INTERSPEECH 2019: Show & Tell Contribution
September 15–19, 2019, Graz, Austria

Off the cuff: Exploring extemporaneous speech delivery with TTS

Éva Székely, Gustav Eje Henter, Jonas Beskow, Joakim Gustafson

Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden
szekely@kth.se, ghe@kth.se, beskow@kth.se, jkgu@kth.se

Abstract

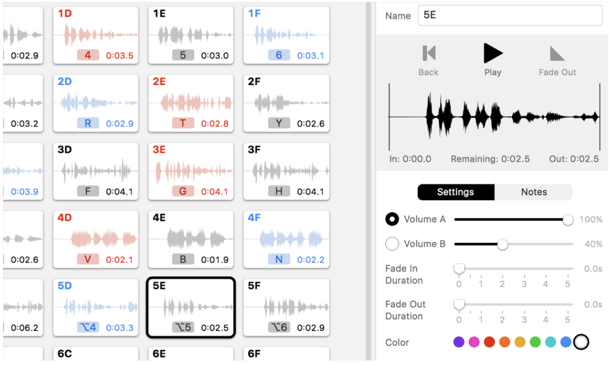
Extemporaneous speech is a delivery type in public speaking which uses a structured outline but is otherwise delivered conversationally, off the cuff. This demo uses a natural-sounding spontaneous conversational speech synthesiser to simulate this delivery style. We resynthesised the beginnings of two Interspeech keynote speeches with TTS that produces multiple different versions of each utterance that vary in fluency and filled-pause placement. The platform allows the user to mark the samples according to any perceptual aspect of interest, such as certainty, authenticity, confidence, etc. During the speech delivery, they can decide on the fly which realisation to play, addressing their audience in a connected, conversational fashion. Our aim is to use this platform to explore speech synthesis evaluation options from a production perspective and in situational contexts.

Index Terms: Spontaneous speech synthesis, public speaking, speech synthesis evaluation, filled pauses, AAC, soundboard

1. Introduction

Public speaking styles can be ordered in four categories according to delivery [1]: *impromptu* speaking involves spontaneous speech with no preparation at all; *manuscript* style is reading a speech word-for-word from its written form, while *memorised* delivery means committing the entire speech to memory. *Extemporaneous* public speaking, on the other hand, is done on the basis of a prepared structure, such as notes or an outline, but is otherwise delivered off the cuff. In this style, the material is presented freely, allowing the speaker to change their speech based on listeners' feedback. Communication coaches often advice against manuscript or memorised delivery styles unless the situation is very formal, because of the risk of sounding "robotic". It is no surprise that robots are used as a reference as conventionally, speech synthesisers (even expressive ones) deliver spoken material by reading text out loud, based on speech data from people doing the same. However, with the rise of natural-sounding spontaneous speech synthesis, TTS built entirely from unscripted spontaneous speech, simulating impromptu and extemporaneous styles becomes a possibility.

In this demo we present an exploratory platform for interacting with synthetic speech samples speaking part of a public speech. The samples are produced by a spontaneous speech synthesiser and differ in aspects such as fluency and filled-pause placement. Users can colour each sample to mark their subjective impression of how it sounds in the particular in context. This allows investigating context-dependent nuances in speech style such as certainty, authenticity, confidence, etc. Our hope is that this application will be a discussion starter in the scientific community about the need to move away from isolated utterances in TTS evaluation and the options of evaluating synthesis from the production perspective, where subjects interact with and use the TTS to attain their communicative goals.



Name 5E

In: 0:00.0 Remaining: 0:02.5 Out: 0:02.5

● Volume A 100%

○ Volume B 40%

Fade In Duration 0.0s

Fade Out Duration 0.0s

Figure 1: *Screenshot of the synthetic speech samples on an interactive grid interface of the soundboard app Farrago.*

2. Spontaneous speech synthesis

2.1. Spontaneous speech data and TTS

The data we use in this study is an untranscribed weekly technology podcast, called the "ThinkComputers" podcast, available in the public domain via the Internet Archive (archive.org). The recordings contain product reviews and discussions of technology news from two male speakers of American English mixed into a single audio channel. To segment the data into clean, well-defined utterances we used the speaker-dependent breath detection method proposed in [2]. With this method we selected 6,218 speech segments from 27 podcast episodes, each starting with a breath event from the target speaker. The utterances, henceforth referred to as the ThinkComputers Corpus (TCC), were automatically transcribed using the Google Cloud Speech API [3]. Filled pauses (FPs) were identified using the Gentle forced aligner [4]. For further details please refer to [5].

All versions of the voice described here were built using the implementation [6] of the Tacotron 2 spectrogram prediction framework [7]. All audio was sampled at 22.1 kHz. The Griffin-Lim algorithm [8] was used for waveform synthesis. Samples from this voice can be found under www.speech.kth.se/tts-demos.

2.2. Voice variants

We have produced several different speech variants – summarised in Table 1 – with differences in how disfluencies were addressed during annotation, training, and synthesis. We have previously [5] found our TCC voices to be rated significantly more appropriate than read speech synthesis on prompts from public speeches. However, the perceptual effect of the different speech variants is rather context-dependent.

