How to train your fillers: uh and um in spontaneous speech synthesis

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

Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden

{szekely,ghe,beskow,jkgu}@kth.se

Abstract

Using spontaneous conversational speech for TTS raises questions on how disfluencies such as filled pauses (FPs) should be approached. Detailed annotation of FPs in training data enables precise control at synthesis time; coarse or nonexistent FP annotation, when combined with stochastic attention-based neural TTS, leads to synthesisers that insert these phenomena into fluent prompts on their own accord. In this study we investigate, objectively and subjectively, the effects of FP annotation and the impact of relinquishing control over FPs in a Tacotron TTS system. The training corpus comprised 9 hours of single-speaker breath groups extracted from a conversational podcast. Systems trained with no or location-only FP annotation were found to reproduce FP locations and types (uh/um) in a pattern broadly similar to that of the corpus. We also studied the effect of FPs on natural and synthetic speech rate and the interchangeability of FP types. Interestingly, subjective tests indicate that synthesiser-predicted FP types from location-only annotation often were preferred over specifying the ground-truth type. In contrast, a more precise annotation, allowing us to focus training on the most fluent parts of the corpus, improved rated naturalness when synthesising fluent speech.

Index Terms: Speech synthesis, spontaneous speech, filled pauses, disfluencies

1. Introduction

The majority of human speech is spontaneous and conversational. Being able to reproduce spontaneity in synthetic speech is therefore an important target for natural machine-mediated human communication. Spontaneous conversational speech contains a variety of phenomena which are not regularly considered in conventional speech synthesis of (and from) read speech. One such phenomenon is *filled pauses* (FPs), which are the focus of this paper. In American English FPs are generally uh and um, typically produced with a lengthened schwa-like vowel. Other important spontaneous speech phenomena include disfluencies like silent pauses, repairs, repetitions, lengthenings, and discourse markers (e.g., "like" and "you know"). When pursuing speech synthesis from spontaneous speech material, we face questions regarding how to approach these disfluencies both during annotation and training, and at synthesis time.

Research indicates that FPs are not mere speech aberrations, but play an important role in human dialogue [1]. For example, the presence of FPs improves recall of speech content in ways that cannot be accounted for merely by the fact that adding FPs slows speech down and thus allows for additional processing time [2]. Hesitations can in addition positively affect language comprehension [3, 4], also in vocoded speech [5], suggesting that highly natural TTS may benefit from this effect as well. Automatic generation of fillers has proven useful to assist human operators in Wizard-of-oz data collections [6]. In a separate effect, question responses with FPs tend to be interpreted as coming from increased uncertainty over responses without FPs [7]. Adjusting the degree of fluency and the insertion of FPs in synthetic speech can therefore moderate listeners' impression of the speaker's degree of certainty.

However, it is not evident how to harness the beneficial effects of spontaneous speech phenomena in TTS systems. Pursuing that goal entails questions about where to insert FPs in synthesis, which type to use (that is, uh or um), how the surrounding prosody should be altered, and how to otherwise adjust the fluency of the output speech more generally. An important question is if these problems can be addressed through machine learning directly on the training corpus with contemporary stochastic speech synthesisers such as sequence-to-sequence neural TTS. Alternative strategies include copying spontaneous phenomena from natural speech (although matching prompts are seldom available), manually inserting them into the prompts to be spoken (which is labour intensive and cognitively demanding), or using a "disfluency language model" trained on large corpora of spontaneous speech transcriptions. The latter approach may not reflect the speaker's characteristic manner of speaking, seeing that FP use has sociolinguistic connotations [8], is influenced by age, gender, and personality [9, 10, 11, 12], and exhibits shifting trends over time [13].

The purpose of this study is to investigate the effect of FPs uh and um in the context of neural TTS trained on a large single-speaker corpus of spontaneous conversational speech, as described in Sec. 3. We investigate different degrees of control over FPs in output speech, arising from differences in the level of detail in FP annotation at training time. We consider both the ability of stochastic TTS to replicate FP patterns from the training data as well as the perceptual implications of different levels of FP control in the output speech. Separately, we also study how disfluency annotation can be leveraged to train more fluent voices from spontaneous speech material. Through objective evaluations (Sec. 4) and two subjective listening tests (Secs. 5 and 6) we – among other results – find that i) neural TTS can learn to automatically reproduce patterns in FP location and type similar to those in the corpus, ii) that relinquishing control over FPs can provide perceptual benefits, and iii) that a significant majority of FPs rendered by our system are judged as plausibly realistic by human listeners for both uh and um.

