The Simple4All entry to the Blizzard Challenge 2013

O. Watts[1], *A. Stan*[2], *Y. Mamiya*[1], *A. Suni*[3], *J.M. Burgos*[4], *J.M. Montero*[4]

[1]Centre for Speech Technology Research, University of Edinburgh, UK
[2]Communications Department, Technical University of Cluj-Napoca, Romania
[3]Institute of Behavioural Sciences, University of Helsinki, Finland
[4]Speech Technology Group, ETSIT, Universidad Politécnica de Madrid, Spain

owatts@inf.ed.ac.uk, adriana.stan@com.utcluj.ro, Antti.Suni@helsinki.fi,
jose.martin@die.upm.es, juancho@die.upm.es

Abstract

We describe the synthetic voices entered into the 2013 Blizzard Challenge by the SIMPLE4ALL consortium. The 2013 Blizzard Challenge presents an opportunity to test and benchmark some of the tools we have been developing to address two problems of interest: 1) how best to learn from plentiful 'found' data, and 2) how to produce systems in arbitrary new languages with minimal annotated data and language-specific expertise on the part of the system builders. We here explain how our tools were used to address these problems on the different tasks of the challenge, and provide some discussion of the evaluation results.

Index Terms: statistical parametric speech synthesis, speech alignment, speech segmentation, style diarisation, unsupervised learning, vector space model, audiobook data, glottal inverse filtering, glottal flow pulse library

1. Introduction

This paper describes the synthetic voices entered into the 2013 Blizzard Challenge by the SIMPLE4ALL consortium. SIMPLE4ALL is a European speech synthesis project focused on creating speech synthesis technology that learns from data with little or no expert supervision.[1] The 2013 Blizzard Challenge provides a good opportunity to test and benchmark some of the techniques we have been developing within the project. Two problems of central importance for SIMPLE4ALL are 1) how best to learn from plentiful 'found' data, and 2) how to produce systems in arbitrary new languages with minimal annotated data and language-specific expertise on the part of the system builders. We here explain how the different tasks of the challenge relate to the problems of interest, and give an overview of how we applied four parts of the SIMPLE4ALL toolkit to the tasks.

Obtaining and transcribing the speech data for training a corpus-based text-to-speech (TTS) system in a new language requires considerable time and expert knowledge. Typically, this speech data is collected during a specially-arranged recording session, for which a recording script has to be prepared, a suitable studio must be found, a voice talent must be recruited and speech recording must be carefully supervised. SIMPLE4ALL aims to ease the building of new voices by developing and distributing tools which allow the reuse of speech data produced for other purposes. A prime example of such 'found' data is freely available audiobook recordings which have been released into the public domain. In [1] we presented a part of our toolkit for segmenting and aligning such recordings, allowing us to circumvent the need to engineer purpose-recorded speech corpora where existing recordings are available. Task EH1 of the challenge lets us test tools addressing this problem, as it involves building a voice from a very large set of audiobook data which is provided as approximately 300 hours of chapter-sized mp3 files.

As well as obtaining a segmentation and alignment for audiobook data, it is also important to deal with the heterogeneity of such data. To this end, another part of the SIMPLE4ALL toolkit was used to provide diarisation of the automatically obtained corpora. If audio from radio broadcasts are to be used for training a TTS system, for example, it is crucial to diarise audio into speech and non-speech (e.g. music, applause, laughter). When pure speech has been obtained, it is further necessary to diarise it into separate speakers, and it may also be desirable to diarise a single speaker's speech into different emotions or speaking styles [2]. Ultimately the goal of the latter would be to build a synthesiser capable of producing speech in a variety of styles. A more short-term approach is to exclude more unusual speaking styles to produce a subset of relatively homogeneous and neutral speech. This gives a set of training data which is as much like a conventional TTS database as possible, but which doesn't incur the associated costs. This is the approach taken here.

