Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 1960–1969
Marseille, 20-25 June 2022
© European Language Resources Association (ELRA), licensed under CC-BY-NC-4.0

Evaluating Sampling-based Filler Insertion with Spontaneous TTS

Siyang Wang, Joakim Gustafson, Éva Székely
Division of Speech, Music and Hearing, KTH Royal Institute of Technology,
Stockholm, Sweden
{siyangw, jkgu, szekely}@kth.se

Abstract

Inserting fillers (such as "um", "like") to clean speech text has a rich history of study. One major application is to make dialogue systems sound more spontaneous. The ambiguity of filler occurrence and inter-speaker difference make both modeling and evaluation difficult. In this paper, we study sampling-based filler insertion, a simple yet unexplored approach to inserting fillers. We propose an objective score called Filler Perplexity (FPP). We build three models trained on two single-speaker spontaneous corpora, and evaluate them with FPP and perceptual tests. We implement two innovations in perceptual tests, (1) evaluating filler insertion on dialogue systems output, (2) synthesizing speech with neural spontaneous TTS engines. FPP proves to be useful in analysis but does not correlate well with perceptual MOS. Perceptual results show little difference between compared filler insertion models including with ground-truth, which may be due to the ambiguity of what is good filler insertion and a strong neural spontaneous TTS that produces natural speech irrespective of input. Results also show preference for filler-inserted speech synthesized with spontaneous TTS. The same test using TTS based on read speech obtains the opposite results, which shows the importance of using spontaneous TTS in evaluating filler insertions. Audio samples: `www.speech.kth.se/tts-demos/LREC22`

Keywords: filler insertion, spontaneous text-to-speech, spoken dialogue system

1. Introduction

Spontaneous TTS has improved much recently due to usage of neural TTS engines. We can now generate conversational speech phenomena such as hesitations, disfluencies (Székely et al., 2019b; Székely et al., 2019a) and breathing (Székely et al., 2020) in such TTS alone. A major hurdle in applying such TTS to a spoken dialogue system is that dialogue systems are mainly trained on written texts, which do not contain fillers such as "um" and "like", that are common in the speech corpora spontaneous TTS systems are trained on (Székely et al., 2019a). This means that these systems can not take advantage of the potential of latest spontaneous TTS. Fillers are important for conversational systems, apart from making the speech sound more spontaneous, they can be used to communicate planning problems (Levelt, 1983), handle turn-taking (Maclay and Osgood, 1959) and communicate personality (Gustafson et al., 2021). A recent speech-to-speech chat-bot system was trained on 2000 hours of transcribed telephone conversations (Nguyen et al., 2022). The speech part of the corpus was enough to reproduce non-verbal vocalizations, but the amount of text was not enough to achieve semantic consistency in the dialogue. Another solution would be to insert fillers into responses from a chat-bot system trained on a large text-based dialogue corpus. However, filler insertion is a difficult problem due to the ambiguity of fillers in natural speech (Dall et al., 2014).

There are previous attempts at developing filler insertion methods, e.g. (Sundaram and Narayanan, 2003; Adell et al., 2012; Dall et al., 2014). These studies have focused exclusively on prediction-based filler insertion, that is to predict a single filler insertion pattern for an input sentence. The inherent ambiguity of fillers suggests that there are more than one "correct" insertion pattern for a given sentence. In this respect, the language modeling-based sampling approach is much faster and gives varied insertion patterns at each pass. But surprisingly, this simple approach to the best of our knowledge has not been studied in prior literature.

This work focuses on building language models to learn the contextual distribution of fillers and then use them for insertion with the sampling approach detailed in section 4. We propose an objective metric called filler perplexity (FPP) and evaluate its correlation with perceptual preference in sampling-based filler insertion. We also address two issues in the perceptual evaluation of filler insertions in general, (1) no evaluation on sentences generated by a dialogue system, a major intended application for filler insertion, (2) no evaluation using neural spontaneous TTS. The second shortcoming especially needs to be addressed as the state-of-the-art neural spontaneous TTS has achieved high naturalness regardless of presence of fillers in the input. Thus, it is not even clear whether or not filler insertion adds any value at all to the system if TTS can sound spontaneous without changing the input text itself. We use a Tacotron2-based state-of-the-art neural spontaneous TTS to synthesize speech audio (Székely et al., 2019b) and test two hypotheses in perceptual studies: (a) Which model is the best at filler insertion? (b) Does a good filler insertion model help spontaneous TTS sound more spontaneous and natural?

We use single-speaker corpora instead of a multi-speaker corpus as in many prior studies (Dall et al., 2014; Tomalin et al., 2015). This allows modeling individual speaking style both in filler insertion and TTS. Our results show that there is little difference between filler insertion models including comparison with ground-truth according to MOS from perceptual tests. 3-gram filler model obtains the highest MOS and improves the perceived colloquialness of chat-bot output when both filler-inserted and no-filler output are synthesized with spontaneous TTS. Running the same filler/no-filler evaluation with read speech TTS obtains opposite results: in this test the filler-inserted speech is clearly preferred with spontaneous TTS but not with read speech TTS. Subjective listening experiments reveal that read speech TTS is unable to voice fillers well. This shows the importance of evaluating filler insertions with spontaneous TTS, a shortcoming of most previous filler insertion studies.

2. Related Work

Dialogue generation has rapidly evolved from seq2seq (Sutskever et al., 2014) to pre-trained models like GPT-2 (Zhang et al., 2020). These are pre-trained on large unlabeled text corpora like Reddit (Adiwardana et al., 2020; Roller et al., 2021) or manually selected corpora like DailyDialog (Li et al., 2017). Recent studies have introduced knowledge that allows chat-bots to better answer factual questions (Dinan et al., 2018), or to display more consistent personalities (Zhang et al., 2018). Text style transfer has been introduced to control the linguistic style of textual output (Hu et al., 2020).

