Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis

O. Watts[1], *A. Stan*[2], *R. Clark*[1], *Y. Mamiya*[1], *M. Giurgiu*[2], *J. Yamagishi*[13], *S. King*[1]

[1]The Centre for Speech Technology Research, University of Edinburgh, UK
[2]Communications Department, Technical University of Cluj-Napoca, Romania
[3]National Institute of Informatics, Japan

{adriana.stan, mircea.giurgiu}@com.utcluj.ro, Simon.King@ed.ac.uk,
{owatts, Yoshitaka.Mamiya, robert, jyamagis}@inf.ed.ac.uk

Abstract

This paper presents techniques for building text-to-speech front-ends in a way that avoids the need for language-specific expert knowledge, but instead relies on universal resources (such as the Unicode character database) and unsupervised learning from unannotated data to ease system development. The acquisition of expert language-specific knowledge and expert annotated data is a major bottleneck in the development of corpus-based TTS systems in new languages. The methods presented here side-step the need for such resources as pronunciation lexicons, phonetic feature sets, part of speech tagged data, etc. The paper explains how the techniques introduced are applied to the 14 languages of a corpus of 'found' audiobook data. Results of an evaluation of the intelligibility of the systems resulting from applying these novel techniques to this data are presented.
Index Terms: multilingual speech synthesis, unsupervised learning, vector space model, text-to-speech, audiobook data

1. Introduction

Collecting and annotating the data necessary for training a corpus-based text-to-speech (TTS) conversion system in a new language requires considerable time and expert knowledge. Conventionally, audio data for training a synthesiser *back-end* (or waveform generator) will be gathered during a specially-arranged recording session. For this, a recording script must be prepared, a suitable studio must be found, a voice talent must be recruited and speech recording must be carefully supervised. One of the primary goals of the *Simple4All*[1] project is to reduce the time and expert knowledge needed to produce new TTS systems. In [1] we presented a toolkit – developed as part of this project – for segmenting and aligning existing freely-available recordings (audiobooks), circumventing to some extent the need to engineer purpose-recorded speech corpora. The outcome of applying those tools to audiobooks in 14 languages is what we have released under the name of the *Tundra corpus*.

However, the problems associated with TTS data-collection do not stop when we have obtained transcribed speech data for training a synthesiser back-end. TTS systems also require a *front-end* (or text analysis module), which accepts input text and outputs a representation of an utterance suitable for input into the back-end. TTS systems generally represent utterances in terms of units and features based on linguistic knowledge, such as phonemes, syllables, lexical stress, phrase boundaries etc. The components of the front-end that predict these from input text are either made up of hand-written rules or statistical modules; acquiring the expert knowledge required either to manually specify those rules, or to annotate a learning sample on which to train the statistical models, represents a major obstacle to creating a TTS system for a new target language and requires highly specialised knowledge. Such non-trivial tasks include, for example, specifying a phoneme-set or part of speech (POS) tag-set for a language where one has not already been defined; annotating plain text with POS tags, as required to train a POS tagger and annotating the surface forms of words with phonemes to build a pronunciation lexicon.

The toolkit we are developing in *Simple4All* 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. The current paper presents these tools and explains how they were applied to the data of the Tundra corpus to produce TTS systems in 14 languages. We present the results of a listening test of the intelligibility of those systems, and thus evaluate the entire pipeline implemented by our toolkit, which begins with raw found data and ends with trained TTS systems. An initial public version of tools for this whole pipeline (for segmenting and aligning found data and for producing TTS systems with minimal expert knowledge) is due to be released in November 2013.

In prior work addressing the bottleneck in TTS system construction represented by the front-end, unified systems aimed at producing complete systems have generally taken the strategy of providing infrastructure to ease the collection by non-experts of the conventional resources necessary for system construction. This infrastructure might take the form of user-friendly development environments [2], or training and on-going support [3]. Prior work has also presented unsupervised methods for building systems based on letters rather than phonemes [4, 5], induction of phone-sets [6, 7], syllable-like units [8, 9], or lexicons [10]. However, this work has not been presented as an integrated framework for producing end-to-end TTS systems. Furthermore, despite the significant work on unsupervised learning in Natural Language Processing [11, 12] and Information Retrieval [13, 14], potentially useful techniques developed in those fields have not been applied to the problem of TTS front-end induction.