A third part of the SIMPLE4ALL toolkit used for our Blizzard Challenge entry is designed to enable the construction of systems in languages where we have access to little or no linguistic expertise or expert-annotated data. We think it is valuable for speech technology to venture beyond the handful of the world's languages where resources such as text normalisers, lexicons and part-of-speech taggers already exist. Thus, part of the SIMPLE4ALL toolkit includes tools for constructing TTS front-ends which make as few implicit assumptions about the target language as possible, and which can be configured with minimal effort and expert knowledge to suit arbitrary new target languages. To this end, the modules rely on resources which are intended to be universal, such as the Unicode character database, and employ unsupervised learning so that unlabelled text resources can be exploited without the need for costly annotation. Task IH1 lets us test tools addressing this problem, as it involves building voices for four Indian languages (Hindi, Bengali, Kannada and Tamil) for which the consortium members have no language-specific expertise or resources.

The fourth and final part of the SIMPLE4ALL toolkit used for our Blizzard Challenge entry is an implementation of new

[1] www.simple4all.org/

speech signal models capable of modelling a large variety of speaking styles and vocal emotions [3].

We note that an initial public version of tools for this whole pipeline of tools is due to be released in November 2013.

2. System Description

2.1. Data preparation

As already mentioned, the training data for task EH1 of the challenge was provided without a sentence-level speech segmentation and text alignment. Therefore one of the sub-tasks was to obtain the correct alignment, prior to building the synthetic voices. Our previous work on automatic alignment of speech with imperfect transcripts [4, 5, 6] has developed tools to perform the alignment without the use of high-level language expertise or existing acoustic models. The method involves two major steps: *1) a sentence-level segmentation of the speech data*, and *2) automatic alignment of speech and text at sentence-level*. Both steps are lightly supervised and require only a minimum amount of manually labelled data, also called *initial training data*. The following paragraphs describe them in more detail.

Step 1. Speech segmentation is performed using a 16 Gaussian Mixture Model (GMM)-based voice activity detection algorithm [6]. Two GMMs are trained, one for silence and one for speech, from 10 minutes of manually-labelled data, in which the inter-sentence silences are marked. Feature vectors consist of energy, 12 dimensional MFCCs, their deltas and the number of zero crossings. After training the GMMs, for each frame within the manually-labelled data, we compute the the log likelihood ratio, followed by a median filter smoothing. This process also detects short intra-sentence silences. In order to discriminate between inter- and intra- sentence silence frames, two Gaussian probability distribution functions are fitted onto the histogram of silence durations. Their intersection represents the threshold for sentence boundary silence duration. The GMMs are then run on the entire speech resource. Results showed over 96% accuracy in sentence boundary detection.

Step 2. The speech alignment step starts from the same 10 minutes of initial training data, which is now segmented and needs to be orthographically transcribed. A first set of poor initial grapheme-level acoustic models is built from it. The models are then used to recognise the entire speech resource with the help of a highly restricted word network built from the full text transcript (see [4] for more details). To determine the correctly recognised utterances, the recognition is run over the speech data with various degrees of freedom within the word networks, and the obtained acoustic scores are compared. Confident data is then used to re-train the acoustic models, and the process repeats. A final step in the alignment is the re-estimation of the acoustic models using tri-graphemes, and this increases the aligned data by over 40% relative. However, for short speech resources, this step might be unfavourable, as the number of tri-graphemes is too large to obtain satisfactory statistics for them. Previous results obtained with an English audiobook showed an average 75% confident data with a 7% SER and 0.5% WER [5].

For the Blizzard Challenge task EH1 , each audiobook was segmented and aligned individually, aligned percentages being similar to our previous results.

