A LATTICE-BASED APPROACH TO AUTOMATIC FILLED PAUSE INSERTION

Marcus Tomalin[1], Mirjam Wester[2], Rasmus Dall[2], Bill Byrne[1], & Simon King[2]

[1]Cambridge University Engineering Department, University of Cambridge, UK

[2]The Centre for Speech Technology Research, University of Edinburgh, UK
mt126@cam.ac.uk, r.dall@sms.ed.ac.uk, mwester@inf.ed.ac.uk, wjb31@cam.ac.uk, Simon.King@ed.ac.uk

ABSTRACT

This paper describes a novel method for automatically inserting filled pauses (e.g., *UM*) into fluent texts. Although filled pauses are known to serve a wide range of psychological and structural functions in conversational speech, they have not traditionally been modelled overtly by state-of-the-art speech synthesis systems. However, several recent systems have started to model disfluencies specifically, and so there is an increasing need to create disfluent speech synthesis input by automatically inserting filled pauses into otherwise fluent text. The approach presented here interpolates Ngrams and Full-Output Recurrent Neural Network Language Models (f-RNNLMs) in a lattice-rescoring framework. It is shown that the interpolated system outperforms separate Ngram and f-RNNLM systems, where performance is analysed using the Precision, Recall, and F-score metrics.

Keywords: Disfluency, Filled Pauses, f-RNNLMs, Ngrams, Lattices

1. INTRODUCTION

In recent years, disfluent speech synthesis has started to receive more attention [1, 2, 3, 13]. The aim is to develop systems that produce convincing disfluencies such as filled pauses (FPs), discourse markers, repetitions, and restarts. It is well-known that such phenomena serve a wide range of important functions in conversational discourse. They can indicate psychological states [11], structure discourses [9], facilitate word recall [14], and improve word recognition [15, 10, 12]. Given this, it is desirable to model them overtly in automatic speech synthesis systems which seek to approximate a human-like conversational style.

The broad motivations underlying research into disfluent synthesis are closely related to those that have prompted the development of emotional or expressive speech synthesis systems [20, 19, 17, 4, 5]. Both endeavours ultimately seek to create synthetic speech that is able to convey a wider range of emotional or psychological states, thereby producing synthetic voices that can simulate certain character and personality types more convincingly. The main difference, however, is that while emotional or expressive speech synthesis concentrates primarily on modifying prosodic phenomena (such as pitch, speech rate, voice quality, inter-lexical pause duration) [20, 19, 17, 4, 5], disfluent speech synthesis additionally requires the augmentation of the (fluent) input token sequence [13].

This paper contributes to this ongoing endeavour by extending the basic approach to automatic FP-insertion introduced in [13]. That paper focused on the relatively simple task of inserting a single FP (*UH*) into a fluent token sequence at an appropriate Insertion Point (IP). By contrast, the current paper describes a system that can insert *multiple* FPs in *multiple* IPs. Therefore the sentence 'I NEVER LIKED GAMES' could be modified automatically to become '**UM** I NEVER LIKED **UH** GAMES'. In addition, the new system has a Disfluency Parameter (DP) that determines the degree of disfluency in the output text. The DP takes a value in the range $[0, 1]$, where 0 = maximally fluent and 1 = maximally disfluent. Finally, while [13] used simple linear interpolation of word-level Ngram and RNNLM probabilities to rerank the potential sentences, a more robust lattice-based rescoring method is introduced here. As a modelling technique, it has clear advantages since simple re-ranking strategies become computationally inefficient when multiple FPs can be inserted in multiple IPs.

The structure of this paper is as follows. Section 2 describes the lattice-based modelling framework, and provides information about the training and test data used. Section 3 gives the results for the various FP-insertion systems compared using the Precision, Recall, and F-score metrics. Scores are given at the sentence level for output containing all the inserted FPs, along with breakdowns for each separate

FP subtype. The main conclusions and directions for future research are summarised in 4.