2. Related work

The majority of prior work in synthesising filled pauses consider estimating *where* to place them and how they should sound as two separate problems. For unit-selection TTS, [14] applied local, speaker-dependent prosodic rules to generate synthetic fillers, restricted to the particular case where FPs are used to indicate an upcoming repair. Meanwhile, [15] used an n-gram model to predict FP placement. To generate FPs they supplemented their main corpus of read speech with a limited amount of spontaneous speech. This approach required the addition of silent pauses after FPs to avoid unnatural concatenations.

Name of voice	Corpus and training	Annotation of FPs	Condition	Prompt	Resulting speech
AutoFP	whole TCC	none	**AutoFP**	fluent	has automatically placed FPs
CtrlFP	whole TCC	yes, differentiating uh and um	**CtrlFP-GT**	FPs copied from GT	FPs exactly as in the prompt
			CtrlFP-SW	FPs opposite type as GT	FPs exactly as in the prompt
			CtrlFP-FL	fluent	no FPs
GenFP	whole TCC	yes, with a generic FP label for both uh and um	**GenFP**	GT FP locations, unspecified type	has FPs in specified locations, type is decided automatically
HalfFluent	fluent 44.4% of TCC	N/A (no FPs in the training data)	**HalfFluent**	fluent	no FPs
TransFluent	whole TCC, then transfer learning to the fluent 44.4%	fluent	**TransFluent**	no	very occasional automatically placed FPs

Table 1: *Summary of the voice configurations and the conditions used in the evaluations in this paper*

In statistical parametric speech synthesis, several approaches have been suggested for the automatic insertion of FPs into text prompts. [16] proposed a lattice-based approach weighting an RNN-based and an n-gram based speaker-independent language model to insert specific FPs into text. [17] extended this work, proposing data mixing approaches and the use of a unique phone label to represent FPs, an approach we follow in our present work. The evaluations in [18] concluded on these approaches that it is not enough to insert FPs in the right places, but they also have to sound right in the context of the utterance. [19] came to a similar conclusion, in that direct insertion of FPs into synthesised speech was detrimental to the quality, attributing this to a need to account for the natural variability in FP durations. More recently, [20] used conditional random fields and language models to insert disfluencies into text, taking into account function and desired frequency. Although ultimately aimed at TTS, their method was only evaluated on textual representations. With the exception of [15], the above approaches all aim for a speaker-independent language model of FPs. Methods for synthesising FPs are thus generally targeted towards overcoming a sparsity of spontaneous speech data required for creating spontaneous TTS.

3. Method

In this section, we describe the spontaneous conversational speech corpus we used, and how we built TTS systems with differences in FP content and annotation level. The experiments in Secs. 4 through 6 then evaluate these voices to shed light on the effects of these differences.

3.1. Corpus and transcription

The audio used in this study was sourced from the "ThinkComputers" podcast, available in the public domain from archive.org. The podcast features product reviews and discussions of technology news from two male speakers of American English. The audio is single-channel and provided without transcriptions. In the absence of clearly-delineated sentences in this audio-only material, we applied the speaker-dependent breath-detection method from [21] to segment the data into clean, well-defined utterances. This segmenter relies on a CNN-LSTM network trained on a small amount of coarsely-annotated seed data with mel-spectrogram and zero-crossing rate input features. It has been found to reliably identify breath events and speech segments for each of the two speakers, separating these from segments containing overlapping speech. As training data we selected 6,218 speech segments extracted by the breath-based segmenter [21] from 27 podcast episodes, each such utterance starting with a breath event from the target speaker, i.e., a *breath group* (BG). We refer to this as the ThinkComputers Corpus, TCC. To obtain text prompts for training TTS on TCC, we relied on automatic transcription, which has been shown [22] to be adequate for Tacotron TTS, particularly when subsequently phonetised using the `g2p_en` front-end [23]. However, the handling of conversational phenomena differs between ASR systems: Many systems deliberately detect and excise disfluencies from transcriptions, to return a more readable and fluent text. The Google Cloud Speech API [24], for example, was found to provide perhaps the most accurate word transcriptions, but omits FPs and repetitions from the transcription [25]. To identify FPs, we used IBM Watson Speech to Text with the US English BroadbandModel, which reflects disfluencies in its output. However, Watson marks these disfluencies with a single *hesitation* token, not differentiating between the FP types uh and um. FP types (as well as other hesitations like lengthenings and repetitions) were identified by subsequently running the Gentle forced aligner [26] on the audio segments and their transcriptions. Evidently, this pipeline can produce different levels of granularity for transcribing FPs: no transcription, location only (with a generic label), or location and type (uh/um).