In human dialogue, filled pauses are used as turn-taking cues (Clark and Tree, 2002; Gravano and Hirschberg, 2011), to improve language comprehension (Fraundorf and Watson, 2011; Corley et al., 2007), to communicate the speaker's feeling of knowing (Brennan and Williams, 1995) and to provide cues to speaker traits (Laserna et al., 2014). This is usually lacking in dialogue generation as the models are typically trained on written or cleaned dialogue corpora that do not include spontaneous speech phenomena such as fillers.

There have been many studies on inserting fillers to clean TTS prompts, where a number of methods have been used: training a model on a limited domain corpus of transcribed human utterances (Sundaram and Narayanan, 2003); using rule-based systems (Cohn et al., 2019); training an n-gram filler insertion model on read speech corpora supplemented with spontaneous speech (Andersson et al., 2010); using a lattice-based approach weighting an RNN-based and an n-gram based speaker independent language model (Tomalin et al., 2015); using data mixing approaches combined with unique phone labels for filled pauses (Dall et al., 2016); and using conditional random fields and language models to insert disfluencies into text aimed at TTS (Qader et al., 2018). One prior study (Székely et al., 2019c) showed that a TTS trained on a spontaneous corpus can convincingly insert fillers by itself while synthesizing speech audio without explicit text-based training.

The previous studies have largely framed filler insertion as a single prediction problem(Tomalin et al., 2015; Dall et al., 2014), that is to predict a single insertion pattern for an input sentence. Thus, calculating F1 with precision and recall on the test set is a natural choice for objective evaluation. Single prediction has not been questioned despite that filler occurrence is ambiguous in natural speech (Dall et al., 2014).

The perceptual evaluation in previous studies is done with either no TTS (Qader et al., 2018; Tomalin et al., 2015), i.e. participants only read filler-inserted sentences and rate them, or with speech synthesized by earlier TTS methods such as concatenative (Adell et al., 2012) or HMM-based TTS (Dall et al., 2014). Only one study (Adell et al., 2012) used a spontaneous TTS in evaluation, but the underlying TTS engine is concatenative with some rule-based aspects to voice fillers.

The choice of multi-speaker corpus is another characteristic of prior studies. Data from many different speakers is pooled together to form one large corpus which is used to train a single model (Dall et al., 2014; Tomalin et al., 2015). This approach gives more data but assumes uniform filler pattern across speakers, an assumption challenged by our inter-speaker analysis. Instead, we use two English spontaneous speech corpora, each containing only one speaker which ensures more consistent filler insertion patterns for the models to learn from and which also allows us to investigate differences in filler usage across speakers. We also train spontaneous TTS on each of the two corpora. This, along with filler models trained on the same corpus, results in a complete pipeline of filler insertion–TTS that is capable to transferring a speaker's own speaking style to unseen text input.

3. Corpus and TTS

Two different spontaneous speech corpora were used in the experiments. The first is from the audio recordings of the Trinity Speech-Gesture Dataset (TSGD) (Ferstl and McDonnell, 2018), comprising of 25 impromptu monologues by a male actor, on average 10.6 minutes long. The actor is speaking in a spontaneous, colloquial style. The second dataset is a 9-hour single-speaker corpus (Székely et al., 2019b), from a public domain conversational podcast, called the ThinkComputers Corpus (TCC). The speaker uses an extemporaneous conversational style, speaking freely using a prepared outline.

Both corpora are transcribed using ASR, and subsequently manually corrected. In order to maximize the utterance length in the corpora and to enable insertion of inhalation breaths in the TTS, we used a data augmentation method called breath group bigrams, which essentially consists of segmenting a speech corpus into stretches of speech delineated by breath events, and then combining these breath groups in an overlapping fashion to form utterances no longer than 11 seconds

filler word	TSGD	TCC
um	0.0108	0.0246
uh	0.0121	0.0208
like	0.0925	0.0028
you know	0.0048	0.0084

Table 1: Frequency of filler words in the two corpora.

(Székely et al., 2020). This method also makes it possible to learn contextual information beyond respiratory cycles during TTS training. Voices were trained with the neural TTS engine Tacotron 2 (Shen et al., 2018). We used a PyTorch implementation[1], training each voice for 200k iterations on top of the pre-trained model released by NVidia. For vocoding, we fine-tuned the pre-trained universal model of HiFi-GAN (Kong et al., 2020) on the respective corpora.

Extracting filler words from the text transcripts can be done with direct matching for "um" and "uh", while "like" and "you know" have non-filler instances thus need extra attention. We use POS tags extracted with Stanza (Qi et al., 2020) to differentiate filler vs non-filler occurrences of these two words as we find that when "like" and "know" are tagged as verbs, they are less likely to be fillers. The frequency of fillers are shown in Table 1. The speaker of the TSGD dataset is a heavy user of the filler "like" which presents a challenge to the perceptual test as we later find that people tend to rate more "like" insertion as bad, irrespective of context or synthesized speech quality. Each corpus is divided into utterances by combining the previously mentioned breath groups with a max length threshold of 40 characters. We then divide sentences into train/dev/test sets on a 0.90/0.05/0.05 ratio for each dataset.

4. Sampling-based Filler Insertion

4.1. Problem Setup

We consider filler insertion as an auto-regressive language modelling task. Language modelling aims at learning a probability distribution of the current token x_t over the vocabulary set X given prior tokens in the sequence $x_0, ..., x_{t-1}$ (where $x_i \in X \; \forall \; i$), modelling:

$$P(x_t|x_0, ..., x_{t-1}).$$

The learned auto-regressive distribution allows a language model to generate new sequences by conditioning on some given context sequence and sampling from the output probability distribution. This process is modified to achieve filler insertion by sampling for only filler tokens as illustrated in Figure 1.

