for collecting, processing, and annotating speech data easily toward developing speech synthesis systems.

1. Introduction

Recent studies have shown that Text-to-Speech (TTS) systems based on deep neural networks (e.g., Tacotron, Deep Voice, etc.) can generate human-like speech with high quality [1, 2]. However, it has been reported that training such a deep learning model to generate human-like speech requires a large amount of speech data, e.g., at least 10 hours of <audio, text> pair data to generate high quality speech [3]. In practice, collecting and processing such a large amount of speech data is challenging.

To address the issue, we introduce VCTUBE[1], an open-source Python library, that can automatically generate <audio, text> pair speech data from a given Youtube video URL. Since Youtube provides a variety of audios with diverse languages, VCTUBE can help to develop a speech model for low-resource languages. Compared to existing libraries/tools [4, 5], VCTUBE can download, segment, and annotate speech data fully automatically without human intervention, resulting in fast development of speech synthesis systems.

2. VCTUBE

Figure 1 illustrates the overall architecture of VCTUBE. As shown in Figure 1, VCTUBE consists of three modules: (i) Audio Download, (ii) Caption Download, and (iii) Audio Split. In the Audio Download module, the audio file of the given video URL is downloaded in the wav file format using the youtube-dl[2]. Note that if there are more than one video in the playlist URL, multiple wav files can be downloaded. In the Caption Download module, to obtain the transcript data (e.g., start time, duration, and text) for each sentence of the given video, the Youtube Transcript API[3] is used. Then an alignment.json file, in <audio path, text> format, is generated. Note that the audio path for each text (sentence) is the path for the corresponding audio file that is generated in the Audio Split module. In the Audio Split module, the full length audio downloaded in the Audio Download module is split based on the start time and the duration of the transcript data obtained from the Caption Download module. The final output of the VCTUBE are a set of segmented audio files and their corresponding <audio path, text> pairs, which can be then fed into a TTS model.

Table 1: *Target sppech data used in our experiment. Note that EM, EF, KM, and KF denotes English Male, English Female, Korean Male, and Korean Female, respectively.*

Speaker	Video URLs	Total length
EM	https://www.youtube.com/watch?v=ACgFC1-9b4A https://www.youtube.com/watch?v=gFoYBO5ZFwo	6.0H
EF	https://www.youtube.com/watch?v=DSSm4QgvkCQ	4.04H
KM	https://www.youtube.com/watch?v=GLcbKiNDNKw https://www.youtube.com/watch?v=k5jL9SdqFAI https://www.youtube.com/watch?v=MVLQ3DC-VM4	5.40H
KF	https://www.youtube.com/watch?v=loBhedMYwrs https://www.youtube.com/watch?v=pHkhYKrJoeo	5.08H

3. Experiment

To demonstrate how speech data prepared by VCTUBE is useful in generating a good quality speech, we perform the following experiment.

3.1. Experiment Setup

3.1.1. Speech Data

We first select multiple Youtube videos based on the following two conditions: (i) by single speaker and (ii) with no background sound. Table 1 describes the target speech data used in our experiment, which consists of four different single speakers, i.e., Korean male/female and English male/female, and the total audio length of each speaker is around 5 hours. We then use the VCTUBE library to segment and annotate speech data for the given Youtube video URLs.

3.1.2. TTS Model Training and Evaluation

The basic structure of our TTS model mostly follows a widely-used TTS model, Tacotron2 [6], followed by Griffin-Lim vocoder module. We use the speaker embedding to train the model in a multi-speaker scheme, where multiple speakers are trained concurrently in a single model. Note that we train two different multi-speaker models 100 K steps for different languages, i.e., an English model with two speakers, EM and EF, and a Korean model with two speakers, KM and KF. To evaluate the model performance, we select 20 Korean and 20 English sentences (e.g., "We report both subjective and objective result.", "Have a nice day.") from books and dictionary example sentences.

* The first two authors contributed equally.

† Corresponding author.