3.2. TTS systems built

We built several different voices on the spontaneous conversational TCC data, all based on the implementation [27] of the Tacotron 2 spectrogram prediction framework [28] followed by Griffin-Lim phase recovery [29], spectrogram inversion, and inverse pre-emphasis for 22.1 kHz waveform synthesis. Like in [22], we first pre-trained a voice on the larger LJSpeech corpus [30] (approximately 24 h of read speech) for 65k iterations, and then trained our new systems on TCC for 150k iterations from this checkpoint. This procedure was found to substantially reduce the number of mispronunciations in the final TTS [22].

Each system we trained used a different approach to disfluencies, as explained below and summarised in Table 1. The first voice, which we call **AutoFP**, was trained on text where FPs were completely omitted from the transcriptions. Due to the nature of the statistical speech synthesis in Tacotron, which stochastically reproduces highly-likely patterns identified in the data, this led to a synthesiser that automatically produced uhs and ums in the output speech when fed fluent prompt texts. It is then not possible to specify the location or type of FPs at synthesis time, but they are automatically inserted and rendered by the system. Conversely, training on texts where FPs were explicitly annotated with different, unique symbols for uh and um produced a synthesiser that affords the user full control over the placement of uhs and ums as if they were regular words. We called this system **CtrlFP**, and used it in the evaluation to speak in three different ways: with FPs specified in the prompt – conditions **CtrlFP-GT** (ground truth) and **CtrlFP-SW** (swapped type, the opposite of the ground truth) – and without FPs, when given fluent text – condition **CtrlFP-FL**; see Table 1. The voice

GenFP is intermediate between AutoFP and CtrlFP, in that it was trained on same corpus with only FP locations (but not their type) transcribed, using a single, generic label to represent both uh and um. The generic FP label can be inserted into prompts to synthesise speech with FPs in user-specified locations, but the *type* of FP (uh/um) is decided by the synthesiser.

We also explored the utility of FP and hesitation annotation in building systems for generating fluent speech. How to synthesise fluent speech from spontaneous speech data is not straightforward; in our TCC data, for example, most utterances are disfluent to some degree. While the CtrlFP-FL condition is capable of generating speech without FPs, it may still have been influenced by the disfluent nature the training data. In fact, all of our voices so far produce occasional false starts and repetitions of function words, since these are common in the TCC material and the stochastic synthesis then reproduces these patterns from the data. To create more fluent speech, we trained two additional synthesisers, using a subset of the data comprising only those utterances with no annotated FP and at most one other disfluency (e.g., repetition, deletion, or lengthening, as located by the Gentle forced aligner [26]). This data amounted to 2,763 breath groups (3 h 31 min, or 44.4% of the full TCC). The first of these voices, which we call **HalfFluent**, was trained only on this fluent half of the corpus. The second voice, called **TransFluent**, used transfer learning to train a more fluent voice without fully excluding any of the corpus. Specifically, starting from the **AutoFP** voice (which was trained on all of TCC), the training of **TransFluent** continued for 70k iterations only on the more fluent subset of the corpus. Informal observations indicate that this voice still produces automatic FPs sporadically, but this was not formally evaluated.

4. Objective evaluation of automatic FP synthesis

4.1. Aim and setup

The purpose of our objective evaluations is to gain an overview of the automatic FP insertion of **AutoFP** and **GenFP**, and answer (in Secs. 4.2 though 4.5) four research questions about the behaviour of the model. Two episodes from the podcast were held out from TTS development to provide a benchmark for evaluation. Using the same selection criteria as used in the creation of the corpus, 611 BGs from the target speaker were extracted from these episodes, of which 51% contain at least one FP. These utterances were transcribed excluding their FPs and then synthesised four times. The output is analysed in the following paragraphs to evaluate if, when and where **AutoFP** adds FPs to the speech when none are explicitly transcribed.