We denote the set of fillers as $FL = \{fl_0, ..., fl_n\}$ where $FL \subset X$. In our study, $FL =$ {um, uh, like, you know} (the filler instances of "like" and "you know" are considered different tokens than non-filler instances of the same words in the modelling vocabulary). Assume an input sequence $x_0, ..., x_T$ free of fillers. We denote the output sentence of the model (with zero or more inserted fillers) as $x'_0, ..., x'_{T'}$, where $T' \geq T$. First, we put $x'_0 = x_0$. Note that we front-append the starting token [START] to input sentence prior to the insertion process, thus $x'_0 = x_0 =$ [START]. This allows us to learn the distribution of the filler words at the beginning of the sentence without any context. We then sample from the language model distribution at time step t,

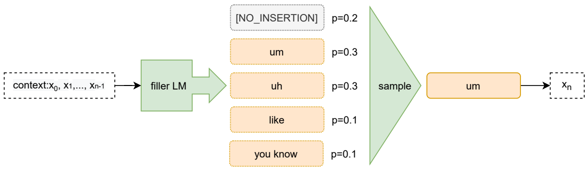

Figure 1: Single sampling filler insertion step.

$$x_t^{sampled} \sim P(X|x'_0, ..., x'_{t-1}),$$

where $x'_0, ..., x'_{t-1}$ is the sub-sequence of the output prior to t (with zero or more filler words inserted). If $x_t^{sampled} \in FL$, then $x'_t = x_t^{sampled}$, which is an insertion. Otherwise, $x_t^{sampled} \notin FL$, i.e. no insertion, then x'_t is taken from the next token in the input sequence. In some cases, we also explicitly model no insertion token [NO_INSERTION] to simplify modeling. An example run of this sampling-based filler insertion process on a sentence is shown in Figure 2.

In perceptual evaluations described in section 5, we additionally enforce an upper limit on the number of fillers inserted per-sentence for all models. We do this by repeatedly sampling full insertion patterns on an input sentence until getting an insertion pattern that has the number of fillers ≤ upper limit. Note that we do not stop sampling mid-sentence when the limit has been reached, instead we choose the first full sampled insertion pattern with the number of fillers less than the upper limit. Our approach can be seen as sampling from the subset of the distribution of all full insertion patterns that meet the upper limit requirement. We enforce this upper limit because we find that the most common failure mode of the sampling-based insertion is inserting too many fillers or run-on insertion such as "... [um] [um] [um] ...". For all experiments in the perceptual evaluation, we enforce an upper limit of 3 fillers per sentence.

4.2. Filler Perplexity

We propose a modified perplexity measure for objective evaluation of filler insertion models. Assume a corpus $X = x_0 x_1 ... x_T$ and a language model with parameters θ that is trained on the corpus, then perplexity is defined as,

$$PP(X, \theta) = \sqrt[n]{\frac{1}{P_\theta(x_0 x_1 ... x_T)}},$$

for which the auto-regressive expansion is,

[1] https://github.com/NVIDIA/tacotron2

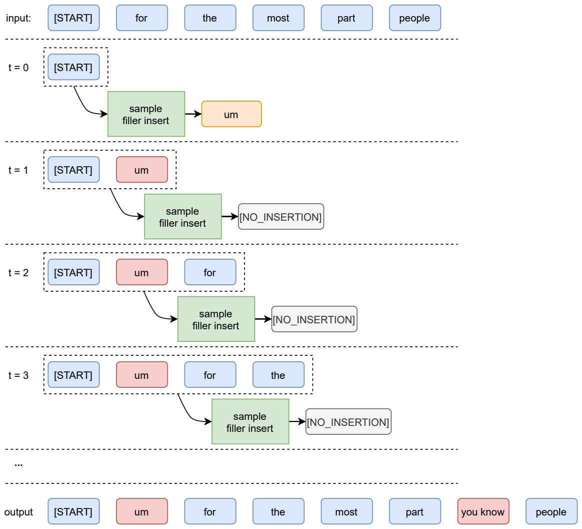

Figure 2: Sampling filler insertion on a sentence.

$$PP(X, \theta) = \sqrt[T]{\frac{1}{\prod^{t=0,\ldots,T} P_\theta(x_t|x_0, \ldots, x_{t-1})}}, \quad (1)$$

The model with smaller perplexity is considered to fit the corpus better. The intuition is the same as maximum likelihood estimation (MLE), i.e. a good model should assign ground truth data with high probability. However, in the filler insertion setting, we are mainly concerned with how well a model learns contextual distribution of fillers and not other words in vocabulary. This can be reflected by a filler perplexity measure modified from the original perplexity measure as,

$$FPP(X, FL^*, \theta) = \sqrt[T]{\frac{1}{\prod^{t=0,\ldots,T} P_\theta(fl_t^*|x_0, \ldots, x_{t-1})}}, \quad (2)$$

Here, $fl^* \in FL^* = FL \cup \{\text{[NO\_INSERTION]}\}$, and in this study, $FL^* = \{$um, uh, like, you know, [NO_INSERTION]$\}$. [NO_INSERTION] is the no-filler token given at positions without fillers. For example, the filler perplexity FPP of the sentence "I am um happy" is,

$$FPP(\text{“I am um happy”}, FL^*, \theta) = \sqrt[T]{\frac{1}{P}}, \quad (3)$$

$$\begin{aligned} P = {} & P_\theta(\text{[NO\_INSERTION]}|\text{[START]})\cdot \\ & P_\theta(\text{[NO\_INSERTION]}|\text{[START], “I”})\cdot \\ & P_\theta(\text{“um”}|\text{[START], “I”, “am”})\cdot \\ & P_\theta(\text{[NO\_INSERTION]}| \\ & \quad \text{[START], “I”, “am”, “um”})\cdot \\ & P_\theta(\text{[NO\_INSERTION]}| \\ & \quad \text{[START], “I”, “am”, “um”, “happy”}). \end{aligned}$$