[1] www.simple4all.org/

2. Database

The Tundra corpus [1] is a standardised multilingual corpus designed for text-to-speech research with imperfect or found data. It consists of 14 audiobooks in 14 different languages (Bulgarian, Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Polish, Portuguese, Romanian, Russian and Spanish) and amounts to approximately 60 hours of speech. A complete list of the audiobooks with their sources and durations can be found here `http://tundra.simple4all.org`.

The corpus provides utterance-level alignments obtained with a lightly supervised process described in [15] and [16]. The accuracy of the alignment method, as described in [16] is of 7% SER and 0.8% WER, therefore some light post-processing is required in order to eliminate some of the erroneous utterances. Initial segmentation of the audiobooks into utterance-size chunks was performed using the lightly supervised GMM-based VAD described in [17]. As most of the used audiobooks are recorded in non-specialised environments, the speech data underwent a light cleaning process: normalising the DC offset, applying a multi-band noise gate removal and an RMS-based deverberation method, as described in [1].

3. System Construction

For each of the 14 languages of the Tundra corpus, a TTS system was trained with no reliance of language-specific expertise. Although speaker and recording differences mean that meaningful comparison between languages is difficult, we wished to make the training conditions for the 14 voices as uniform as possible. Therefore, we selected a 1 hour subset of each of the languages' data on which to train voices for this evaluation: the method of data selection we used is explained in Section 3.1. Then text analysis and waveform generation components were trained on that selected data as explained in Sections 3.2 and 3.3, respectively.

3.1. Lightly-supervised data selection

Our principal current interest in audiobook data is that it presents a source of 'found' data from which TTS training databases can be harvested without the need to construct a recording script, recruit a native speaker of the target language, and supervise the recording of a script from scratch. In the present work, therefore, we ignore the other possible advantage of using audiobook data: that harnessing the variety of speaking styles present in audiobooks might enable us to produce less 'mechanical'-sounding TTS systems. Although this is a longer-term goal, we here follow an approach similar to the one presented in [18], which aims to select a neutral subset of a database containing *diverse* speech. In that paper, 9 utterance-level acoustic features are used along with several textual cues to exclude diverse speech from the training set. Thresholds over these features are set manually by the system builder to exclude non-neutral utterances.

For the current work we perform utterance selection using an active learning approach, with uncertainty sampling [19]. Rather than being required to tune thresholds manually, the system builder is presented with example utterances and asked to indicate whether or not they are spoken in a neutral style. The interface therefore insulates the user from the details of the features used, and lets the user focus on what should be key: their intuitive response to hearing speech samples. The procedure we used is as follows:

1) **Feature extraction** First, frame-level features (F_0, energy and spectral tilt – approximated by 1st mel cepstral coefficient) are obtained, from which utterance-level features are computed. The fact that no thresholds need to be manually tuned means that we can afford to use a great many more features than the 9 employed in [18]. Our feature set is based on the one described in [20]: we compute mean, standard deviation, range, slope, minimum and maximum (6-level factor) for $F0$, spectral tilt, and energy (3-level factor) in the following sub-segments of each utterance: entire utterance, 1st and 2nd halfs, all 4 quarters, first and last 100ms, first and last 200ms (11-level factor), giving a total of 198 features.

2) **Initial labelling** The user is presented with the audio of s randomly-selected *seed utterances* from the whole corpus (via a text-based user interface) and asked to label them *keep* or *discard* – utterances are labelled with the user's decision.