Of the resulting 2,444 utterances, 76 were excluded from the analysis because attention/stopping failed, producing gibberish speech.[1] With the same method as in Sec. 3.1, the Gentle forced aligner was used on the synthesised samples to identify FPs that were generated. FPs identified at the beginning and end of the utterance were manually reviewed as Gentle sometimes confused these with a loud breath event and vice versa. Overall, FPs were generated by **AutoFP** in 34% of the utterances.

[1]This is a well-known occasional failure mode of the non-monotonic attention scheme used in Tacotron 2. However, we hypothesise that non-monotonic attention also might be important for **AutoFP** to learn to generate FPs, as these are acoustic events in training data that have no counterpart in the transcript to attend to; see also [31] on the impact of untranscribed words in deterministic sequence-to-sequence TTS.

FP at: B	M	E	Held-out	Synthesis	p-value
			49%	66%	<0.001
✓			23%	20%	0.109
	✓		17%	6%	<0.001
		✓	3%	5%	0.055
✓	✓		6%	1%	<0.001
✓		✓	1%	1%	0.844
	✓	✓	1%	1%	0.592
✓	✓	✓	0%	0%	0.200

Table 2: *Percentage of samples containing FPs at any given combination of the beginning (B), middle (M), and end (E) of the breath group (utterance), for the held-out episodes (611 BGs) as well as for **AutoFP** TTS of the same BGs (4 times each). p-values are for a two-sided Fisher's exact test for each row*

4.2. Frequency distribution of FPs in synthesis

Research question 1: *Do the generated filled pauses follow the frequency distribution in the corpus?*

Results show that utterances including a FP at the beginning or at the end are as frequent in the held-out sample as they are in the synthesis of the same utterances (Table 2). FPs are significantly less likely to be inserted in the middle of the breath group in synthesised utterances; either as the only FP or in combination with a FP at the start of the utterance.

In the synthesis, the overall distribution between uhs and ums is almost equal (51% vs. 49%) whereas in the held-out utterances um appears a little more frequently (42% vs. 58%). Based on Fisher's exact test, the difference in ratio of uhs to ums does not prove to be significant, neither on an overall basis ($p = 0.69$) nor when examined at the beginning, middle or at the end of the utterance ($p > 0.3$ for each).

4.3. Speech rate

Research question 2: *Does the insertion of FPs affect the speech rate (in syllables per second) of the rest of the utterance in the same way as in the corpus?*

To be able to analyse the impact of FPs on the speech rate of the remainder of the utterance without being biased by the direct effect of the length of the FP (which tends to be longer than syllables in speech articulation), the duration of the FPs is subtracted from the total duration of the utterance when calculating speech rate and utterance length. Resulting speech-rate distributions are graphed in Fig. 1. In the held-out samples, the average speech rate drops from 4.55 to 4.32 syllables/second (abbreviated syl/s) when FPs occur. This reduction in speech rate is reflected in the synthesis, where the speech rate is 4.50

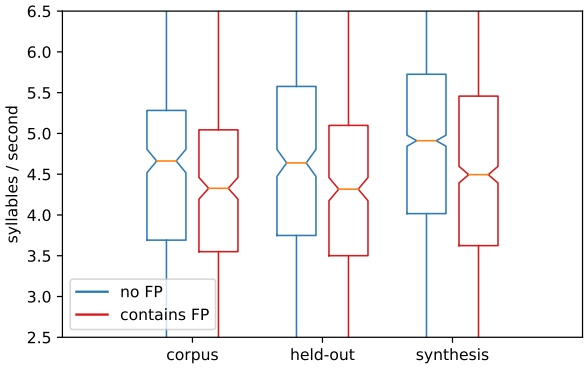


Figure 1: *Speech rates within the TCC corpus, the held-out and the synthesised samples, split on whether they contain FPs*

syl/s for utterances including a FP, compared to 4.89 syl/s for speech not containing FPs. Synthesised utterances have a higher speech rate in general, with an average speech rate of 4.76 syl/s in synthesis compared to 4.43 syl/s in the held-out sample.

4.4. FP model

Research question 3: *Is there a structure/model to automated FP insertion, beyond reproducing positional (utterance beginning/middle/end) FP frequencies in the corpus?*

Two observations in the objective evaluation point towards the existence of a FP model. First, if an utterance in the held-out episodes contained a FP, it is more likely to contain a FP in the synthesis as well (37% vs. 30% if the held-out utterance contained no FP). This indicates that the content of the utterance may be a factor in whether a FP is synthesised. Second, a FP is more likely to occur in the corpus when the speech rate is near the lower end of what can be achieved in a single breath group given the number of syllables (see Fig. 2).