FPP is proportional to the product of model perplexity at filler positions (FPP_1) and non-filler positions (FPP_0). Let $T_{FL} = \{t \in \{0, \ldots, T\} | fl_t^* \in FL\}$ denote the time steps that are filler, and let $T_{\text{[NI]}} = \{t \in \{0, \ldots, T\} | fl_t^* = \text{[NO\_INSERTION]}\}$ denote the set of time steps that are not fillers, then the full filler perplexity is proportional to,

$$\begin{aligned} & FPP(X, FL^*, \theta) \sim \\ & \sqrt[|T_{FL}|]{\frac{1}{\prod^{t \in T_{FL}} P_\theta(fl_t|x_0, \ldots, x_{i-1})}}\cdot \\ & \sqrt[|T_{\text{[NI]}}|]{\frac{1}{\prod^{t \in T_{\text{[NI]}}} P_\theta(\text{[NO\_INSERTION]}|x_0, \ldots, x_{t-1})}} \\ & \quad = FPP_1(X, FL, T_{FL}, \theta)\cdot \\ & \qquad FPP_0(X, T_{\text{[NI]}}, \theta), \quad (4) \end{aligned}$$

The subscripts 1 and 0 of FPP_1 and FPP_0 reflect that FPP_1 is calculated on filler positions T_{FL} while FPP_0 on non-filler positions $T_{\text{[NO\_INSERTION]}}$. A low FPP_1 means that the model gives high probability to the correct filler insertions at the correct insertion positions. The second part FPP_0 calculates model perplexity on non-filler positions. A low FPP_0 suggests that the model gives high probability to [NO_INSERTION] token at the correct non-filler positions. Thus, a low FPP_0 model is good at *not* inserting fillers at positions where fillers are not appropriate. As seen here, looking at FPP through FPP_1 and FPP_0 helps us understand model performance in greater detail than a single FPP score. This also reveals a potential trade-off in modeling filler insertion. A low FPP_0 model could be biased towards no filler insertion overall which leads to a high FPP_1 . The opposite is true when a low FPP_1 model is biased towards inserting fillers irrespective of position and thus has a high FPP_0 . Such a trade-off is analogous to the trade-off between precision and recall in an F1 score.

4.3. Filler Insertion Models

We build 3 models for filler insertion: n-gram (n=3, with KN smoothing) (Kneser and Ney, 1995)), LSTM-LM (Sundermeyer et al., 2012) and modified GPT-2 (Radford et al., 2019). We use the NLTK implementation for the n-gram language model which comes with KN smoothing built in (Bird and Loper, 2004). To choose the hyperparameter n, we calculate FPP with varying n from 1 to 5 as shown in Figure 3. We found that increasing n results in higher perplexity scores in all three categories with the exception of n=1. This is reasonable because as n increases, the context length increases, it becomes more difficult to match context seen in test time with what the model has seen during training. Thus, the higher-n models tend to miss more fillers at test time resulting in higher FPP_1 . However, this has little effect on FPP_0 as it sees little change as n increases. This result suggests using n=2. However, through informal tests, we found that n=3 gives slightly better insertion perceptually, so we choose n=3

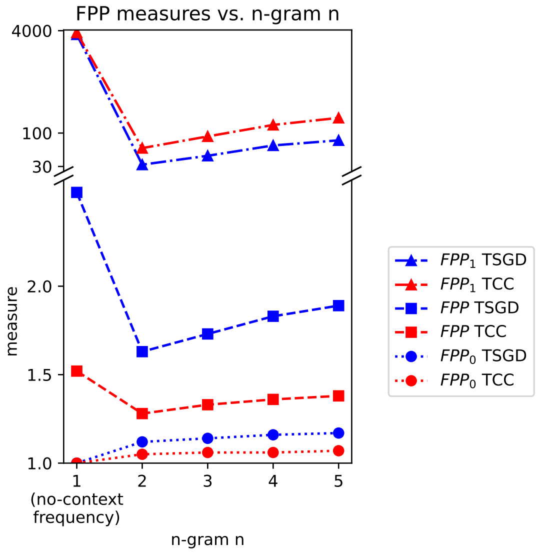

Figure 3: FPP measures vs n-gram n. All measures increase monotonically as n (context length) increases except for n=1 (no-context, just frequency itself).

for our study. We note that the choice of n=3 is different from prior studies where a larger n was used, n=4 (Dall et al., 2014), n=6 (Tomalin et al., 2015). This difference could be due to different corpora being used in the studies. We build an LSTM-LM by first building a vocabulary from the training set with rare word threshold set to 2. The LSTM-LM has 2 hidden layers with 1024 hidden units per layer, the word embedding has dimension 128. It is trained with a between-layer dropout of 0.5 and learning rate 1e-6.

GPT-2 (Radford et al., 2019) is a large-scale language model trained auto-regressively on a large text corpus from various online sources. It can be fine-tuned to down-stream language tasks efficiently (Radford et al., 2018). We can not simply fine-tune GPT-2 on the spontaneous corpus to model fillers due to GPT-2 being pre-trained on written text without explicit differentiation between fillers and non-fillers, especially for “like” and “you know” which have both filler and non-filler instances in a spontaneous corpus. We add another output branch to GPT-2, a single 5-way softmax prediction layer that predicts the probability of four filler words plus the probability of no filler insertion, which is used for filler insertion instead of GPT-2’s own language modelling output branch. However, the model is trained with combined loss of the original language modelling branch and the added filler modeling branch to adapt the model to the spontaneous corpus better.