2. LATTICE-BASED LM INTERPOLATION

The lattice-based FP-insertion system developed here is similar to those recently implemented for Automatic Speech Recognition (ASR) tasks in [7, 18]. In the context of ASR language models (LMs), RNNLMs have become increasingly popular in recent years due to their inherently strong generalization performance. Specifically, Chen et al 2014 [7] has shown that f-RNNLMs facilitate an efficient parallisation of training in a Graphics Processing Unit (GPU) implementation. In addition, when used in a lattice-rescoring framework, they give both Perplexity and Word Error Rate improvements over standard RNNLMs. This is due in part to their use of an unclustered 'Full-Output' architecture.

This framework can be adapted for the FP-insertion task. There are five main stages in the modified process:

1. Create initial lattices in which each FP is accessible from each word token (Figure 1)
2. Expand the initial lattices using an Ngram (6g)
3. Rescore the expanded lattices using an interpolated LM with weighted Ngram and f-RNNLM sub-components
4. Output an *n*-best list for each sentence (where $n = 10000$)
5. Specify the desired degree of disfluency using the DP and generate final 1-best disfluent output

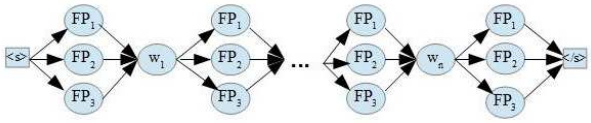

Figure 1: Example Initial Lattice for *n* words and 3 FPs

After FP-insertion, the versions of sentence *S* in the *n*-best list will have varying token counts since they will contain different numbers of automatically inserted FPs. All versions of *S* with *p* tokens are rank-ordered using the sentence-level interpolated LM score, and the 1-best version is output. The closed interval $[0,1]$ is divided equally between the various 1-best outputs for different *p* values. This provides the DP that determines the degree of disfluency. The impact of varying the DP parameter is shown in Table 1. This example provides a concrete instance of the impact that the DP value has on the resulting token sequences, and it illustrates the graded nature of the different DIS-insertion outputs. In particular, it shows how the perceived psychological state of the (synthetic) speaker can be altered as a fluent lexical sequence becomes increasingly disfluent.

DP	Output Sentence
0.00	WELL I GUESS THEY WERE SAYING
0.25	WELL I GUESS THEY WERE SAYING **UM**
0.50	**UM** WELL **UH** I GUESS THEY WERE SAYING **UM**
0.75	**UM** WELL **UH** I GUESS **HM** THEY WERE SAYING **UM**
1.00	**UM** WELL **UH** I GUESS **HM** THEY **UH** WERE SAYING **UM**

Table 1: An example of the impact of DP values on output disfluent token sequence

The LMs used in the experiments were trained on 20M words (1M sentences) of data from the Switchboard, Fisher, and AMI corpora, as well as an unreleased corpus of British conversational telephone speech [16, 8, 6]. Dev and Test sets were extracted from different subsets of the same corpora, and they comprised 7,365 sentences (145k words) and 6,910 sentences (139K words) respectively. Each sentence in the scoring reference contained at least one FP, and these FPs were removed to create the 'fluent' version of the test sets that were processed by the FP-insertion systems. The purpose of the experiments was to see whether the systems could insert the same FPs into the same IPs as those found in the scoring reference files. Seven FPs in total were modelled overtly by the various FP-insertion systems: *UH*, *UM*, *OH*, *UHUM*, *UHU*, *HM*, and *AH*. Information about the occerrence of these FPs in the training data is given in Table 2.

	#occs [%]
UH	213,924 [1.09%]
UM	200,499 [1.02%]
OH	123,028 [0.63%]
AH	69,288 [0.35%]
UHUM	29,515 [0.15%]
UHU	16,180 [0.08%]
HM	3,456 [0.01%]

Table 2: FP occurrence counts for the training data (and % of training data)

As the counts in Table 2 indicate, the FPs *UH* and *UM* occur most frequently in the training data. The fact that some of the other FPs have relatively low counts ($<$30,000) facilitates the exploration of the impact of data sparcity on the modelling of speech disfluencies.