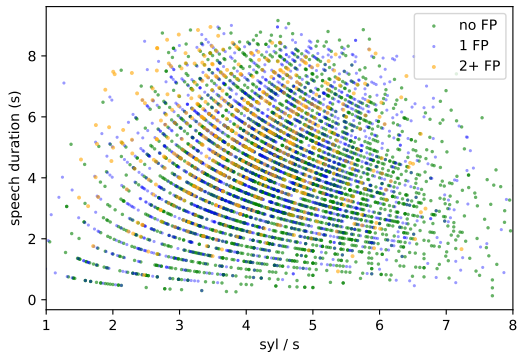


Figure 2: *BG length and speech rate vs. FPs in the corpus*

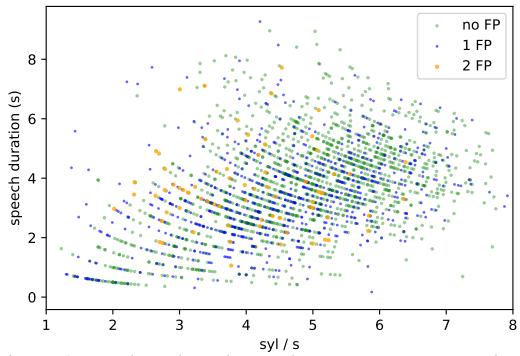


Figure 3: *BG length and speech rate vs. FPs in synthesis*

A similar, but weaker relationship between speech duration, speech rate and the occurrence of FPs can be observed in the synthesised samples (see Fig. 3). To test the hypothesis that a FP-model exists within the TTS system, we built a generalised linear model (GLM) of the binomial family to predict whether a FP will be inserted. The binary dependent variable is whether a FP is created in the synthesised utterance. Factors included in the model are those that appeared to have predictive value in the observations above. The existence of a FP in a particular utterance in the held-out episodes was included, which functions as proxy for a language model. Secondly, the speech duration (in seconds, excluding FPs) and speech rate (in syllables per second, excluding FPs) are included in the model, as well as the interaction effect between these two factors. In the predicted

Dep. Variable:	FP	Link Function:	logit
Model:	GLM	No. Obs.:	2368
Model Family:	Binomial	Df Model:	4

	coef	std err	z	P>|z|	4-fold bootstrap [P_min	P_max]
Intercept	-0.480	0.181	-2.66	0.01	0.01	0.03
C(FP in Orig., Treatment)	0.361	0.090	4.03	0.00	0.00	0.01
Syl/s	-0.014	0.033	-0.42	0.67	0.47	0.94
SpeechDuration	0.194	0.073	2.64	0.01	0.01	0.048
Syl/s:SpeechDuration	-0.059	0.013	-4.40	0.00	0.00	0.00

Table 3: *GLM design, coefficients and p-value ranges for different subsets of the repeated synthesis*

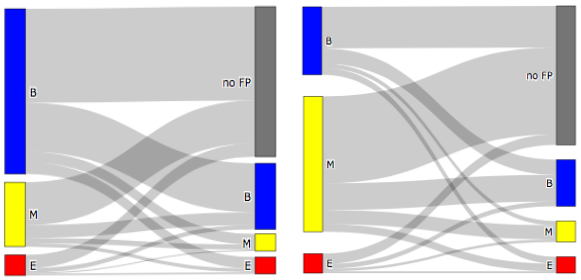
model both the presence of a FP in the original utterance as well as the duration of the BG (excluding FPs) and interaction effect between BG duration and speech rate proved to be significant factors. To test the robustness of the estimated model, we used a bootstrapping approach, removing one of the four synthesis runs from the dataset on each iteration. In Table 3 we report the minimum and maximum p-values for each factor under 4-fold bootstrap. The factors with a significant contribution to the original GLM remained significant in each of these runs.

From the results of the GLM modelling we conclude that even in the presence of inbuilt randomness in the synthesis, a model for FP inclusion can be found that takes into account the lexical content of the utterance, its duration in seconds and the duration compared to the speech rate.