4.4. Results of *FPP* Evaluation

The calculated perplexity scores including the full FPP and its two factored components on the test set of the two corpora are shown in Table 2. Both the 3-gram model and the modified GPT-2 obtain low FPP_0 scores on the two datasets, suggesting that they are very certain about locations where there should not be any filler insertion. This is a major reason why their FPP is low as there are more non-fillers than fillers (Table 1). However, the modified GPT-2 is also better at modeling insertion location than the 3-gram model as it obtains a lower FPP_1 score on both datasets and lowest of the three on TCC dataset. It also obtains the lowest FPP on both datasets. This could be because of its much larger model capacity and large-scale pretraining that helps it fit to novel texts well. However, the 3-gram model obtains a level of FPP similar to GPT-2. LSTM-LM is an outlier as it obtains relatively low FPP_1 but very high FPP_0 . This suggests that it considers all locations to be possible insertion locations with substantial probability. It also receives the highest FPP with a large margin, suggesting that it simply does not fit the datasets well. This is potentially due to the fact that it has to learn good word embeddings from scratch and the corpora are not big enough to do that.

To understand between-speaker differences, we calculate filler perplexity cross-corpus, that is we apply models trained on one corpus to the other. The results are shown in Table 3. We highlight two aspects of the results. The TSGD-trained models obtain lower FPP and FPP_0 cross-corpus than on TSGD’s own test set. However surprising, this could be explained by lower cross-corpus FPP_0 which means that the models are conservative about inserting fillers at unseen context, and since there are more non-filler positions than fillers, a low FPP_0 has a bigger influence on FPP than a higher FPP_1 . But we also note that this phenomenon is not present in TCC-trained models. The second interesting aspect is that FPP_1 increases significantly for all models in the cross-corpus setting. This demonstrates that the usage of fillers by the two speakers are different. Either they do not use the same fillers at particular positions, or they use fillers at different positions.

5. Perceptual Study

5.1. Evaluation Setup

5.1.1. Evaluation 1: Model Comparison

We evaluate the synthesized speech with filler-inserted text from the three models with carefully designed user studies. A single mean opinion score (MOS) test takes in one input sentence. We apply three models to insert fillers by the aforementioned sampling method. We then add a fourth grounding stimulus, which is groundtruth (resynthesized with the same vocoder as the TTS) for in-data sentences and no filler insertion for chatbot sentences. The difference between the two types of sentences is described later in this section. The filler-inserted sentences are fed into the same TTS trained on the same dataset that the filler insertion models are trained on. The four stimuli (3 from filler models and 1 grounding stimulus) are presented side-by-side in random order to listeners. Subjects are asked the to rate how natural each sample sounds on the scale of 1-5. To

	TSGD			**TCC**		
Models	$\mathbf{FPP_1}$	$\mathbf{FPP_0}$	$\mathbf{FPP}$	$\mathbf{FPP_1}$	$\mathbf{FPP_0}$	$\mathbf{FPP}$
3-gram (KN smoothing)	42.97	1.14	1.68	61.60	1.05	1.32
LSTM-LM	7.53	16.76	16.01	11.96	22.34	21.89
Modified GPT-2	34.46	1.08	1.64	10.06	1.03	1.17

Table 2: Perplexity measures on in-data test set.

	TSGD(trained)⇒TCC(tested)			**TCC(trained)⇒TSGD(tested)**		
Models	$\mathbf{FPP_1}$	$\mathbf{FPP_0}$	$\mathbf{FPP}$	$\mathbf{FPP_1}$	$\mathbf{FPP_0}$	$\mathbf{FPP}$
3-gram (KN smoothing)	698.12	1.12	1.59	4845.79	1.08	2.68
LSTM-LM	19.49	18.92	18.94	42.77	18.99	19.92
Modified GPT-2	54.52	1.06	1.35	47.92	1.03	1.54

Table 3: Cross-corpus perplexity measures.

	TSGD		**TCC**	
Models	**in-data**	**chat-bot**	**in-data**	**chat-bot**
3-gram (KN smoothing)	**3.61 ± 0.12**	**3.39 ± 0.13**	* **3.78 ± 0.11**	* **3.34 ± 0.12**
LSTM-LM	3.55 ± 0.12	3.26 ± 0.12	3.65 ± 0.10	3.26 ± 0.13
Modified GPT-2	3.60 ± 0.12	3.36 ± 0.13	3.63 ± 0.12	3.19 ± 0.13
GT (in-data) / no-filler (chat-bot)	3.55 ± 0.12	3.30 ± 0.13	3.64 ± 0.11	3.29 ± 0.13

Table 4: MOS score. n=30 for both datasets (two separate groups of testers). 10 sentences for each sub-category (column). p=0.05 confidence intervals are shown. * if better than the worst in related-sample t-test with $p < 0.05$.

encourage listeners to make judgements based on both content and audio, the text transcripts of the stimuli are also provided on the same page, where the filler insertion is marked with brackets.

To thoroughly evaluate the filler insertion models, we form two separate groups of test sentences by choosing 10 filler-word-removed sentences from the test set of TSGD and TCC, which we refer to as *in-data* test sentences. We also chose 10 sentences from *chat-bot* generated texts (Zhang et al., 2020; Adiwardana et al., 2020; Roller et al., 2021; See et al., 2019). We form one MOS test for each sentence. Thus, both TSGD and TCC tests have 20 input sentences with 10 in-data and 10 chat-bot. For each test, the 10 in-data sentences and the 10 chat-bot sentences are mixed and presented in a randomized order for each user. This setup enables evaluating the difference between the two categories of sentences, in addition to comparing the models. Only one sampled insertion from each model is evaluated for each input sentence, so the randomness in sampling may add noise to the evaluation. However, we found that sampled insertion patterns on the same input sentence often coincide. It is of course possible to add more tests for the same input sentences. However, we decided against this, to limit listeners' exposure to repetitious sentences, which can be exasperating and result in less reliable ratings. The randomness introduced by sampling is mitigated by having 20 distinct test sentences per test.