3. EXPERIMENTS AND RESULTS

Three FP-insertion systems were compared:

1. **Ngram**: a standard 6g built using the training data; SRILM toolkit [21]; K-N discounting
2. **f-RNNLM**: a non-class-based f-RNNLM with

512 hidden layer nodes

3. **Ngram+f-RNNLM**: the 6g and f-RNNLM are interpolated with a 50%-50% weighting in the lattice-based framework described in section 2

The initial lattices (Figure 1) were expanded and rescored using the Ngram, the f-RNNLM, and the interpolated Ngram+f-RNNLM LMs. System performance was evaluated using standard Precision, Recall, and F-score metrics. The full range of sub-component weightings was explored for the Ngram+f-RNNLM system (e.g., 40%-60%, 60%-40%), but the 50%-50% weighting gave the optimal performance (as determined by the three metrics). Consequently, the 50%-50% weighting was adopted for all the experiments reported in this paper. The metric scores were also used to determine the optimal DP value for the Dev data and Figure 2 shows the metric scores for the Ngram+f-RNNLM system. The inverse relationship between Precision and Recall/F-score is apparent, and a DP value of 0.5 achieves a desirable balance between these extremes. Similar patterns were obtained for all three systems, so the DP was set to 0.5 for all subsequent experiments.

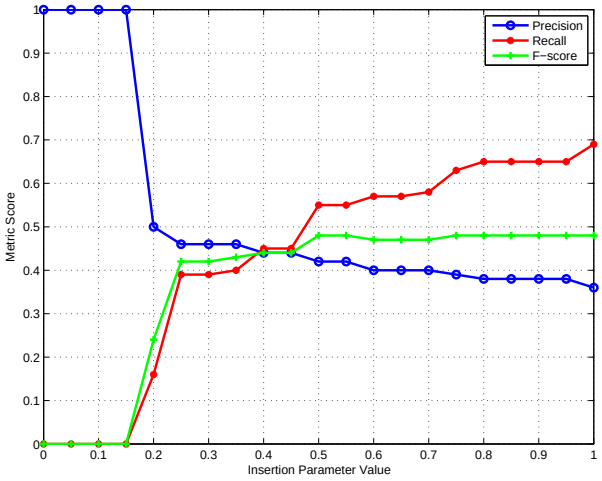

Figure 2: Precison, Recall, and F-score for Ngram+f-RNNLM for Different DP Values

	Precision (Dev/Test)		**Recall (Dev/Test)**		**F-score (Dev/Test)**	
Ngram	0.41	0.44	**0.55**	**0.60**	0.47	0.51
f-RNNLM	0.42	**0.47**	0.52	0.57	0.47	0.51
Ngram+f-RNNLM	**0.43**	**0.47**	**0.55**	0.59	**0.48**	**0.52**

Table 3: Dev and Test Sentence-level results for the Ngram, f-RNNLM, and Ngram+f-RNNLM systems

Table 3 shows that the Ngram+f-RNNLM system obtained the best (sometimes joint best) sentence-level Precision and Recall performance for every case except the Recall results for the Test set. Notably, the Ngram+f-RNNLM system obtained the best F-score results for both the Dev and Test sets. This suggests that the interpolated system combines the complementary properties of the two component LMs. Consequently, the Ngram+f-RNNLM system is comparatively more robust than either the Ngram or f-RNNLM systems, and the latter two are beneficially interpolated in the lattice-based framework.