4.5. Interchangeability of uhs and ums

Research question 4: *Do uhs and ums differ in function or are they interchangeable?*

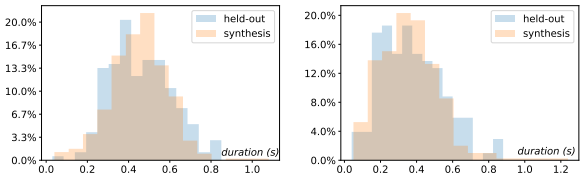
The 311 utterances from the held-out episodes that contained at least one FP were synthesised four times with **GenFP**. The overall distribution between uhs and ums remains nearly unchanged in the synthesis. Also, the likelihood of finding either one or the other FP appears to be determined much more by the overall distribution of these tokens at the particular position in the utterance, than by which FP that was found in the ground-truth realisation of the utterance by the speaker.



(a) *If held-out contains um* (b) *If held-out contains uh*

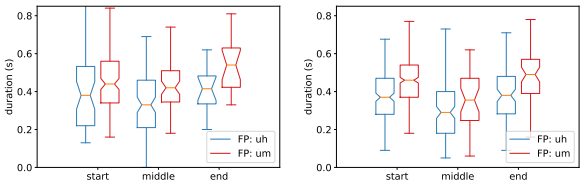
Figure 4: *Sankey diagrams of FP position in the held-out data (left in each diagram) and the synthesis (right)*

Observing the distribution of the length of the different realisations of uh and um, the ones synthesised by the **GenFP** system appear to be drawn from the same distribution as the held-out samples. Based on a K-S test between the samples, the hypotheses that the distributions are the same for held-out sample and synthesis do not get rejected for either uh ($p = 0.37$) or um ($p = 0.26$). As seen in Figs. 5 and 6, ums are generally longer than uhs, with BG-final ums being the longest.



(a) *Duration of um* (b) *Duration of uh*

Figure 5: *Held-out and the synthesised uh and um durations*



(a) *Length of FPs in held-out* (b) *Length of FPs in synthesis*

Figure 6: *Boxplots of held-out and synthesised FP durations*

5. Perceptual evaluation of disfluent speech

5.1. Evaluation design

For a perceptual evaluation of speech containing FPs, 20 utterances were selected from the AMI corpus [32], which is a corpus consisting of meetings where a group of four people discuss the design of a new remote control. We chose to use prompts from an unrelated corpus because FP placement tends to differ in individuals, and we wanted to be able to test how well the voices do on other speakers' FPs, and not only reproducing already plausible FP locations from the source speaker. The aim of the evaluation was to assess how much FP type mattered in different locations. Each selected utterance contained one FP, uh or um, in the beginning or in the middle of the utterance, yielding 4 different categories. BG-final FP position was not evaluated, because the lack of information of the speakers' breath events in the AMI corpus. The meaning of utterance-final FPs are often related to the subsequent utterance, and as such are difficult to judge when taken out of context.

Each utterance was synthesised in 3 conditions: **CtrlFP-GT** (GT location and type of FPs), **CtrlFP-SW** (GT location but *opposite* type of FP), **GenFP** (GT location but automatically selected FP type) A pairwise listening test was designed, yielding 60 comparisons across the 3 versions of each utterance.

It was brought to the attention of listeners that they will hear hesitations such as uh or um in the synthetic speech samples. Listeners were instructed to indicate if one of the two versions hesitated more realistically (like a human), or if they were both equally plausible. They were also given the option to choose that neither were plausible realisations.

5.2. Results

40 native speakers of English recruited through Prolific Academic took the test. At the beginning of the utterance in 43% of the cases 'both are plausible' was selected, versus 47% for utterance-internal FPs (Fig. 7). Only in 7% of the cases were neither of the two samples considered to produce natural hesitations. The **GenFP** voice was considered natural significantly more often in the middle of the utterance compared to the two other voices, based on Fisher exact test versus **CtrlFP-GT** and **CtrlFP-SW** ($p = 0.02$ and $p = 0.04$, respectively). At the start of the utterance, there is no significant difference in the evaluation of the three conditions in any of the pairwise combinations ($p > 0.30$). With the controlled voices, the voice that produced

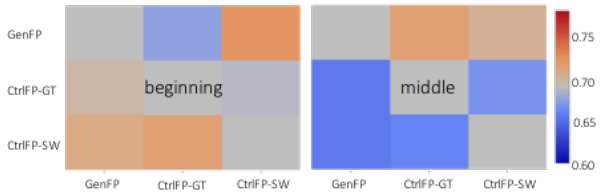


Figure 7: *Ratio where FP from voice y was rated as plausible when compared with voice x, incl. both being rated plausible*

an um was preferred 62% more often than the rendition of uh ($p < 0.001$); independent of the location in the utterance. This observed overall preference for ums was present regardless of gender or age of listeners.