5.1.2. Evaluation 2: Does filler insertion make TTS for chat-bots sound more spontaneous?

A major motivation for this work lies in improving spontaneous TTS for conversational systems by inserting fillers to the response generated from a dialogue system. We design a user study using the same 10 chat-bot sentences from Evaluation 1. For each sentence, we synthesize two versions, one has no fillers and the other one is filler-inserted by the best model (3-gram) from Evaluation 1. Listeners are asked to choose which version sounds more conversational and which version sounds better overall (two separate questions). Subjects can choose one of the two versions for preference or choose "neither". We ask overall preference mainly to find out if adding fillers makes the resulting speech worse and if there is no clearly worse speech, we expect subjects to choose the "neither" option, since both samples are synthesized with the same TTS. We use *two* sampled insertions on the same input sentence, yielding 20 total tests for each dataset and 40 test pages in total, We are able to increase the number of tests to 40 from 20 in Evaluation 1 because choosing preference between 2 samples is a simpler task. All stimuli were presented in randomized order [2].

Finally, as a reference we performed the same listening test where all samples were synthesized using a state-of-the-art speech synthesis system (Battenberg et al., 2020), trained on the read speech corpus LJSpeech (Ito and Johnson, 2017).

5.2. Results of the Perceptual Evaluation

5.2.1. Results of Evaluation 1: Model Comparison

We recruited 30 native speakers of English through the crowdsourcing platform Prolific for the two model comparison tests on TSGD and TCC, totalling 60 testers. Results are shown in Table 4. The 3-gram

[2] Audio samples from both Evaluation 1 and 2 are available at `www.speech.kth.se/tts-demos/LREC22`.

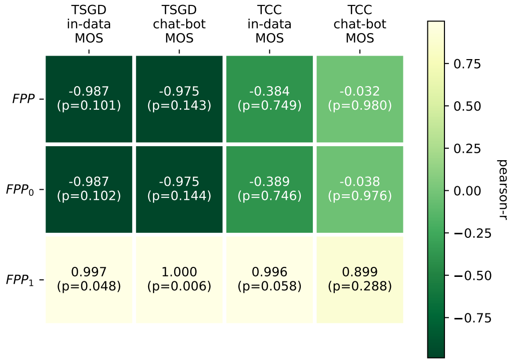

Figure 4: Correlation between FPP and MOS.

model outperforms others in both datasets in both datatypes (in-data and chat-bot). These results suggest that 3-gram not only performs well on in-data test sentences but also generalizes well to unseen chat-bot sentences. It is surprising that ground-truth (GT) scored lowest in TSGD and second to lowest in TCC. We believe that this is due to users not liking fillers when asked to explicitly listening to them. The GT speech in TSGD has a lot more fillers than the model inserted versions (for which we set an upper limit of 3 fillers per sentence), which may have negatively impacted its scores. This hypothesis is supported by an ANOVA analysis on the results (Table 5), which shows that the rating is negatively correlated with the number of fillers irrespective of model with $p < 0.002$.

The ANOVA also reveals that there is little significant difference between models, which agrees with the pairwise statistical test results in Table 4. Even though 3-gram scores highest on average, the difference is not significant in most cases. This is partially due to the fact the input test sentences come from the test set of filler insertion modeling but are included in training set of the TTS, thus in some cases the TTS is able to reproduce the prosody of the GT. However, in other cases TTS still produces good spontaneous prosody different than GT suggesting the neural TTS engine's strong generalization ability. This brings all samples to similar level of naturalness prosody-wise and makes it different to differentiate different filler insertion patterns on the same input sentence.

We quantify the correlation between all three FPP measures and the MOS scores with Pearson-r as shown in Figure 4. FPP_1 is strongly correlated with MOS. The correlation between FPP and FPP_1 and MOS is corpus-dependent. We note that the correlation between FPP_1 and MOS is consistently close to 1 in both datasets. However lower FPP_1 suggests that the model is better at modeling filler positions [3], thus we would expect a negative correlation between FPP_1 and MOS. This result indicates that for a model to obtain high perceptual MOS, it needs to have a level of bias towards fewer insertions, i.e. having higher FPP_1 . This is confirmed by our finding that increasing the number of fillers inserted regardless of model has a negative impact on MOS (Table 5).

We can reliably compare MOS obtained for the two categories, in-data and chat-bot, because sentences from both categories are mixed together and randomly shuffled for each user (section 5.1.1). It can be seen in Table 4 that filler-inserted in-data sentences obtain overwhelmingly higher MOS than filler-inserted chat-bot sentences irrespective of filler insertion models or training corpus, a result further confirmed in ANOVA Table 5 where chat-bot sentences have a significant negative coefficient to rating ($p < 0.001$). This could be explained by the fact that the TTS is trained on the same dataset as the in-data sentences but not chat-bot sentences, thus the TTS would sound more natural for in-data sentences. This result suggests that filler insertion alone is not enough to make chat-bot sentences sound more natural with spontaneous TTS, and a full-scale text style transfer (Fu et al., 2018) maybe needed.

5.2.2. Results of Evaluation 2: Filler vs. No-filler

30 listeners from Prolific completed this study. The results are shown in Table 6. The results using spontaneous TTS trained on the two datasets (TSGD, TCC) show that there is a clear preference of filler-inserted speech being more conversational. This shows that filler insertion makes chat-bot sentences sound more conversational. It is important to point out that the raters are simply given two utterances and asked to choose which one is more conversational. They are not told that the two side-by-side compared TTS utterances differ in one inserted fillers while the other not. They are also not given transcripts which could easily expose this fact. The overall preference results in the two spontaneous TTS show that filler-inserted speech is preferred in the TCC dataset, while the no-filler inserted speech has higher preference in the TSGD dataset. This could be explained by the type of fillers they inserted, the TCC-model mostly inserted "uh" and "um", while the TSGD model mostly inserted "like". Even though "like" can be used to mark a more casual conversational style, many think that it is a marker of lack of intelligence that people should best avoid (Tree, 2006). Furthermore, the preference towards no-filler speech in TSGD does not obtain more than 50%. This suggests that filler insertion makes the voice sound more conversational without significant deterioration in perceived overall speech quality in either dataset.