3) **Classifier training** A classifier is trained on the labelled examples. Our choice of classifier is a bagged ensemble of decision trees [21] because it can be trained quickly (allowing online active learning in real time), is robust against noisy features and able to accept unnormalised input variables, and mixtures of discrete and continuous input variables (allowing a great many different acoustic features to be used, and different types of features), allows the space of utterances to be partitioned recursively (enabling complex interactions between features to be detected), and provides robust estimates of class probabilities (important for step 4).

4) **Uncertainty sampling** The set of u uncertain examples (utterances about which the classifier is most uncertain – in the present case, the utterances which have closest to 0.5 *keep* probability). The utterances in this set are presented to the user for labelling.

5) Steps 3 and 4 are repeated as many times as time allows.

6) The set of utterances either labelled *keep* by the user are kept for training, as well as enough of the utterances to which the trained classifier gives the highest *keep* probability to, to make up the desired quantity of training data.

For the work presented here, s was set to 15 and u was set to 1. That is, the user was asked to provide 15 labels at the outset, and presented with a single uncertain example at each iteration. The stopping criterion we used in this work was to limit the number of iterations to 15 – in the present, utterance selection time was limited to approximately 20 minutes per language, and 15 was found to be a reasonable number of iterations in that time. Informal comparison suggested the approach outlined is beneficial for this task, but in ongoing work we are testing this rigorously and comparing uncertainty sampling with random sampling, as well as applying our active learning tool to other TTS tasks.

3.2. Front-end construction with unsupervised learning

The TTS front-end building tools used for this work are based on ideas outlined in [22] and applied to Spanish TTS in [23]. Input to the system consists of the audio of utterances selected as described in Section 3.1, together with their text transcription (aligned at the utterance level): in the present case, these are taken from the Tundra corpus, and had been obtained as summarised in Section 2. As an additional input, 5 million words of running text data were obtained from Wikipedia in the target languages 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 Uni-

