STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LX**, Number 1, 2021
DOI:

OBJECTIVE EVALUATION OF TEXT-TO-SPEECH USING AUTOMATIC SPEECH RECOGNITION

MĂRGINEAN LORENA SIMINA, UDREA HORAȚIU ALEXANDRU, LŐRINCZ BEÁTA

Abstract. The performance of text-to-speech can be most accurately measured with human verification. When this option is too costly and time-consuming, automatic verification methods can be utilized. In this work we propose to evaluate the output of TTS systems generated from different domains of input text (medical, culinary, cultural and lyrics data) and verify the output speech with the help of speech recognition. We observe that, even though the method's accuracy is highly dependent on the input domain and the speech recognition tool, it can be efficiently used for validating large amounts of synthesised speech automatically.

1. Introduction

Speech processing applications are widely used in our daily lives through voice assistants, navigation and telecommunication systems, or any other application that relies on the conversion of text to speech or its reverse, speech to text. These systems allow the end user to access information in either text or audio format, and facilitate the availability of data for the visually or auditory impaired as well.

Text-to-Speech (TTS) refers to the conversion of text into its corresponding audio waveform. It highly depends on the language and special characters contained. On the other hand, Automatic Speech Recognition (ASR) is the process of transcribing audio data into its corresponding textual representation and is also dependent on the language and the speaker's style of conversation.

Applications relying on speech processing strive to achieve high naturalness and accuracy. In order to validate these properties of TTS, manual validation

Received by the editors: ...

2010 *Mathematics Subject Classification.* 68T07, 68T50, 68T10, 62M45.

1998 *CR Categories and Descriptors.* I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Speech recognition and synthesis*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text analysis*.

Key words and phrases. text-to-speech, automatic speech recognition, validation, deep learning.

can be used. However, this is not feasible if large amounts of speech data need to be verified. As a first step towards enhancing the process of TTS validation an automatic verification can be performed by employing an ASR system. This can be used to signal eventual errors that occur in the output audio generated by the TTS system and improve it.

In this work we propose to evaluate TTS with the help of ASR. To achieve this, the result of speech synthesis is processed through speech recognition and compared with the initial text. Afterwards, the accuracy is calculated based on the resulting differences.

The paper is structured as follows: Section 2 presents recent developments on speech processing applications, Section 3 describes the proposed method for validation, Section 4 discusses the results and conclusions are drawn in Section 5.

2. Related Work

Text-to-Speech (TTS) and Automatic Speech Recognition (ASR) have long been interesting research topics in the field of artificial intelligence [1]. Recent successes of deep learning methods have pushed TTS and ASR into end-to-end modeling that achieves high quality synthesised speech and highly accurate textual transcriptions. [2], [3], [4] and [5] are among the first attempts to model ASR in an encoder-decoder based framework using neural networks. For TTS synthesis, a variety of neural models such as Deep Voice [6], Tacotron [7], Tacotron2 [8] and ClariNet [9] have improved the quality of synthesized speech and brought it even closer to human parity. However, both TTS and ASR require large amounts of high quality data in the form of speech-and-text pairs, usually reaching hundreds of hours for ASR and dozens of hours for TTS. High quality TTS data for a certain speaker was always hard to collect, and a large number of languages lack resources in terms of this kind of paired data, therefore posing a major practical problem for TTS and ASR.

In order to overcome the issue of lacking large amounts of training data, many works focus on the challenge of zero/low-resource TTS and ASR. Other studies propose multilingual solutions that benefit from transfer learning or combine TTS and ASR systems to mutually improve each other's performances.

[10], [11] and [11] address the low-resource ASR by formulating the problem in a multilingual scenario, where the data from other languages can be used as data augmentation. The methods described in [12], [13], [14] and [15] synthesize the speech of a target speaker with relatively few samples of paired data, but leverage large amounts of labeled speech and text data from other speakers. Both the two scenarios of low-resource ASR and TTS are a typical

learning transfer setting that just leverage few paired data and extra unlabeled data.

Unsupervised ASR methods are presented in [16], [17], [18] and [19] that follow a similar pipeline of speech segmentation, speech embedding learning, speech and text alignment where the data is processed on phonetic level. Neural TTS systems process speech on frame level, and use mel-spectrum [7] or MFCC [6] to generate the output speech waveform.