The results for the LJSpeech voice show clearly that it cannot handle fillers well. Adding fillers makes it sound worse and does not make it sound more conversational. The preference for both conversational and overall almost completely flipped when switching the TTS from LJSpeech to spontaneous. This shows the importance of evaluating filler insertion in high-quality spontaneous TTS, otherwise the results may be negatively affected by the shortcomings of the TTS.

[3] Lower is better for all FPP , FPP_0 , FPP_1 .

	TSGD		TCC	
	coef	**P> |t|**	**coef**	**P> |t|**
Intercept	3.7536	0.000	3.8604	0.000
C(model)[T.LSTM-LM]	-0.0899	0.192	-0.0933	0.160
C(model)[T.GPT-2]	-0.0009	0.989	-0.1562	0.019
C(model)[T.GT/no-filler]	-0.1086	0.116	-0.1069	0.121
C(datatype)[T.chat-bot]	-0.2558	0.000	-0.4290	0.000
Filler count	-0.0619	0.002	-0.0633	0.024

Table 5: ANOVA multi-factor analysis results, *MOS* $\sim$ *C(model) + C(datatype) + filler count.* C(model) = {3-gram, LSTM-LM, GPT-2, GT(in-data)/no-filler(chat-bot)}. C(datatype) = {in-data, chat-bot}.

	Filler model trained on TSGD			
	LJSpeech TTS		**TSGD TTS**	
	conversational	**overall**	**conversational**	**overall**
prefer no-filler	* **65.1 ± 0.55%**	* **79.6 ± 0.46%**	33.6 ± 0.54%	* **47.3 ± 0.57%**
neither	15.0 ± 0.41%	12.9 ± 0.38%	11.1 ± 0.36%	16.1 ± 0.42%
prefer filler-inserted	19.9 ± 0.46%	7.5 ± 0.30%	* **55.2 ± 0.57%**	36.6 ± 0.55%
	Filler model trained on TCC			
	LJSpeech TTS		**TCC TTS**	
	conversational	**overall**	**conversational**	**overall**
prefer no-filler	* **52.4 ± 0.57%**	* **67.9 ± 0.54%**	23.1 ± 0.48%	34.1 ± 0.54%
neither	16.3 ± 0.42%	16.0 ± 0.42%	13.2 ± 0.39%	21.5 ± 0.47%
prefer filler-inserted	31.3 ± 0.53%	16.1 ± 0.42%	* **63.7 ± 0.55%**	* **44.4 ± 0.57%**

Table 6: Preference ratio on no-filler vs. filler-inserted chat-bot sentences. n=30, 20 sentences for each dataset category. p=0.05 confidence intervals are shown. * if better than the next best alternative with $p < 0.05$.

6. Discussion

We observed that the neural spontaneous TTS gives highly natural speech almost regardless of filler insertion models. At the same time, regardless which filler insertion model is used, the neural read speech TTS lead to decreased overall quality and colloquialness. This suggests that performance of filler insertion models is dependent on the TTS, and that the two parts (filler insertion and TTS) should be developed in conjunction if they are intended to be used in the same system.

The perceptual study results show that increasing the number of fillers has a significant negative impact on MOS (Table 5). We believe that this is the result of users being told to evaluate "filler-inserted speech". Such instruction, common in most filler insertion studies, directs listeners' attention to fillers, the overuse of which people generally dislike even in regular conversation with another human. Future work could try other evaluation schemes such that fillers are implicitly evaluated rather than explicitly, for example, during a human-robot interaction.

We only tested the sampling-based approach to filler insertion and did not compare with single-prediction approaches such as lattice rescoring (Tomalin et al., 2015). We leave this comparison to future studies. However, we observe that sampled insertions on the same sentence often coincide, suggesting that there are only a few insertions with high probability, which implies that our sampling approach likely arrives at similar results as prediction approaches.

7. Conclusions

We built sampling-based filler insertion models and evaluated them with a proposed quantitative metric filler perplexity FPP and perpetual evaluations. Three models are built and trained on two single-speaker corpora. FPP is useful in analyzing the models but does not correlate well with perceptual MOS. The perceptual study has two innovations that address shortcomings of prior studies, (1) with chat-bot generated sentences in test input and, (2) using a neural spontaneous TTS to synthesize speech. The difference in MOS obtained by different models (including a grounding stimulus) is not statistically significant, partially due to high naturalness of neural spontaneous TTS regardless of input and ambiguity of what a good filler insertion is. This suggests that evaluating filler insertion synthesized by high-quality spontaneous TTS is hard with generic naturalness criteria. In the filler/no-filler comparison test with spontaneous TTS, filler-inserted speech is shown to have higher perceived colloquialness, but the result flips when using read speech TTS, potentially due to its inability to voice realistic-sounding fillers. This shows the importance of evaluating filler insertions with spontaneous TTS.

8. Acknowledgements

This research is supported by the Swedish Research Council project Connected (VR-2019-05003), the Riksbankens Jubileumsfond project CAPTivating (P20-0298) and the Digital Futures project Advanced Adaptive Intelligent Systems (AAIS).

9. Bibliographical References

Adell, J., Escudero, D., and Bonafonte, A. (2012). Production of filled pauses in concatenative speech synthesis based on the underlying fluent sentence. *Speech Communication*, 54(3):459–476.

