OTAKO: A Comic Voice Synthesis System

Yisi Liu
Sujitech, Inc.
yisiliu@sujitech.com

January 22, 2018

Abstract

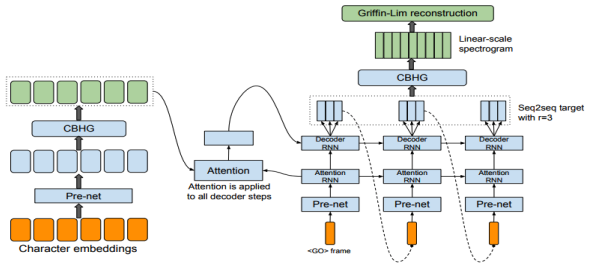
This paper presents a comic voice synthesis system, OTAKO, that can automatically synthesize speech directly from sentences in a similar voice of some comic characters. Based on the existing end-to-end speech synthesis model, Tacotron, given some text and audio clip pairs from one character in some anime series, we are capable of recovering his or her voice and produce speech using that voice. We introduce such framework to mimic one character's voice and hope to generate some interesting speeches based on it.

1 Introduction

It has always been an interesting task to make computer systems capable of reading and talking. In recent years, as the demand for autonomous robot is increased, such task to make synthesized voice more like human beings is drawing more attention. Modern Tetxt-To-Speech(TTS) pipelines are quite complicated[14, 15]. It is common for those systems to have various components, including preliminary text frontend linguistic features, a duration model, an acoustic feature prediction model and a complex signal-processing-based vocoder[1, 17]. These features needs extensive manual annotations and expertise supports. Therefore, such complexity adds much difficulties to designing a good general speech synthesis model framework.

With the development of deep learning and end-to-end models in recent years, an integrated end-to-end TTS system becomes available and feasible[15]. Such model could easily tackle the extensive feature engineering problem, since the end-to-end models only require raw data as input and thus we only need some minimal manual annotation. Furthermore, such framework could potentially minimize the bias and flaws in human annotations, which may contain a lot of subjective opinions and ideas. Another advantage of deep models is that we can capture some latent features and conditionings on different attributes of the text-audio pairs. With richer representations generated from such latent features, we could train a better model. Lastly, we could adapt such model to any data, unlike extensive newly engineered human annotations requirement previously.

Figure 1: Full Model Overview of Tacotron



In most of these TTS papers, researchers are using some existing text audio datasets, such as LJ Speech Dataset [8] or Nancy Dataset [16]. These two datasets still requires large human efforts. However, in the real world, we could easily find text audio resources, such as animes, TV series and movies. We can extract the audio data and corresponding subtitles and use such pairs to create a similar dataset. In this paper, we aim to introduce a method to create such dataset and use Tacotron as our baseline to train a model to mimic the voice actors' or actress' sound to create new audios based on some text that has never been seen in the originals.

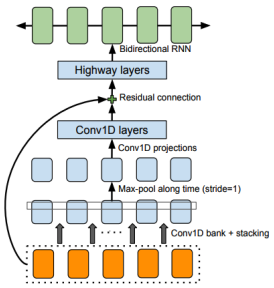
2 Tacotron Model

Tacotron consists of three major components: CBHG module, encoder and attention-based decoder. It is indeed a Seq2Seq model with attention mechanism[2], with an extra CBHG module. Figure 1 depicts the model at a high level and we will introduce each component in the following:

2.1 CBHG Module

CBHG is the abbreviation of Convolution Bank, Highway network and bidirectional GRU and we can see the module construction in Figure 2. It consists of a bank of 1-D convolutional filters[9], connected by highway networks[13] and at the top connected by a bidirectional

Figure 2: CBHG Module



gated recurrent unit (GRU) [5]. In their paper, it is reported that CBHG module is very powerful for extracting representations from sequences. The raw input sequence is first gone through the 1-D convolution filters with K different sizes. Then these outputs are stacked together and max pooled to pass into a few extra 1-D convolutions. They believe such processing could efficiently and explicitly model local and contextual information. This convolution outputs are fed into a multi-layer highway network to extract high-level features. Finally, at the top level, a GRU RNN is used to extract sequential features from the context in both directions. Such module is reported to be quite efficient to extract features.

2.2 Encoder

Like other encoder-decoder architectures, we have a similar setting in the encoder in Tacotron. The goal is to extract sequential representations of text. The input to the encoder is a character sequence, where each character is represented as a one-hot vector and embedded into a continuous vector. Then a set of non-linear transformations are performed to each embedding. Then a CBGH module transforms such output into the final encoder representation.

2.3 Decoder

In Tacotron, a content-based tanh attention decoder is used and applied on the encoder representation. We concatenate the context vector and the attention output to form the input to the decoder RNNs, which consist of a stack of GRUs. We then use a post processing network to convert such outputs from the seq2seq target to a target that can be synthesized into waveforms, which is the fundamental step of generating audios. We use another CBHG module as the post-processing net and use Griffin-Lim[6] algorithm to synthesize waveforms.

2.3.1 Experiment Setting

In our work, we preserve the original hyper-parameters and network architectures as the original Tacotron paper [15] because our contribution is not to adapt Tacotron framework. Specific settings for each unit can be referred to Figure 3.