[1]https://dsail-skku.github.io/VCTUBE.github.io/

[2]https://github.com/ytdl-org/youtube-dl

[3]https://github.com/jdepoix/youtube-transcript-api

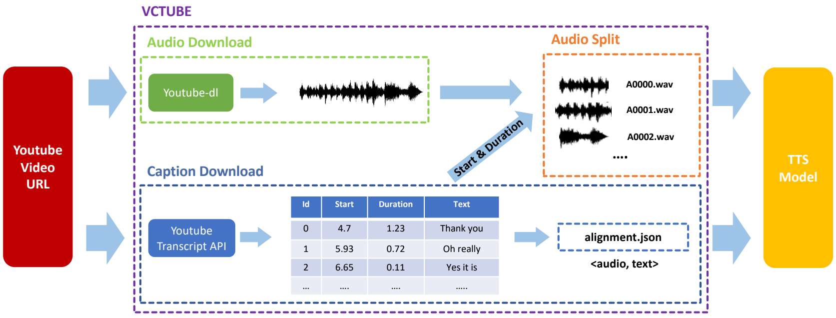

Figure 1: *Overall architecture of VCTUBE.*

3.1.3. *Performance Metric*

We calculate the Word Error Rate (WER) for each synthesized speech as follows:

$$WER = \frac{S + D + I}{N}$$

where S is the number of substituted words, D is the number of deleted words, I is the number of inserted words, and N is the number of words in the test phrase. Note that the perfect synthesized result has WER as 0. We use the Clova Speech Recognition API[4] for calculating WERs.

3.2. Result

Table 2 shows the WERs of the four different speakers. As shown in Table 2, the WERs of EM, EF, KM, and KF are 0.296, 0.291, 0.275, and 0.198, respectively. This indicates that *intelligible* speech is generated by training TTS models with data provided by VCTUBE. Figure 2 shows that the attention alignments for different speaker data are clear, meaning that the TTS models are well-trained with data provided by VCTUBE.

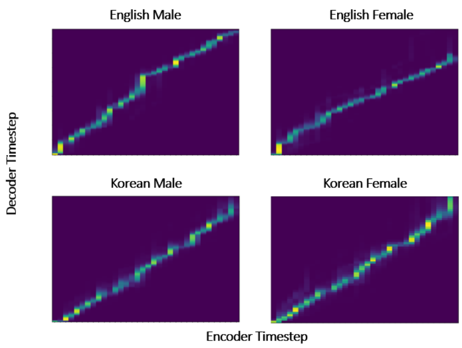

Figure 2: *Attention alignments plots for four different speakers. The x-axis in each graph indicates the encoder timesteps, and the y-axis represents the decoder timesteps.*

4. Conclusion

We introduced VCTUBE that can automatically generate <audio, text> pair speech data for a given Youtube video URL. By conducting experiments with a TTS model, we demonstrated that speech data provided by VCTUBE can be a good resource for generating *intelligible* speech. We believe VCTUBE is useful for collecting, processing, and annotating speech data easily toward developing speech synthesis systems.

[4]https://www.ncloud.com/product/aiService/csr

Table 2: *WERs (with 95% confidence interval), and the portions of substituted words (SUB), deleted words (DEL), and inserted words (INS) out of all the error words, respectively, for the experiments on different speech data.*

Model	Speaker	WER	SUB	DEL	INS
English	EM	0.296 ± 0.11	0.67	0.25	0.08
	EF	0.291 ± 0.10	0.68	0.28	0.04
Korean	KM	0.275 ± 0.06	0.76	0.24	0
	KF	0.198 ± 0.04	0.84	0.16	0

5. Acknowledgement

This work was supported in part by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [20ZT1100, Development of ICT Convergence Technology based on Urban Area] and the MSIT (Ministry of Science and ICT), Korea, under the ICAN (ICT Challenge and Advanced Network of HRD) program (2020-0-01816) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation).

6. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech*, 2017, pp. 4006–4010.

[2] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, "Deep voice: Real-time neural text-to-speech," in *International Conference on Machine Learning (ICML)*, 2017.

[3] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6940–6944.

[4] X. Li, Z. Zhou, S. Dalmia, A. W. Black, and F. Metze, "Santlr: Speech annotation toolkit for low resource languages," *Interspeech*, pp. 3681–3682, 2019.

[5] G. Levy, R. Sitman, I. Amir, E. Golshtein, R. Mochary, E. Reshef, O. A. Reichart12, and O. Allouche, "Gecko-a tool for effective annotation of human conversations," *Interspeech*, pp. 3677–3678, 2019.

[6] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.