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. (2020). Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Andersson, S., Georgila, K., Traum, D., Aylett, M., and Clark, R. A. (2010). Prediction and realisation of conversational characteristics by utilising spontaneous speech for unit selection. In *Proc. Speech Prosody 2010-Fifth International Conference*.

Battenberg, E., Skerry-Ryan, R., Mariooryad, S., Stanton, D., Kao, D., Shannon, M., and Bagby, T. (2020). Location-relative attention mechanisms for robust long-form speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6194–6198. IEEE.

Bird, S. and Loper, E. (2004). NLTK: the natural language toolkit. Association for Computational Linguistics.

Brennan, S. E. and Williams, M. (1995). The feeling of another s knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of memory and language*, 34(3):383–398.

Clark, H. H. and Tree, J. E. F. (2002). Using uh and um in spontaneous speaking. *Cognition*, 84(1):73–111.

Cohn, M., Chen, C.-Y., and Yu, Z. (2019). A large-scale user study of an alexa prize chatbot: Effect of tts dynamism on perceived quality of social dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 293–306.

Corley, M., MacGregor, L. J., and Donaldson, D. I. (2007). It's the way that you, er, say it: Hesitations in speech affect language comprehension. *Cognition*, 105(3):658–668.

Dall, R., Tomalin, M., Wester, M., Byrne, W., and King, S. (2014). Investigating automatic & human filled pause insertion for speech synthesis. In *Proc. Interspeech*.

Dall, R., Tomalin, M., and Wester, M. (2016). Synthesising filled pauses: Representation and datamixing. In *Proc. 9th ISCA Speech Synthesis Workshop (SSW 9)*, pages 7–13.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., and Weston, J. (2018). Wizard of wikipedia: Knowledge-powered conversational agents. In *Proc. International Conference on Learning Representations*.

Ferstl, Y. and McDonnell, R. (2018). Investigating the use of recurrent motion modelling for speech gesture generation. In *Proc. IVA*, pages 93–98.

Fraundorf, S. H. and Watson, D. G. (2011). The disfluent discourse: Effects of filled pauses on recall. *Journal of memory and language*, 65(2):161–175.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2018). Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 663–670.

Gravano, A. and Hirschberg, J. (2011). Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3):601–634.

Gustafson, J., Beskow, J., and Székely, É.. (2021). Personality in the mix-investigating the contribution of fillers and speaking style to the perception of spontaneous speech synthesis. In *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, pages 48–53.

Hu, Z., Lee, R. K.-W., and Aggarwal, C. C. (2020). Text style transfer: A review and experiment evaluation. *arXiv preprint arXiv:2010.12742*.

Ito, K. and Johnson, L. (2017). The LJ speech dataset. `https://keithito.com/LJ-Speech-Dataset/`.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184. IEEE.

Kong, J., Kim, J., and Bae, J. (2020). Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033.

Laserna, C. M., Seih, Y.-T., and Pennebaker, J. W. (2014). Um... who like says you know: Filler word use as a function of age, gender, and personality. *Journal of Language and Social Psychology*, 33(3):328–338.

Levelt, W. J. (1983). Monitoring and self-repair in speech. *Cognition*, 14(1):41–104.

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (2017). Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Maclay, H. and Osgood, C. E. (1959). Hesitation phenomena in spontaneous English speech. *Word*, 15(1):19–44.

Nguyen, T. A., Kharitonov, E., Copet, J., Adi, Y., Hsu, W.-N., Elkahky, A., Tomasello, P., Algayres, R., Sagot, B., Mohamed, A., et al. (2022). Generative spoken dialogue language modeling. *arXiv preprint arXiv:2203.16502*.

Qader, R., Lecorvé, G., Lolive, D., and Sébillot, P. (2018). Disfluency insertion for spontaneous tts: Formalization and proof of concept. In *International Conference on Statistical Language and Speech Processing*, pages 32–44.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Radford, A., Narasimhan, K., Salimans, T., and

Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y.-L., et al. (2021). Recipes for building an open-domain chatbot. In *Proc. 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325.

See, A., Roller, S., Kiela, D., and Weston, J. (2019). What makes a good conversation? How controllable attributes affect human judgments. In *Proc. NAACL-HLT*, pages 1702–1723.

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

Sundaram, S. and Narayanan, S. (2003). An empirical text transformation method for spontaneous speech synthesizers. In *Proc. Eurospeech*, pages 1221–1224.

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). LSTM neural networks for language modeling. In *Proc. Interspeech*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Székely, É., Henter, G., Beskow, J., and Gustafson, J. (2019a). How to train your fillers: uh and um in spontaneous speech synthesis. In *Proc. 10th ISCA Speech Synthesis Workshop (SSW 10)*, volume 10, pages 245–250.

Székely, É., Henter, G. E., Beskow, J., and Gustafson, J. (2019b). Spontaneous conversational speech synthesis from found data. In *Proc. Interspeech*, pages 4435–4439.

Székely, É., Henter, G. E., Beskow, J., and Gustafson, J. (2019c). How to train your fillers: uh and um in spontaneous speech synthesis. In *The 10th ISCA Speech Synthesis Workshop*.

Székely, É.., Henter, G., Beskow, J., and Gustafson, J. (2020). Breathing and speech planning in spontaneous speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7649–7653. IEEE.

Tomalin, M., Wester, M., Dall, R., Byrne, W., and King, S. (2015). A lattice-based approach to automatic filled pause insertion. In *DiSS The 7th Workshop on Disfluency in Spontaneous Speech*.

Tree, J. E. F. (2006). Placing like in telling stories. *Discourse studies*, 8(6):723–743.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too? In *Proc. 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, W. B. (2020). Dialogpt: Large-scale generative pretraining for conversational response generation. In *Proc. 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.