6. Perceptual evaluation of fluent speech

6.1. Evaluation design

The aim of the second perceptual evaluation was to answer the question: *What is the best way to synthesise speech without FPs?* This is a valid concern given that FPs are present in over half of the BGs in the training data. In order to be able to include a reference sample from natural speech, 15 utterances were synthesised from the held-out episodes. A MUSHRA-like listening test was designed, including 4 synthetic versions and a natural utterance processed through Griffin-Lim [29]. The 4 different conditions of the synthetic speech were as follows: **AutoFP** synthesised until a version was produced without any FPs, **CtrlFP-FL**, **TransFluent**, **HalfFluent**. The task was to rate each sample on *naturalness*, which is a characteristic looking for listeners' global impression of the sample as opposed to focusing on local features as in Sec. 5. To make sure that the evaluated perceived differences were present persistently and due to the systems being different, as opposed to listeners' ratings being biased by small prosodic differences resulting from the stochastic nature of the synthesiser, each version was synthesised twice, and two identical evaluations were carried out with between-subjects design. Stimuli from both listening tests can be found under: www.speech.kth.se/tts-demos.

6.2. Results

Each experiment was completed by 20 listeners. Based on pairwise Wilcoxon signed-rank tests on naturalness ratings, **HalfFluent** and **TransFluent** were each rated significantly better than either the **AutoFP** and **CtrlFP** voices for fluent speech (max. p-value 0.01); see Fig. 8. Although there is no significant

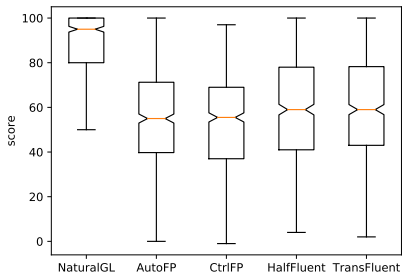


Figure 8: *MUSHRA ratings of fluent speech across both tests*

difference between the **HalfFluent** and the **Transfluent** voice in the overall experiment ($p = 0.76$) in one evaluation the **HalfFluent** voice is perceived best ($p = 0.02$) and in the other the **TransFluent** voice ($p < 0.01$), demonstrating the impact of the stochastic nature of the synthesis on these similar approaches.

7. Conclusions

We have shown that not annotating FPs in the training corpus results in the stochastic synthesiser reproducing FPs similarly to the frequency distribution in the corpus, likely with the help of an underlying FP model which takes into account the linguistic content of the message. From a theoretical perspective, we now have the ability to treat uh and um: a) like conventional English words, by spelling them out in the prompt; b) like an aspect of prosody, by leaving them out of the linguistic message entirely; or c) like something in between, by specifying their location but without asserting control over type and rendering. Perceptual tests reveal that the latter perspective most often delivers realistic sounding speech, but from a practical point of view, all three of these approaches yield strategies of addressing FPs that can be functional and desirable, depending on the application the TTS is deployed in. The problem of synthesising fluent speech from a disfluent corpus was also addressed, finding that minor improvements in fluent synthesis quality can be achieved by detailed annotation that allows for removing disfluencies during training.

8. 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).

9. References

[1] H. H. Clark and J. E. Fox Tree, "Using uh and um in spontaneous speaking," *Cognition*, vol. 84, no. 1, pp. 73–111, 2002.

[2] S. H. Fraundorf and D. G. Watson, "The disfluent discourse: Effects of filled pauses on recall," *J. Mem. Lang.*, vol. 65, no. 2, pp. 161–175, 2011.

[3] M. Corley and R. J. Hartsuiker, "Hesitation in speech can… um… help a listener understand," in *Proc. CogSci*, vol. 25, no. 25, 2003.

[4] M. Corley, L. J. MacGregor, and D. I. Donaldson, "It's the way that you, er, say it: Hesitations in speech affect language comprehension," *Cognition*, vol. 105, no. 3, pp. 658–668, 2007.