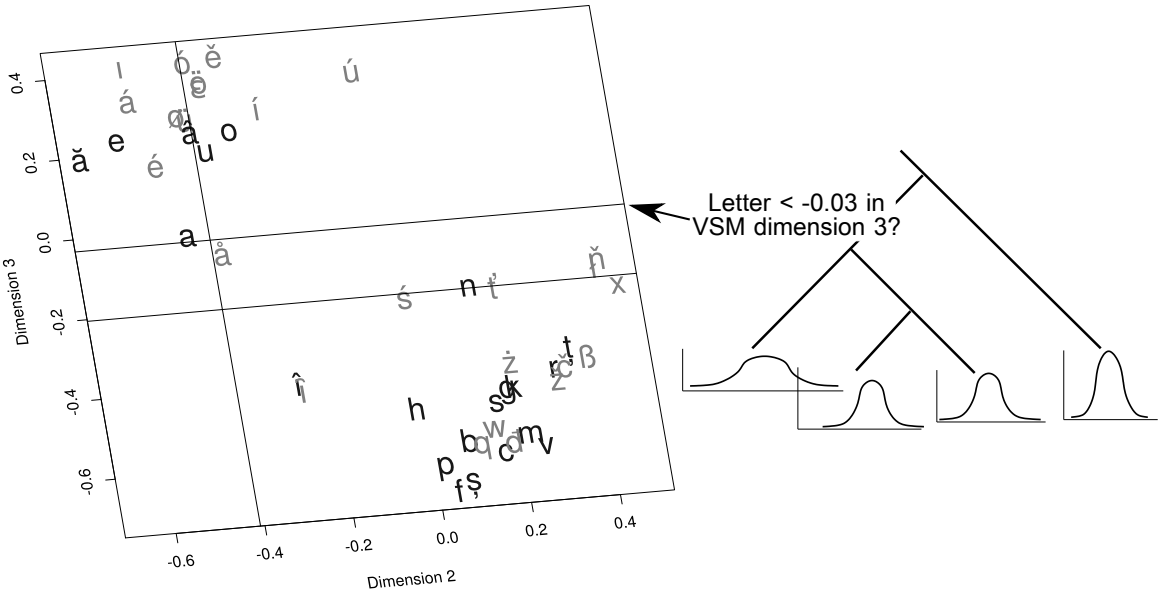

Figure 1: Use of a letter space to replace phonetic knowledge in decision-tree based state-tying. Shown here are 2 dimensions of the actual letter space induced in training the Romanian system described in the paper. The 3 lines bisecting the space represent the 3 questions actually asked in the uppermost fragment (first three 'generations') of the state-tying decision tree for the central state of the model for spectral envelope features. Letters shown in black are 'heard' by the system (i.e. are present in the transcriptions of the audio training data) but ones shown in grey are only 'seen' (i.e. appear only in textual training data) and are mainly foreign language letters.

code database. Unicode character properties are used to tokenise the text and characterise tokens as words, whitespace, punctuation etc. Our modules have so far been successfully applied to a variety of alphabetic (Latin-based, Cyrillic) and alphasyllabic (Brahmic) scripts. 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 [24], 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 [22, 4, 5, 7].

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 [22], 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 10-dimensional space was produced to characterise tokens.

Two dimensions of the letter space induced in training the Romanian system are shown in Figure 1. It can be seen that in these dimensions of the space, vowel and consonant symbols are clearly separable. When a decision tree for clustering acoustic model states is built and allowed to query items' positions in these 2 dimensions, it can use all partitions of the space orthogonal to its axes. A decision tree question such as *Is the letter's value in VSM dimension 3 < -0.03?* is very nearly equivalent to a question based on linguistic knowledge such as *Is the letter a consonant?* The categories of vowel and consonant are useful for clustering acoustic models, and so decision trees actually built using this space use such partitions of the space: the 3 lines shown bisecting the space in the figure represent the 3 questions actually asked in the uppermost fragment (first three 'generations') of the state-tying decision tree for the central state of the model for spectral envelope features.

Distributional analysis places linguistic or textual units in a continuous space which is then partitioned on acoustic evidence. The space constrains the possible groupings of objects that can be considered during decision tree growing. Distributional analysis also allows splits made to generalise to items

that are 'seen' by the system in text data but not 'heard' in the audio data. This is most obviously useful where units such as words are concerned, where many items not present in the training speech corpus are likely to occur at run-time. It can, however, also be useful where letters are concerned, and some examples that illustrate our models' ability to generalise beyond what is heard can be seen in the letter space shown in Figure 1. There, letters shown in black are 'heard' by the system but ones shown in grey are only 'seen' – these are mainly due to foreign language words within Romanian Wikipedia entries. It can be seen that unheard foreign vowels such as *á* and *ö* are suitably placed near the Romanian vowels, and unheard consonants such as *ß* and *q* are placed near the consonants that are actually heard. Splits such as those shown – made only on the basis of the heard items – therefore generalise to unheard items. In the case of letters, this allows rare and foreign letters to be handled despite their absence in the transcriptions of acoustic training data. It can also allow better handling of non-standard spellings: in the case of the vowel *î* (*i* with circumflex), there is a variant (with inverted breve instead of circumflex) which is not present in any of the speech transcriptions but which is used in a few Wikipedia articles. From Figure 1 it can be seen that almost identical representations are learned for these two letters, meaning a decision tree built using those representations will be able to handle the variant form correctly at run-time, even though no instances of that variant were seen in the transcription of the speech training corpus.

The front ends make use of decision trees to predict pauses at the junctures between words. Data for training these trees are acquired automatically by force-aligning the training data with their transcriptions, and allowing the optional insertion of silence between words. The independent variables used by the trees are whether words are separated by punctuation or space, and the VSM features of the tokens preceding and following the juncture.

A rich set of contexts is created using the results of the analysis described here for each letter token in the database. Features include the identity of the letter and the identities of its neighbours (within a 5-letter window), the VSM values of each of those letters, and the distance from and until a word boundary, pause, and utterance boundary. In the current systems, word VSM features are not included directly in the letter contexts, but are used by the decision tree for predicting pauses at runtime.