2.2. Data selection

The speaker diarization system described in [7] was used to cluster the segmented utterances obtained as described in section 2.1 for a single audiobook. As we are clustering the speech of a single speaker, the result is a set of 'pseudo-speakers', each corresponding to some automatically detected speaking style as in [2]. A difference in the current case is that we seek only a single cluster of neutral style speech to use, and discard the other clusters. 12 such clusters were produced by an iterative process of speaker segmentation and agglomerative clustering of segments. For each sentence, the system output the dominant 'speaker' of the sentence and the purity of the sentence (fraction of the sentence spoken by the dominant speaker). A single cluster accounted for 90% of the sentences processed – informal listening suggested that this corresponded well with the speaker's neutral style of reading. Taking only the completely pure utterances reduced this to 89%.

For the EH1 voice acoustic models, a 5 hour subset of this pure neutral data was selected. Note however that the whole of the data for which a confident alignment was obtained (section 2.1) was used for the pause prediction model (see section 2.4).

2.3. Text processing

The tools used for building TTS front-ends for entries to all parts of the challenge are based on ideas outlined in [8], applied to Spanish TTS in [9], and to 14 different languages in [10]. We summarise the tools here, drawing heavily on descriptions given in those previous publications.

Input to the system consists of the audio of utterances together with their text transcription. For the EH1 voice, these utterances made up 5 hours of the neutral speech extracted as described in Section 2.2. For each of the Indian languages of task IH1, 950 of the available 1000 sentences and their plain orthography UTF-8 transcriptions were used as input; 50 sentences were set aside for use as an internal development set.

As well as the training speech data and its transcripts, our tools exploit the large amount of unannotated text data which is available for many languages on the web. For the task IH1 voices, this consisted of approximately 13.4, 2.2, 4.4 and 6.4 million tokens of text for Hindi, Bengali, Kannada and Tamil, respectively, which we obtained from Wikipedia. For the English voice for Task EH1, we used only the transcripts of the full audiobook training corpus only as we wanted to experiment with using only in-domain data. For all languages, these unannotated text data were used for construction of the word- and letter-representations described below.

Text which is input to the system is assumed to be UTF-8 encoded: given UTF-8 text, text processing is fully automatic and makes use of a theoretically universal resource: the Unicode database. Unicode character properties are used to tokenise the text and characterise tokens as words, whitespace, punctuation etc. Our front-ends currently expect text without abbreviations, numerals, and symbols (e.g. for currency) which require expansion; however, the lightly supervised learning of modules to expand such non-standard words is an active topic of research [11], and we hope to integrate such modules into our toolkit in the near future.

A letter-based approach is used, in which the names of letters are used directly as the names of speech modelling units (in place of the phonemes of a conventional front-end). This has given good results for languages with transparent alphabetic orthographies such as Romanian, Spanish and Finnish, and can give acceptable results even for languages with less transparent orthographies, such as English [8, 12, 13, 14]. We decided to submit letter-based systems for both the EH1 and IH1 tasks, even though high-quality lexicons are available for English. Al-

though the complicated letter-to-sound relations of English orthography mean that we expect this to severely degrade synthesis quality, we wished to make use of the opportunity presented by the Blizzard Challenge to evaluate this naive approach using many listeners against state-of-the-art systems. In this way, we have a useful benchmark against which to compare the results of ongoing attempts to tackle the same problem in a less naive way.

The induced front-ends make use of no expert-specified categories of letter and word, such as phonetic categories (vowel, nasal, approximant, etc.) and part of speech categories (noun, verb, adjective, etc.). Instead, features that are designed to stand in for such expert knowledge but which are derived fully automatically from the distributional analysis of unannotated text (speech transcriptions and Wikipedia text) are used. The distributional analysis is conducted via vector space models (VSMs); the VSM was originally applied to the characterisation of documents for purposes of Information Retrieval. VSMs are applied to TTS in [8], where models are built at various levels of analysis (letter, word and utterance) from large bodies of unlabelled text. To build these models, co-occurrence statistics are gathered in matrix form to produce high-dimensional representations of the distributional behaviour of e.g. word and letter types in the corpus. Lower-dimensional representations are obtained by approximately factorising the matrix of raw co-occurrence counts by the application of slim singular value decomposition. This distributional analysis places textual objects in a continuous-valued space, which is then partitioned by decision tree questions during the training of TTS system components such as acoustic models for synthesis or decision trees for pause prediction. For the present voices, a VSM of letters was constructed by producing a matrix of counts of immediate left and right co-occurrences of each letter type, and from this matrix a 5-dimensional space was produced to characterise letters. Token co-occurrence was counted with the nearest left and right neighbour tokens (excluding whitespace tokens); co-occurrence was counted with the most frequent 250 tokens in the corpus. A 20-dimensional space was produced to characterise word tokens.

