INTERSPEECH 2021: Show & Tell Contribution
30 August – 3 September, 2021, Brno, Czechia

Save your Voice: Voice Banking and TTS for Anyone

Daniel Tihelka[1], *Markéta Řezáčková*[1], *Martin Grůber*[1], *Zdeněk Hanzlíček*[1], *Jakub Vít*[1], *Jindřich Matoušek*[12]

[1]Technologies for the Information Society (NTIS), University of West Bohemia, Pilsen, Czechia
[2]Department of Cybernetics, University of West Bohemia, Pilsen, Czechia
{dtihelka,juzova,gruber,zhanzlic,jvit}@ntis.zcu.cz, jmatouse@kky.zcu.cz

Abstract

The paper describes the process of automatic building of a personalized TTS system. The system was primarily developed for people facing the threat of voice loss; however, it can be used by anyone who wants to save his/her voice for any reason. Regarding the target group of users, the whole system is designed to be as simple to use as possible while still being fully autonomous.

Index Terms: speech synthesis, voice banking, voice loss

1. Introduction

The threat of voice loss affects us very deeply. There are people facing the diagnosis of laryngeal cancer, where their larynx may be partially or fully removed, or people with neurodegenerative illnesses (such as ALS) losing the ability to control their muscles, but there are also people with less serious diagnoses, such as thyroid surgery, with a risk of voice damage. For all of these, but also for anyone interested, we provide a voice banking system designed to build the personalized TTS voice package, which can subsequently be used on any PC or Android smartphone.

There are two primary design choices of the system. First, there are no requirements on the speakers who want to use the system, except some minimal acoustic conditions such as low background noise and the recommendation to use of at least a head-mounted microphone. Therefore, the voice banking is available to anyone, no matter his/her voice quality or the number of recordings carried out, and it is accessible from the no-stress comfort of their homes (similar to [1]). And, second, the whole process must run without the need of any human intervention.

In the present paper, we describe the details of our fully functional autonomous voice banking system evolved from [2], based on research started a decade ago [3].

2. Voice banking framework

The whole voice banking system is composed of several independent modules, which may even run on different machines on different locations. The communication among them is carried out either through RabbitMQ service bus for event-based processing actions, or through streaming HTTPS API when fast response is required.

2.1. Text corpus

The selection of the sentences to be recorded was one of the fundamental issues in the whole process. As the system was primarily designed for people facing laryngeal cancer, who usually have only a few weeks or even days before the surgery and for whom the long-time speaking is often painful at this stage of their illness, the sentences selected to the corpus are rather short. It is beneficial for any other user of the system, though, since we expect the majority of speakers being non-professionals, mostly even non-experienced ordinary persons, with no guarantee of consistent speaking style. With this in mind, the special multi-level selection algorithm was used to build the text corpus [4]. This allows us to maximize the coverage of various unit types, depending on how large part of the corpus will be recorded.

2.2. Recording and voice banking process orchestration

The recording procedure employs the frontend and backend modules, communicating through HTTPS API. The frontend module is responsible for the UI presentation and communication with the user, and it is the only visible part of the whole framework. It manages the registration of new users, allowing them to link their voice-banking profile to their Google or Facebook accounts. Then, it guides them through the recording process, where individual sentences are successively displayed, recorded, validated and stored. To make the process even simpler, there is a possibility to play a synthetic version of the sentence to be recorded. Especially for older users, it brings them more confidence in how the sentences should be read.

There are milestones defined within the recording process, triggering the automatic background build of the voice once crossed. The speaker may then listen to samples of his/her voice to get a notion of quality related to the milestone. Naturally, speakers may finish the recording anytime, which triggers the final build of the voice. But even then they are allowed to continue the recording further on.

The backend module of the system manages the personal data of the speakers as well as their recordings (both encrypted) and voice packages built later in the process. It interacts with the database and provides all the required information to the other modules in the whole system. Moreover, it controls the flow of events, starting from the speaker registration and going through the recording and voice build steps which are necessary for both the presentation of voice quality on individual milestones as well as the final build and distribution of the voice to the speaker. It also handles various error states which may occur during the voice banking process.

2.3. Validation

Every recording is instantly validated through a dedicated system, with multiple criteria being checked, such as the appropriate loudness and the length of the recording, voice activity detection (including a detection of noise, non-speech sounds, background music, etc.) and leading and trailing silence detection (prevents an undesirable cut of speech) [5]. The most complex is the ASR check[1] which validates if the speaker read the text correctly,

[1] Our in-house phoneme recognizer with specialized alignment to estimate a confidence measure for each reference word. The acoustic

rejecting the recording when significantly deviating from the text to be read. And additionally, the alternative transcript provided by the ASR is passed to the automatic segmentation process allowing the choice of the more probable variant during the force-alignment.

The result of the validation is either acceptance if everything is correct, or rejection with a message explaining the reason for the rejection and suggestions for improvement. There is also "partial acceptance" (e.g. highly confident recognition not matching the text), behaving as the full acceptance from the speaker's perspective, but the information is passed through the system to be used in a module which can handle it. The scheme of the validation can be found at [5].