	Dev #occs [%] (ref/hyp)		**Test #occs [%] (ref/hyp)**	
UH	3660 [2.29%]	6359 [3.97%]	3658 [2.44%]	6311 [4.10%]
UM	3331 [2.09%]	4711 [2.94%]	3392 [2.26%]	4201 [2.73%]
OH	2035 [1.27%]	3685 [2.30%]	2083 [1.39%]	3538 [2.30%]
AH	1053 [0.66%]	192 [0.12%]	348 [0.23%]	209 [0.14%]
UHUM	432 [0.27%]	73 [0.46%]	423 [0.28%]	92 [0.06%]
UHU	222 [0.14%]	8 [0.00%]	228 [0.15%]	23 [0.01%]
HM	61 [0.04%]	2 [0.00%]	55 [0.04%]	0 [0.00%]

Table 4: FP #occs for the Dev/Test reference files (ref) and the Ngram+f-RNNLM system output (hyp)

Table 4 gives the occurrence counts for both the scoring reference files and the Ngram+f-RNNLM system output hypotheses. These counts show that the Ngram+f-RNNLM models the various FP subtypes rather differently. There is a tendency to overgenerate the three most frequently occurring FPs (i.e., *UH*, *UM*, *OH*). The overgeneration ranges from 17.4% to 76.9%. By contrast, the system undergenerates the less frequently occurring subtypes (e.g., *AH*, *HM*). Presumably this is a consequence of the occurrence counts in the training data, which ensure that the LMs associate higher likelihoods with frequently occurring FPs. The patterns for all FP subtypes are similar for the Dev and Test sets. Table 5 further illuminates this by giving the Precision, Recall, and F-score scores for the distinct FP subtypes.

	Precision (Dev/Test)		**Recall (Dev/Test)**		**F-score (Dev/Test)**	
UH	0.42	0.46	0.72	0.74	0.53	0.57
UM	0.42	0.45	0.56	0.54	0.48	0.49
OH	0.48	0.53	0.70	0.70	0.57	0.60
UHUM	0.35	0.58	0.04	0.09	0.08	0.16
UHU	0.25	0.81	0.01	0.06	0.02	0.11
HM	0.50	0.00	0.02	0.00	0.04	0.00
AH	0.14	0.12	0.03	0.08	0.05	0.10

Table 5: Individual FP Results for the Ngram+f-RNNLM system

The scores in Table 5 show a fair amount of variation between the different FP subtypes. The scores for the three most frequently occurring FPs are relatively stable across the Dev and Test sets, achieving F-scores in the range 0.48-0.60. By contrast, the scores for the less common FPs sometimes fluctuate considerably (e.g., the Dev Precision for *HM* is 0.50, while the Test Precision is 0.00). Once again, this quantifies the impact of the data sparcity manifest in Table 2.

4. CONCLUSION

In recent years, interest in emotional or expressive speech synthesis has burgeoned. Dominant traits such as extraversion, conscientiousness, agreeableness, and openness are often considered to be essential to the creation of artificial personalities – and FPs are commonly occurring phenomena in natural conversational speech which convey important information about such traits. Consequently, this paper has described a novel approach to the task of inserting FPs into otherwise fluent token sequences to create disfluent input texts for speech synthesis systems.

A lattice-based rescoring framework has been presented which enables Ngram and f-RNNLM LMs to be interpolated. This framework enables multiple FPs to be inserted into multiple IPs. The experiments involving seven FPs show that, using standard metrics, the Ngram+f-RNNLM system is more robust than its constituent Ngram and f-RNNLM sub-components since it combines ther complementary tendencies.

Future research will focus on the modelling of other (more structurally complex) disfluency types, such as discourse markers, repetitions and restarts. It is also important to improve the way speech synthesis systems cope with disfluent input texts, and, to this end, data mixing, better outlier detection, and improved alignment methods will be explored.

5. ACKNOWLEDGEMENTS

This research was supported in part by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

6. REFERENCES

[1] Adell, J., Bonafonte, A., Escudero, D. 2007. Filled pauses in speech synthesis: Towards conversational speech. *Proc. 10th International Conference on Text, Speech and Dialogue* volume 1. Springer 358–365.