Speech version 1 is synthesised by a voice we call **AutoFP**. This was built using the entire TCC corpus, but filled pauses 'uh' and 'um' were not annotated. This resulted in the synthes-

Table 1: *Summary of the six different synthetic speech versions used in our demo.*

Version	Name of voice	Corpus and training	Transcription of FPs	Prompt	Resulting speech
1	**AutoFP**	whole TCC	no	fluent	has automatically placed FPs
2	**CtrlFP**	whole TCC	yes, differentiating 'uh' and 'um'	FPs copied from ground truth	FPs exactly as in the prompt
3				fluent	no FPs
4	**GenFP**	whole TCC	yes, with a generic FP label for both 'uh' and 'um'	Ground-truth FP locations, unspecified type	has FPs in specified locations, type is decided automatically
5	**HalfFluent**	fluent 44.4% of TTC	N/A (no FPs in the training data)	fluent	no FPs
6	**TransFluent**	whole TCC, then transfer learning to the fluent 44.4%	no	fluent	very occasional automatically placed FPs

iser automatically inserting FPs in approximately 35% of synthesised utterances. With **AutoFP**, it is not possible to specify the location or type of FPs at synthesis time. Speech versions 2 and 3 were produced by **CtrlFP**, a voice trained on the entire TCC corpus but with FPs locations and type ('uh' or 'um') explicitly annotated. This voice can synthesise speech with FPs, if specified in the prompt (v2), or without FPs, when given fluent text (v3). Another FP placement strategy is employed by **GenFP** (in v4), which was built from TCC with FP locations annotated using only a single, generic label for both types of FPs. The generic FP label can be inserted into the prompt in order to synthesise speech with FPs in the user-specified locations, but where the *type* of FP ('uh' or 'um') is decided by the synthesiser. While 'uh' and 'um' are not always interchangeable, it can be desirable to leave the decision to the system, for instance to reduce the cognitive load if a person is producing the prompts.

To be able to synthesise utterances sounding more fluent than v3 we selected those 2,763 TCC breath groups (3 h 31 min, 44.4% of the corpus) that had no filled pauses and a maximum of one other disfluency (e.g., repetition, deletion, or prolongation, as located by the Gentle forced aligner [4]). This was used to build a voice we call **HalfFluent** (v5), created to synthesise speech that is as fluent-sounding as possible, at the cost of losing more than half of the training data. Finally, speech version 6 was synthesised by **TransFluent**, a voice variant trained using transfer learning: Starting from the **AutoFP** voice (which was trained on the entire TCC corpus for 150k iterations), **TransFluent**'s training continued for 70k iterations, using only the more fluent 44.4% of the corpus described above. The resulting synthesis sounds more fluent, while still benefiting from the entire set of training data. However, **TransFluent** still produces automatic FPs sporadically.

In [5], we have shown that the insertion of FPs does not take away from how engaging the speaker is perceived, but it does seem to increase listeners' impression of the authenticity of the speaker. Further perceptual evaluations of the above presented variations of the voice are subject to ongoing work.

3. Platform

Our demo platform currently uses the Farrago soundboard app by Rogue Amoeba, which has a tile grid interface that allows for quick audio playback via the mouse or keyboard, as well as an option to add colour-based marking and individually customised settings for each speech sample. The tiles in each row speak the same text prompt in the six different ways detailed in Sec. 2.2, ordered randomly. The synthesised prompts were taken from two Interspeech keynotes ("Dialogue as collaborative problem solving", by James F. Allen, 2017, and "Still talking to machines (cognitively speaking)" by Steve Young, 2010). By playing one sample from each row in succession, the TTS reproduces the first 20 utterances of the presentation.

A sample task, presented in the video attached to this demo paper, involves colour-tagging randomly ordered versions of each text prompt, such that a clicking through a progression of same-coloured utterances gives an impression of either a confident or uncertain speaker (coloured blue and red in Figure 1).

4. Use case scenarios for simulating extemporaneous speech production

Our proposed platform can be used to develop TTS evaluation strategies that take a production perspective, and also allows for perceptual judgments to take place in specific situational contexts. Another use case scenario is an application for Alternative and Augmentative Communication (AAC), specifically for people who use synthetic speech as their main verbal communication method because of a medical condition. The user can prepare a speech by synthesising different versions of each sentence in their presentation, then rehearse the speech in a similar manner as people who speak with their natural voice. The platform would allow samples to be annotated by the users according to any communicative nuances they find relevant. This would let AAC users modify their speech delivery on the fly, enabling them to be responsive to their audience in the moment.

5. Acknowledgements

This research was supported by the Swedish Research Council Project Incremental Text-To-Speech Conversion VR (2013-4935) and by the Swedish Foundation for Strategic Research project EACare (RIT15-0107).

6. References

[1] H. Gregory, *Public Speaking for College and Career*. McGraw-Hill Higher Education, 2010.

[2] É. Székely, G. E. Henter, and J. Gustafson, "Casting to corpus: Segmenting and selecting spontaneous dialogue for TTS with a CNN-LSTM speaker-dependent breath detector," in *Proc. ICASSP*, 2019.

[3] "Google Cloud Speech API video model," cloud.google.com/speech-to-text, accessed 2019-03-18.

[4] R. M. Ochshorn and M. Hawkin, "Gentle forced aligner," github.com/lowerquality/gentle, 2017, accessed 2019-02-14.

[5] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "Spontaneous conversational speech synthesis from found data," submitted to *Interspeech 2019*.

[6] R. Mama, "Tacotron-2 Tensorflow implementation," github.com/Rayhane-mamah/Tacotron-2, accessed 2019-02-14.

[7] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[8] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE T. Acoust. Speech*, vol. 32, no. 2, pp. 236–243, 1984.