3. Proposed method

In this work we propose to validate the speech output of a TTS system with the help of ASR. The flow of the validation model is illustrated on Figure 1. The system takes the text as input, synthesizes it with the TTS engine and produces an audio stream. The generated stream is transcribed with the ASR engine and then the system compares the original text with the transcribed text, calculating WER and CER.

Word error rate (WER) is a common metric of the performance of a speech recognition or machine translation system. WER is derived from Levenshtein distance and is computed by dividing the number of errors with the total number of words. WER is a common metric of the performance of a speech recognition or machine translation system [20]. Another common method of evaluating the similarity of textual data is Character Error Rate (CER). Unlike WER, CER measures error at the character level instead of word level, therefore being a useful complementary metric that accounts for finer granularity.

We will use *gTTS* (Google text-to-speech API) as the TTS engine because it has a wide range of languages for converting a specific text into speech and also multiple accents for each of these languages.

Also, we will use Google's speech recognition engine as the ASR engine because it is easy to configure, widely used for speech to text conversions and it provides superior accuracy. The system must be adapted to receive the output of the TTS system and convert it to a common format (wav, mp3, aac). The *pydub* library can be used to achieve this by performing the required conversions.

WER is used for validating the original text with the generated transcription from the ASR service. For validating the initial text with the generated text WER and CER is computed using the *fastwer* python module.

For computing WER, the library uses *Levenshtein distance*. This is a string metric for measuring the difference between two sequences. The Levenshtein distance between two words is the minimum number of single-character edits

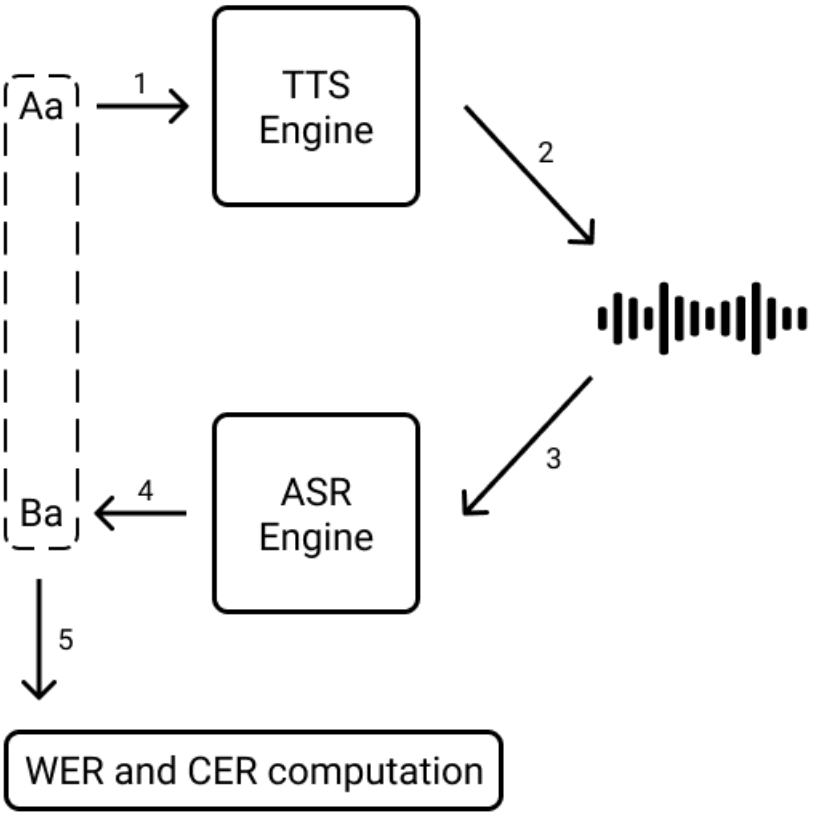

FIGURE 1. Overview of the proposed system's pipeline

(insertions, deletions or substitutions) required to change one word into the other [21].

For computing CER, the library uses its definition:

$$CER = \frac{i + s + d}{n} \tag{1}$$

where n is the total number of characters and i, s and d are the minimal number of character insertions, substitutions and deletions required to transform the reference text into the target text.