Figure 3: Hyper-parameters and network architectures. conv-k-c-ReLU denotes 1-D convolution with width k and c output channels with ReLU activation. FC stands for fully-connected.

Spectral analysis	*pre-emphasis*: 0.97; *frame length*: 50 ms; *frame shift*: 12.5 ms; *window type*: Hann
Character embedding	256-D
Encoder CBHG	*Conv1D bank*: K=16, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-128-ReLU → conv-3-128-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Encoder pre-net	FC-256-ReLU → Dropout(0.5) → FC-128-ReLU → Dropout(0.5)
Decoder pre-net	FC-256-ReLU → Dropout(0.5)→ FC-128-ReLU → Dropout(0.5)
Decoder RNN	2-layer residual GRU (256 cells)
Attention RNN	1-layer GRU (256 cells)
Post-processing net CBHG	*Conv1D bank*: K=8, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-256-ReLU → conv-3-80-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Reduction factor (r)	2

3 Dataset Construction

The core step in this work is to construct a dataset that can be passed into Tacotron framework, from unstructured data instead of professional recording from one speaker.

We choose *Natsume's Book of Friends* [10] as our text-audio dataset. We record the audios from all five seasons of it and corresponding subtitles. In total, there are 20,000 sentences in five seasons, spoken by 3 protagonist characters and 7 antagonist characters. We used pydub [12] to segment each episode into audio clips according to the subtitle files. In subtitles, there is no information of the speaker of each sentence so this gives us a lot of difficulties to construct a single speaker text-audio dataset. Therefore, due to the complexity and effort of sentence character assignment, we decide to construct a multi-character text-audio dataset. Another reason is that there are only a few sentences for each character and such small sample size could not provide enough signals for the model to train.

In our future work, we will integrate CV techniques to help us do image classification tasks to identify the characters on screen at each time step at each spoken sentence. With such a smaller range of candidates, we can approximate the actual speaker of each sentence with a high confidence.

Another potential future work on this would be to fully take advantage of crowdsourcing[7, 3]. According to [11], it is found that a survey conducted on Amazon Mechanical Turk[4] has both higher completion rate yet higher accuracy. We could use Amazon Mechanical Turk or other online crowdsourcing platforms to help assign the actually speaker of each sentence by providing the screenshot of each time step of each sentence. This does

not need too much expertise knowledge and we just need to give the users some character information along with their portraits.

With the help of such highly efficient speaker assignment and potentially our previously proposed image recognition based model, we could not only construct a good dataset for training the speech synthesis model but also a good dataset for speaker recognition dataset.

Another big problem is the lack of data points. This is highly based on the length of anime series and number of characters. Therefore, such speech synthesis pipeline may not perform well on some short animes.

4 Experiments

We run the Tacotron model on our constructed dataset, which contains 20,000 sentences and 200 mins audio clips overall. We run the model on a single Nvidia GTX 1080 graphical card and it takes 30 hours to train. This is much longer than the reported 12-hour training time on LJ speech dataset which contains 13,100 sentence pairs, probably because Japanese words uses unicode which requires more memory to store than English words.

We do not have a golden testing criteria so we decide to analyze our synthesized work by human judges. The randomly chose 10 sentences from books to pass into the model to generate audios. The generated audios sound very weird since they come from different characters who have quite different voices. However, in the generated audios, we could tell the corresponding words accurately.

5 Conclusion

In our work, we prove that current end-to-end speech synthesis model Tacotron could 1, be used on Japanese based dataset and 2, be applied to dataset constructed from unstructured data. Our work is just a preliminary work on this direction and we hope that in the future we can construct a better dataset with extractions of single character's voice instead of our current multi-character dataset, using our proposed methods in section 3. We will also try our model to apply to some longer anime series to have enough data points for each individual characters. We hope that our work can be used to help generate and design some interesting side projects of some animes in a much more convenient and efficient way.

References

[1] Y. Agiomyrgiannakis. Vocaine the vocoder and applications in speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4230–4234. IEEE, 2015.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] D. C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75–90, 2008.

[4] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[6] D. Griffin and J. Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[7] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[8] K. Ito. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Y. Midorikawa. Natsume's book of friends. `http://www.natsume-anime.jp/`, 2003 (accessed Dec 10, 2017).

[11] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on amazon mechanical turk. 2010.

[12] J. Robert. Pydub. `https://github.com/jiaaro/pydub`, 2011 (accessed Dec 10, 2017).

[13] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[14] P. Taylor. *Text-to-speech synthesis*. Cambridge university press, 2009.

[15] 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 syn. *arXiv preprint arXiv:1703.10135*, 2017.

[16] R. Wilhelms-Tricarico, B. Mottershead, R. Nitisaroj, M. Baumgartner, J. Reichenbach, and G. Marple. The lessac technologies system for blizzard challenge 2011. In *Blizzard Challenge 2011 Workshop paper. DOI= http://festvox.org/blizzard/bc2011/LESSAC_Blizzard2011.pdf*, 2011.

[17] H. Zen, K. Tokuda, and A. W. Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009.