2.4. Pause Prediction

Phrase-break prediction is an essential part in text-to-speech synthesis because it determines the rhythm, as well as prominence in the output synthetic speech. As previously stated, our system tries to avoid supervised and language-dependent modules. Hence, our phrase-break prediction step is also lightly supervised, and we treat silences detected from the acoustics as surrogate phrase-breaks. We exploit the large amount of speech data made available for task EH1, and extract a training set from the forced alignment of the audio and its corresponding orthographic transcripts obtained in the alignment step (see Section 2.1). (The same approach was used for the IH1 voices, except in those cases the training corpus was much smaller and a sentence segmentation was already available.) To discriminate between the short inter-word pauses and pauses which might signal actual phrase-breaks, we plotted the histogram of all the silence segments within the available data. This lead to a separation threshold of 200 ms. Silences below this threshold were discarded and added to the no-pause (NP) set. A list of all the consecutive pairs of words from the text and the length, and existence of a phrase break constitutes our training data. This method works under the assumption that the test data will be part of the same domain as the training one (i.e. audiobooks), and the phrase break durations would be similar, which also means that the method is corpus-dependent.

But, as the surface form of the words does not inherently contain enough information to predict the phrase breaks, we rely on the vector representations of words mentioned in section 2.3. The vectors for each pair of consecutive words from the training data, along with their pause indicator constitute the input for a classification and regression tree. Results showed an overall 0.9 F-measure, but only an 0.4 F-measure for *pause* instances (P). This is mostly due to the unbalanced training data set (i.e. there are more NP word pairs than P). Even when the set was artificially built from equal amounts of and NP pairs, the results remained similar. This might be caused by the VSMs not being able to capture the essential features required for the pause prediction, and hence a more elaborate set of features would be beneficial in future work.

Punctuation is also an important pause indicator, and so we included the punctuation marks as word-pair constituents. This lead to an increase of 0.1 in the F-measure of the P class. Still the results are below expectation, but we estimate that they are caused by the poor alignment of speech with its orthographic transcripts, especially for English which is known to have a high letter-to-sound complexity.

To estimate the phrase breaks in the testing data, we converted the sentences into word pairs, extracted their corresponding vectors and predicted the P/NP class with the previously trained CART.

2.5. Acoustic Modelling

As mentioned previously, a five-hour subset of the available training corpus for EH1 was used to train acoustic models. The inconsistent recording conditions and small amounts of training data for the IH1 tasks meant that extra robustness for acoustic parameterization and training was required. The 4 IH1 voices were each built in an identical fashion, except that half of the Bengali training data was discarded due to being recorded in excessively reverberant conditions. Various other inconsistencies were present too. Style-adaptive training and the use of extra contextual labels were considered for distinguishing these different recording conditions, but our tools for unsupervised recording quality classification are not yet ready.

2.5.1. Parameterisation

For the EH1 voice, the training data were parameterised using STRAIGHT, almost as described in [15]. The only difference is that instead of the committee of different pitch-trackers used in the earlier work, pitch tracks obtained with GlottHMM (using a glottal source signal estimated by glottal inverse filtering [3]) were used for their greater accuracy.

For the IH1 voices, full GlottHMM parameterisation [3] was used after initial denoising of the training speech. 24 vocal tract LSF coefficients and 10 voice source LSF coefficients were extracted as well as harmonic-to-noise ratio with 5 bands, energy and F0. Pulse libraries [16] were extracted from 10 utterances for each voice.