4. Experiments and Results

4.1. **Input data.** The system uses sentences as input data. We study the way in which different domains of the input affect the accuracy of the transcriptions by separating the data into different clusters according to their category and calculating the accuracies separately. The domains used in this experiment, along with some examples, are described below:

- **Medical data** - Sentences form articles that contain medical information, for example: *"The amino acid sequence of any polypeptide created by a cell is in turn determined by the stretches of DNA called genes"*
- **Culinary data** - Sentences form from articles that contain culinary information, for example: *"Risotto is one of the most challenging rice dishes for new chefs and there is a traditional method as well as a shortcut used in many restaurants"*
- **Cultural data** - Sentences from articles that contain cultural information, for example: *"Some artists countered the decline in patronage support by holding their own exhibitions and charging an entrance fee"*
- **Lyrics data** - Sentences from song lyrics , for example: *"Fall is a daydream from when I was young"*

The data was collected manually from various sites on the internet. The average number of sentences from a dataset category is around 100.

4.2. **Model architectures.** *gTTs* is a python library that uses the Google Cloud API in order to convert the text data into audio. There is also a CLI tool available for interfacing with Google Translate's text-to-speech API. The library uses verified data (from Google Translate's database) to train the model, constantly improving it.

Similarly, the *speech recognition* library uses the Google Cloud API in order to convert the speech data into text format. The audio must be at least 8 kHz in order to be transcribed into text. This library is also linked to Google Translate, constantly training it's model.

The speech recognition module also supports different languages (almost 125 different languages) and dialects. However, if the language is not supported, recognition falls back to the en-US locale. If a dialect is not supported (or doesn't exist) recognition falls back to main dialect or en-US in some cases. Main dialect can be specified just as first part of identifier. So 'en-US' and 'en' gives same results.

Because both libraries are using APIs the systems needs to have a stable internet connection in order to use them.

4.3. **Evaluation & Results.** The presented bar chart (see Figure 2) describes the accuracy results of the different datasets used in order to test the text-to-speech validation system. The blue bar represents the average word error rate (WER) and the orange bar represents the average character error rate (CER) for each domain. Most of the accuracy values are situated in the interval $[average - 3 * \sigma, average + 3 * \sigma]$, where $\sigma = standard\_deviation$, represented in this figure using dotted line, computed for both WER and CER.

Since the TTS engine is constantly improving, the results on further tests might be very different. The presented tests were done on 23rd April, 2021.

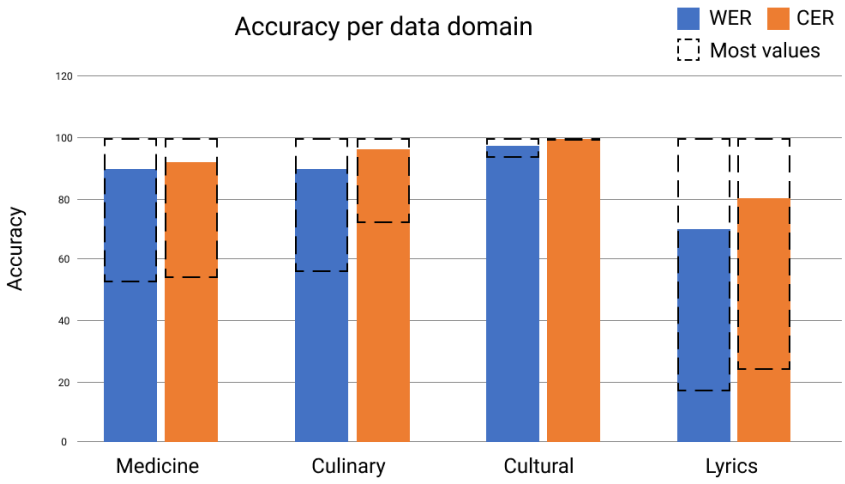

FIGURE 2. The accuracy results for the different data domains

4.4. **Discussions.** The datasets's accuracy seems to vary greatly with the origin domain. For example, the lyrics dataset gives the worst accuracy overall because many of the sentences do not have a specific meaning and their purpose is not to communicate facts but to express emotions and ideas. Because of this, the ASR engine that attempts to make sense out of the lyrics based on common speech patterns fails to transcribe the audio accurately.