3.3. Back-end construction

For training the waveform generation modules for the 14 voices, the waveforms of the training corpora were parameterised almost as described in [25]. The one difference is that instead of the committee of different pitch-trackers used in the earlier work, pitch tracks obtained from a glottal source signal estimated by glottal inverse filtering [26] were used for their greater accuracy.

For all systems, speaker-dependent acoustic models were built from this parameterised speech data and the annotation described in Section 3.2, using the speaker-dependent model-building recipe described in [27].

Static and interactive demos of the resulting voices are available at `http://tundra.simple4all.org/demo`. A screen shot of the geographically-organised demo page is shown in Figure 2.

Figure 2: Demo screenshot: this geographical interface to voices can be found at `http://tundra.simple4all.org/demo`.

4. System Evaluation

4.1. Procedure

We are primarily interested in having our systems produce *intelligible* speech; evaluation therefore focused on the intelligibility of TTS output as measured by the word and letter error rates of listeners' transcriptions of those outputs. Conventionally in TTS evaluation, listeners are asked to transcribe semantically unpredictable sentences (SUS) [28]. However, such SUS are not currently available in all the Tundra languages and it is not trivial to construct new SUS, and so we resorted to using short natural sentences from the held-out test sets of the Tundra corpus.

For all 14 Tundra languages, 40 sentences were manually segmented from the held-out chapters of the relevant audiobook. Note that these test sets are distributed with the Tundra corpus, and so the results presented below can be considered benchmarks for future work. An attempt was made to select sentences of 6–8 words in order to make the inherent difficulty of transcription as uniform as possible. However, in some languages these thresholds had to be relaxed; Table 1 gives statistics of test-sentence lengths in all languages.

Subjects for the evaluation were recruited through a web-based crowdsourcing service. The advert for the evaluation specified that native speakers of the relevant language were required; in addition, participation in each part of the evaluation was restricted to users registered in countries where the relevant language is an official or majority language. We attempted to recruit listeners to evaluate all 14 systems built. However, as the option to restrict participation to workers registered in Denmark, Finland and Hungary was not available in the service we used, listening test for only 11 of the systems were publicised. The number of responses from participants varied greatly between languages. At the time of writing, responses from a sufficient number of listeners (25+) had been collected in only 5 of the languages (Bulgarian, English, Italian, Polish and Romanian) . Results for these five languages are presented here; evaluation of the remaining voices is left for future work.

In all languages, two conditions were evaluated: the natural speech of the natural sentences from the test set, and the

Table 1: Statistics of Tundra test-sentence lengths (number of words)

Language	Mean	Standard deviation
German	6.63	0.87
Finnish	6.8	0.91
Bulgarian	6.85	0.83
English	6.88	0.94
Italian	6.9	0.87
Polish	6.95	0.88
Hungarian	7.05	0.81
Russian	7.13	1.18
Danish	7.4	1.19
Portuguese	8.08	1.47
Dutch	8.1	2.15
Romanian	8.55	1.97
French	8.58	1.96
Spanish	8.8	1.65

TTS system reading the same text. In the four languages of the Simple[4]All consortium members (including two of the languages for which results are presented here: Romanian and English), however, SUS were available, and so for those languages a third condition was evaluated: the TTS system producing SUS texts. This is designed to provide a way of broadly gauging the relative difficulty of transcribing natural and SUS sentences, although language and text differences mean it is obviously not advisable to treat extrapolation of the differences to the remaining languages with any great confidence.

The evaluation was run as a set of webpages where participants were asked – using headphones – to listen to the samples and to type in what they heard. Multiple listens were allowed as some of the the natural sentences were longer than the short SUS we would typically use. For the first two conditions, a balanced design was used so that each listener heard each utterance text only once, while each text was heard an equal number of times in both conditions over the whole evaluation. Each listener heard 20 sentences spoken in each condition. For English and Romanian where the SUS condition was also included, listeners heard a further set of 20 SUS sentences.