Some alterations to the parameterization scheme described in [3] were made to increase robustness. First, the iterative adaptive inverse filtering method was replaced with direct inverse filtering using a pre-emphasis filter only. Second, the pre-emphasis filter was added to unvoiced analysis, to ensure continuous LSF trajectories across voicing boundaries, thus reducing the audible distortion of voicing errors.

Notably, we did not use the vocal tract LSF parameters directly in the training, but instead converted the parameters to mel-cepstral representation via LPC spectrum. As mel-cepstral coefficients are decorrelated, focus on perceptually relevant frequencies and provide smooth trajectories, they might be more suitable than LSFs for HMM training, especially on difficult material such as the current challenge. Further investigation on this topic would be needed to verify this.

2.5.2. Training and synthesis

A rich set of contexts was created using the results of the analysis described in section 2.3 for each letter token in the training data for all languages. Features used include the identity of the letter and the identities of its neighbours within a window of given length. A 5-letter window was used for the IH1 voices, and a 9-letter window for the EH1 voice. Some informal experiments suggested this to be an appropriate size for the 5 hour subset of the EH1 data we used. Additional features were the VSM values of each letter in the window, and the distance from and until a word boundary, pause, and utterance boundary.

For the EH1 voice, speaker-dependent acoustic models were built from the parameterised speech data and labelling using the speaker-dependent model-building recipe described in [17].

For the IH1 voices, the HMM models were trained with the standard HTS 2.0 [18] recipe, modified for additional GlottHMM streams, but using three iterations of decision tree clustering instead of two. MGE training was also applied. Parameter generation was performed considering global variance, with stream-dependent thresholds. Generated mel-cepstral coefficients were converted back to the LSF form for stability checking and vocoding purposes. Excitation was generated using the PCA-mean pulse approach [19].

Informal listening by the authors and feedback from several native speakers suggested that the denoised GlottHMM version performed better than previous SIMPLE^{4}ALL voices built on the same data using the STRAIGHT vocoder, but detailed analysis of the exact reasons for this improvement remains to be done.

3. Results

The identifier for our system in the published results is P.

On Task EH1 ours was consistently the worst-performing system of all entries. On the intelligibility sections of the evaluation, there was a c.10% gap in WERs between our system and the second worst performing one. This gap was higher among the paid subset of listeners, and lower among online volunteers and speech expert listeners, where it dropped to c.5–6%.

Performance relative to the other systems in the IH1 tasks was much better. For the speaker similarity and naturalness sections of the evaluation for all 4 languages, our system tends to score somewhere in the middle of all TTS systems. The intelligibility results published for Hindi and Kannada follow a similar pattern. In the Hindi test, 4 TTS systems achieved lower WERs than ours, 1 was worse, and 1 scored within 1% WER of our system; in the paid listener subset, our system achieves precisely the middle rank in the Hindi intelligibility results. In both listener group sections of the Kannada intelligibility test, our system also achieves precisely the middle rank.

4. Conclusions

The poor performance of our system in EH1 was anticipated due to the difficulty of TTS from the surface orthographic forms of English words, and to the high level of expertise that has been accumulated for doing TTS in English where there is no self-imposed limit on the amount of target-language expertise that can be used in a system. However, we wished to know exactly how much the lack of a lexicon would set us back in an extensive evaluation with many listeners. Furthermore, these results are envisaged as being useful for on-going improvements to our system, where light supervision and unsupervised lexicon induction techniques are exploited. Because Blizzard stimuli are released after the challenge, it is possible to evaluate improved systems by re-running the evaluation locally on a smaller scale, using a subset of 'landmark' systems from the challenge which allow new results for improved systems to be placed among existing Blizzard results. Having our own baseline among the original results is useful for sanity-checking when projecting results for new systems into the space of existing results.