On the other hand, the cultural dataset gives the best accuracy among the others because the words and sentences used are typical in daily speech. The ASR engine is most likely trained on conventional words and phrases rather than the sentences that occur rarely in common speech. Furthermore, the numbers are transcribed by the ASR system in numerical format, which

doesn't diminish the accuracy of the dataset since the original text also contains numbers in the same format.

The datasets don't contain special characters or uncommon symbols that although can be transformed into speech by the TTS system, cannot be transcribed back into the original character by the ASR but only to its sentential form.

As can be seen, the results mostly vary because of the ASR engine accuracy, which in turn depends on its training data domain. To the best of our knowledge, the TTS engine might create accurate speech for all the domains, but the measurement method employing the ASR system diverges them.

5. Conclusions

In this work we proposed to validate the synthesised speech output produced by TTS systems with the aid of ASR by comparing the transcription of the synthesised speech with the original text input. The method is highly dependent on the quality of the recognition engine and its training data domain. With our choice of engines and input data, we have concluded that the common speech domain produces more accurate results.

Human evaluation would clearly overcome the issue of depending on the accuracy of ASR engine, but since the proposed system is automatic and still offers an objective evaluation for TTS systems, it might be desirable in most cases where human interaction is costly or overall inconvenient.

Even when native speakers attempt to articulate text, the results may vary, so no general method of verbalization exists. Also, languages in general have many inconsistencies and their phonetic rules are not clearly defined. Because of this, it follows that the TTS engine's accuracy varies greatly with the evaluation method.

References

[1] Tokunbo Ogunfunmi, Ravi Prakash Ramachandran, Roberto Togneri, Yuanjun Zhao, and Xianjun Xia. A primer on deep learning architectures and applications in speech processing. *Circuits, Systems, and Signal Processing*, 38(8):3406–3432, 2019.

[2] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*, 2014.

[3] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *arXiv preprint arXiv:1506.07503*, 2015.

[4] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE, 2016.

[5] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE, 2018.

[6] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017.

[7] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis, 2017.

[8] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[9] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. *arXiv preprint arXiv:1807.07281*, 2018.

[10] Ekapol Chuangsuwanich. Multilingual techniques for low resource automatic speech recognition. Technical report, Massachusetts Institute of Technology Cambridge United States, 2016.

[11] Siddharth Dalmia, Ramon Sanabria, Florian Metze, and Alan W. Black. Sequence-based multi-lingual low resource speech recognition, 2018.

[12] Yutian Chen, Yannis Assael, Brendan Shillingford, David Budden, Scott Reed, Heiga Zen, Quan Wang, Luis C Cobo, Andrew Trask, Ben Laurie, et al. Sample efficient adaptive text-to-speech. *arXiv preprint arXiv:1809.10460*, 2018.

[13] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *arXiv preprint arXiv:1806.04558*, 2018.

[14] Sercan O Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. *arXiv preprint arXiv:1802.06006*, 2018.

[15] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR, 2018.

[16] Chih-Kuan Yeh, Jianshu Chen, Chengzhu Yu, and Dong Yu. Unsupervised speech recognition via segmental empirical output distribution matching. *arXiv preprint arXiv:1812.09323*, 2018.

[17] Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, and Hung yi Lee. Towards unsupervised automatic speech recognition trained by unaligned speech and text only, 2018.

[18] Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, and Lin shan Lee. Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings, 2018.

[19] Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, Hung yi Lee, and Lin shan Lee. Almost-unsupervised speech recognition with close-to-zero resource based on phonetic structures learned from very small unpaired speech and text data, 2018.
[20] Ahmed Ali and Steve Renals. Word error rate estimation for speech recognition: e-wer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 20–24, 2018.
[21] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

Babeş-Bolyai University, Department of Computer Science, 1 M. Kogălniceanu Street, 400084 Cluj-Napoca, Romania
Email address: {lorena.marginean, horatiu.udrea}@stud.ubbcluj.ro
Email address: {beata.lorincz}@ubbcluj.ro