One of the most important checks is the *dropout detection* [6]. Since the recording environment and technical device used by speakers is out of our control, there is a non-negligible chance of dropouts occurrence, i.e. when a part of audio signal is missing. As this may substantially distort both the automatic segmentation process and the synthetic output, the recording is rejected if a dropout is detected.

The tricky part of the validation is its responsivness, since the recording session must be very fluent for a speaker, without annoying him/her by latencies or lags between individual recordings. Therefore, streaming API is used here to communicate between recording application and the validator. Also, several concurrent validator instances may be registered in the backend which will then balance their load optimally.

2.4. Voice package build

The voice builder module waits on the service bus for requests to build the voice package, which may either be triggered by a milestone crossed or user-defined request to build the voice. The module itself has a form a of mutually connected "workers", through which the speech recordings, validation results and speaker-related meta-data (age, sex accent, etc.), are passed (see the scheme in [5]). All the data are provided by the backend through its HTTPS API.

There are multiple steps involved in the build of the voice package, starting at mastering where audio is filtered and normalized, through pitch-marks detection, speech segmentation, unit selection features build and training of DNN duration, F_0 and voice models. Two types of voice package are currently created: one suitable for the unit selection, the other for LSTM-based DNN synthesizer [7]. Once voice package is created, the module stores it in the backend and sends an appropriate message through the service bus to notify the framework about the finish of work. If the build has been triggered by the milestone cross, the preview TTS samples are generated and passed to the backend instead of the voice package. In case of error during the build phase, it also notifies the backend about the details of failure through the bus.

As the voice build process takes a couple of hours depending on the number of sentences recorded, any number of builder modules may run in parallel to speed up the concurrent build requests. However, the run of a single builder module works just fine as well.

Table 1: *Statistics of 60 recorded voices*

No. of sentences	avg. 1527, min. 200, max. 3500
Speech data [h:mm]	avg. 1:23, min. 0:11, max. 3:52

3. Conclusion

The whole system is now deployed in a pre-release mode.We already have several positive responses from people who lost their voice recently, and the use of TTS with their own voice created through the system is now among the important ways of simple (e.g. with family) and non-exhausting (e.g. when giving lectures) communication. The personalized TTS has been built for 60 speakers so far (see Table 1, listen to samples in [8]).

While the design of the system is fully language-independent (and localized to Czech and English), it is only available to Czech speakers for now. But we have also successfully tested the build of other languages (English, Russian and Germany) through the framework. Although most of the system parts (texts, voice builders, etc.) are ready, these languages are not allowed yet due to the lack of an acoustic model used by the validator. The extension for other languages is thus planned as next steps.

4. Acknowledgements

This research was supported by the Technology Agency of the Czech Republic, project No. TH02010307, and by the grant of the University of West Bohemia, project No. SGS-2019-027. The whole framework was developed in cooperation with Certicon, SpeechTech and Dept. of Otorhinolaryngology and Head and Neck Surgery, Faculty Hospital Motol.

5. References

[1] F. Malfrere, O. Deroo, E. Franques, J. Hourez, N. Mazars, V. Pagel, and G. Wilfart, "My-own-voice: A web service that allows you to create a text-to-speech voice from your own voice," in *Interspeech 2016*, 2016, pp. 1968–1969.

[2] M. Jůzová, D. Tihelka, J. Matoušek, and Z. Hanzlíček, "Voice conservation and TTS system for people facing total laryngectomy," in *Interspeech 2017*, 2017, pp. 3425–3426.

[3] Z. Hanzlíček, J. Romportl, and J. Matoušek, "Voice conservation: Towards creating a speech-aid system for total laryngectomees," in *Beyond Artificial Intelligence: Contemplations, Expectations, Applications*, ser. Topics in Intelligent Engineering and Informatics. Berlin, Heidelberg: Springer, 2012, vol. 4, pp. 203–212.

[4] M. Jůzová, D. Tihelka, and J. Matoušek, "Designing high-coverage multi-level text corpus for non-professional-voice conservation," in *SPECOM 2016*, ser. Lecture Notes in Computer Science. Cham: Springer, 2016, vol. 9811, pp. 207–215.

[5] J. Matoušek, "Save your voice: Voice banking and tts for anyone – more details," http://www.kky.zcu.cz/en/sw/voiceconservation.

[6] D. Tihelka, M. Jůzová, and J. Vít, "Grappling with web technologies: The problems of remote speech recording," in *SPECOM 2020*, ser. Lecture Notes in Computer Science, vol. 12335. Springer, 2020, pp. 592–602.

[7] D. Tihelka, Z. Hanzlíček, M. Jůzová, J. Vít, J. Matoušek, and M. Grůber, "Current state of text-to-speech system ARTIC: A decade of research on the field of speech technologies," in *Text, Speech, and Dialogue*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2018, vol. 11107, pp. 369–378.

[8] M. Řezáčková, "Save your voice: Voice banking and tts for anyone – samples," https://bit.ly/3yiXE4M.

model (TDNN-HMM) was trained on clear microphone speech, the 4-gram phoneme language model was trained on the set of sentences to be recorded. A specifically designed algorithm maps recognized phonemes to the reference transcriptions, allowing to catch inserted words apart from common deletions and substitutions. Each word recognised is assigned with a confidence score, used for the accept/reject decision.