We regard the middling performance of our system on the IH1 tasks as a success, given that the system makes no use of expert script knowledge, while we assume that other systems probably all make use of at least the phonetic annotation distributed for the challenge. This is the first formal evaluation of our letter-based front-end as applied to a non-alphabetic script: we regard its reasonable performance on the four alphasyllabic scripts of IH1 as a validation for the unsupervised approach for our main target domain of under-resourced languages.

5. References

[1] A. Stan, O. Watts, Y. Mamiya, M. Giurgiu, R. A. J. Clark, J. Yamagishi, and S. King, "TUNDRA: A Multilingual Corpus of Found Data for TTS Research Created with Light Supervision," in *Proc. of Interspeech (accepted)*, 2013.

[2] J. Lorenzo, B. Martinez, R. Barra-Chicote, V. LopezLudena, J. Ferreiros, J. Yamagishi, and J. Montero, "Towards an unsupervised speaking style voice building framework: Multistyle speaker diarization."

[3] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 153–165, 2011.

[4] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. IEEE Workshop on Spoken Language Technology, Miami, Florida, USA*, 2012.

[5] A. Stan, P. Bell, J. Yamagishi, and S. King, "Lightly Supervised Discriminative Training of Grapheme Models for Improved Sentence-level Alignment of Speech and Text Data," in *Proc. of Interspeech (accepted)*, 2013.

[6] Y. Mamiya, J. Yamagishi, O. Watts, R. A. Clark, S. King, and A. Stan, "Lightly Supervised GMM VAD to use Audiobook for Speech Synthesiser," in *Proc. ICASSP*, 2013.

[7] J. Pardo, R. Barra-Chicote, R. San-Segundo, R. de Cordoba, and B. Martinez-Gonzalez, "Speaker Diarization Features: The UPM Contribution to the RT09 Evaluation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 426–435, 2012.

[8] O. Watts, "Unsupervised learning for text-to-speech synthesis," Ph.D. dissertation, University of Edinburgh, 2012.

[9] J. Lorenzo-Trueba, O. Watts, R. Barra-Chicote, J. Yamagishi, S. King, and J. M. Montero, "Simple4All proposals for the Albayzin Evaluations in Speech Synthesis," in *Proc. Iberspeech*, 2012.

[10] O. Watts, A. Stan, R. Clark, Y. Mamiya, M. Giurgiu, J. Yamagishi, and S. King, "Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis," in *Proc. of 8th ISCA Workshop on Speech Synthesis*, 2013.

[11] R. San-Segundo, J. M. Montero, V. Lopez-Ludeña, and S. King, "Detecting acronyms from capital letter sequences in Spanish," in *Proc. Interspeech*, Portland, Oregon, USA, Sep. 2012.

[12] A. Black and A. Font Llitjos, "Unit selection without a phoneme set," in *IEEE TTS Workshop 2002*, 2002.

[13] G. Anumanchipalli, K. Prahallad, and A. Black, "Significance of early tagged contextual graphemes in grapheme based speech synthesis and recognition systems," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-April 4 2008, pp. 4645–4648.

[14] M. P. Aylett, S. King, and J. Yamagishi, "Speech synthesis without a phone inventory," in *Proc. of Interspeech*, 2009, pp. 2087–2090.

[15] J. Yamagishi and O. Watts, "The CSTR/EMIME HTS System for Blizzard Challenge," in *Proc. Blizzard Challenge 2010*, Sep. 2010.

[16] T. Raitio, A. S. H. Pulakka, M. Vainio, and P. Alku, "Utilizing glottal source pulse library for generating improved excitation signal for HMM-based speech synthesis," in *Proc. ICASSP*, 2011.

[17] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[18] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of 7th ISCA Workshop on Speech Synthesis*, 2007, pp. 294–299.

[19] T. Raitio, A. Suni, M. Vainio, and P. Alku, "Comparing glottal-flow-excited statistical parametric speech synthesis methods," in *Proc. ICASSP*, 2013.