[5] R. Dall, M. Wester, and M. Corley, "The effect of filled pauses and speaking rate on speech comprehension in natural, vocoded and synthetic speech," in *Proc. Interspeech*, 2014.

[6] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," in *Proc. SIGDial*, 2010, pp. 1–8.

[7] S. Brennan and M. Williams, "The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers," *J. Mem. Lang.*, vol. 34, no. 3, pp. 383 – 398, 1995.

[8] J. Fruehwald, "Filled pause choice as a sociolinguistic variable," *Univ. Pa. Work. Pap. Ling.*, vol. 22, no. 2, p. 6, 2016.

[9] J. Gustafson and K. Sjölander, "Voice creation for conversational fairy-tale characters," in *Proc. SSW*, 2004, pp. 145–150.

[10] E. K. Acton, "On gender differences in the distribution of um and uh," *Univ. Pa. Work. Pap. Ling.*, vol. 17, no. 2, p. 2, 2011.

[11] C. M. Laserna, Y.-T. Seih, and J. W. Pennebaker, "Um… who like says you know: Filler word use as a function of age, gender, and personality," *J. Lang. Soc. Psychol.*, vol. 33, no. 3, pp. 328–338, 2014.

[12] M. Wester, M. Aylett, M. Tomalin, and R. Dall, "Artificial personality and disfluency," in *Proc. Interspeech*, 2015, pp. 3365–3369.

[13] M. Wieling, J. Grieve, G. Bouma, J. Fruehwald, J. Coleman, and M. Liberman, "Variation and change in the use of hesitation markers in Germanic languages," *Lang. Dyn. Chang.*, vol. 6, no. 2, pp. 199–234, 2016.

[14] J. Adell, D. Escudero, and A. Bonafonte, "Production of filled pauses in concatenative speech synthesis based on the underlying fluent sentence," *Speech Commun.*, vol. 54, no. 3, pp. 459–476, 2012.

[15] S. Andersson, K. Georgila, D. Traum, M. Aylett, and R. A. J. Clark, "Prediction and realisation of conversational characteristics by utilising spontaneous speech for unit selection," in *Proc. Speech Prosody*, 2010.

[16] M. Tomalin, M. Wester, R. Dall, W. Byrne, and S. King, "A lattice-based approach to automatic filled pause insertion," in *Proc. DiSS*, 2015.

[17] R. Dall, M. Tomalin, and M. Wester, "Synthesising filled pauses: Representation and datamixing," *Proc. SSW*, pp. 7–13, 2016.

[18] R. Dall, "Statistical parametric speech synthesis using conversational data and phenomena," Ph.D. dissertation, School of Informatics, The University of Edinburgh, Edinburgh, UK, 2017.

[19] S. Betz, P. Wagner, and D. Schlangen, "Micro-structure of disfluencies: Basics for conversational speech synthesis," in *Proc. Interspeech*, 2015, pp. 2222–2226.

[20] R. Qader, G. Lecorvé, D. Lolive, and P. Sébillot, "Disfluency insertion for spontaneous TTS: Formalization and proof of concept," in *Proc. SLSP*, 2018, pp. 32–44.

[21] 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, pp. 6925–6929.

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

[23] K. Park and J. Kim, "g2p_en: A simple Python module for English grapheme to phoneme conversion," https://github.com/Kyubyong/g2p, 2018, accessed: 2019-02-14.

[24] Google LLC, "Google Cloud Speech API video model," https://cloud.google.com/speech, accessed: 2019-03-18.

[25] T. Baumann, C. Kennington, J. Hough, and D. Schlangen, "Recognising conversational speech: What an incremental ASR should do for a dialogue system and how to get there," in *Dialogues with Social Robots*. Springer, 2017, pp. 421–432.

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

[27] R. Mama, "Tacotron-2 Tensorflow implementation," https://github.com/Rayhane-mamah/Tacotron-2, 2018, accessed: 2019-02-14.

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

[29] 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.

[30] K. Ito, "The LJ Speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[31] J. Fong, P. Oplustil Gallegos, Z. Hodari, and S. King, "Investigating the robustness of sequence-to-sequence text-to-speech models to imperfectly-transcribed training data," in *Proc. Interspeech*, 2019.

[32] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus," *Lang. Resour. Eval.*, vol. 41, no. 2, pp. 181–190, 2007.