[2] Adell, J., Bonafonte, A., Escudero, D. 2007. Statistical Analysis of Filled Pauses Rhytm for Disfluent Speech Synthesis. *SSW6* Bonn, Germany. 223–227.

[3] Adell, J., Bonafonte, A., Escudero-Mancebo, D. 2010. Modelling Filled Pauses Prosody to Synthesise Disfluent Speech. *Speech Prosody* Chicago, USA.

[4] Aylett, M., Potard, B., Pidcock, C. 2013. Expressive Speech Synthesis: Synthesising Ambiguity. *In Proceedings of 8th ISCA Speech Synthesis Workshop*.

[5] Burkhardt, F., Campbell, N. 2014. Emotional Speech Synthesis. *The Oxford Handbook of Affective Computing*.

[6] Carletta, J. 2007. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation Journal* 41(2), 181–190.

[7] Chen, X., Wang, Y., Liu, X., Gales, M., Woodland, P. 2014. Efficient GPU-based Training of Neural Network Language Models Using Spliced Sentence Bunch. *In Proceedings of Interspeech* Singapore.

[8] Cieri, C., Miller, D., Walker, K. 2004. The Fisher Corpus: A Resource for the Next Generation of Speech-to-Text Fisher. *In Proceedings of LREC* Lisbon, Portugal.

[9] Clark, H. H., Tree, J. E. F. 2002. Using uh and um in spontaneous speech. *Cognition 84* 73–111.

[10] Collard, P., Corley, M., MacGregor, L. J., Donaldson, D. I. May 2008. Attention orienting effects of hesitations in speech: evidence from ERPs. *Journal of experimental psychology. Learning, memory, and cognition* 34(3), 696–702.

[11] Corley, M., Hartsuiker, R. J. 2003. Hesitation in speech can. . . um. . . help a listener understand. *Proceedings of the twenty-fifth meeting of the Cognitive Science Society* Boston, USA.

[12] Corley, M., Hartsuiker, R. J. Jan. 2011. Why um helps auditory word recognition: the temporal delay hypothesis. *PloS one* 6(5), e19792.

[13] Dall, R., Tomalin, M., Wester, M., Byrne, B., King, S. 2014. Investigating Automatic and Human Filled Pause Insertion for Speech Synthesis. *In Proceedings of Interspeech* Singapore.

[14] Dall, R., Wester, M., Corley, M. The Effect of Filled Pauses and Speaking Rate on Speech Comprehension in Natural, Vocoded and Synthetic Speech. *In Proceedings of Interspeech*.

[15] Fox Tree, J. E. 2001. Listeners' uses of um and uh in speech comprehension. *Memory and Cognition* 29(2), 320–326.

[16] Goodfrey, J. J., Holliman, E. C., McDaniel, J. 1992. SWITCHBOARD: Telephone speech corpus for research and development. *ICASSP* San Francisco, CA, USA. 517–520.

[17] He, L., Hyang, H., Lech, M. 2013. Emotional Speech Synthesis Based on Prosodic Feature Modification. *In Proceedings of the 8th International Conference on Bioinformatics and Biomedical Engineering*.

[18] Liu, X., Wang, Y., Chen, X., Gales, M., Woodland, P. 2014. Efficient Lattice Rescoring using Recurrent Neural Network Language Models. *In Proceedings of Interspeech* Singapore.

[19] Nass, C., Brave, S. 2005. Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship. The MIT Press.

[20] Schröder, M., Cowie, R., Cowie, E. 2001. Emotional Speech Synthesis: A Review. *In Proceedings of Eurospeech* volume 1 561–564.

[21] Stolke, A., Zheng, J., Wang, W., Abrash, V. 2011. SRILM at Sixteen: Update and Outlook. *In Proceedings of ASRU* Hawaii, USA.