4.2. Results

Word error rates for the first 2 conditions are shown in Figure 3. For all languages besides English, a similar pattern can be observed: listeners' transcriptions of natural speech attain a WER of 8–12%, and in all cases the TTS system attain WERs approximately 1.5 times worse. This is consistent with the difference between WERs for natural speech and decent benchmark systems in larger scale evaluations on standard corpora. For example, natural speech and the Festival benchmark system attained WERs of 17% and 25% respectively in the 2011 Blizzard Challenge evaluation [29]. The results for English are the exception to the general pattern: the WER for synthetic speech is over 4 times worse than that of natural speech. From prior knowledge and from looking at listeners' transcriptions, it seems clear that this is due to the fact that TTS is based on letters in a language with such an opaque letter-to-sound relationship. In all languages except Polish, the difference between the first two conditions (natural speech and TTS) found to be statistically significant (with $\alpha = 0.05$) using the bootstrap procedure of [30].

As expected, WERs for the SUS sentences are much higher than those for natural sentences: 24.8% and 69.4% for Romanian and English, respectively.

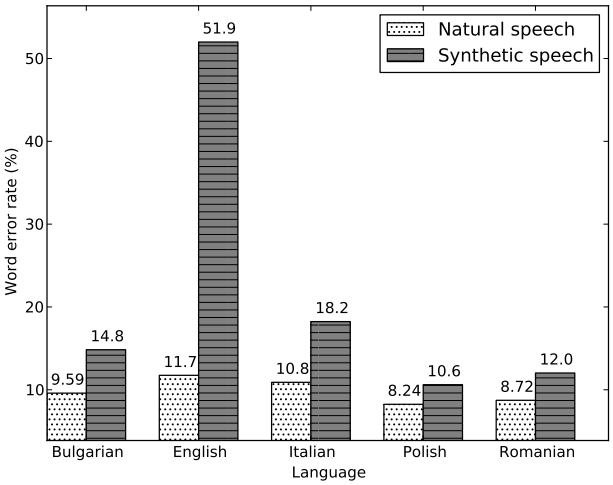

Figure 3: Word error rates for TTS systems and natural speech for 5 of the 14 systems built from the Tundra corpus.

5. Conclusions

We have presented tools for building TTS front-ends in a way that exploits unsupervised learning techniques to side-step the need for language-specific expert knowledge and resources such as pronunciation lexicons, phoneme inventories and part of speech taggers. We have shown how the tools were applied to the languages of the Tundra corpus to produce TTS systems in 14 languages. As we had previously built the Tundra corpus from found data using minimal supervision and language specific knowledge, these TTS systems represent the output of our entire pipeline of tools, and show the type of voice which any interested developer should be able to build using our toolkit (which will be made freely available) despite a lack of language-specific or speech technology expertise, if a source of speech and text data can be found. Five of the voices were evaluated in a listening test for intelligibility, which we consider to show that systems of reasonable quality can be built by applying our tools to publicly available audiobook data, assuming orthographies of similar transparency to those of Bulgarian, Italian, Polish and Romanian. While evaluation of the remaining systems that can be heard in the demo is still ongoing, the results for five languages published here – having been obtained from a standardised, publicly available corpus – are intended to be useful benchmarks against which future work can be compared.

6. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement Nº 287678.

The research presented here has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF: `http://www.ecdf.ed.ac.uk`). The ECDF is partially supported by the eDIKT initiative (`http://www.edikt.org.uk`).

Thanks to Vasilis Karaiskos for setting up the webpages for the listening test.

7. 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. Kominek, T. Schultz, and A. W. Black, "Voice building from insufficient data – classroom experiences with web-based language development tools," in *Proc. 6th ISCA Speech Synthesis Workshop*, Bonn, Germany, 2007, pp. 322–327.

[3] R. Tucker and K. Shalonova, "Supporting the creation of TTS for local language voice information systems," in *Proc. Interspeech 2005*, Lisbon, Portugal, Sep. 2005, pp. 453–456.

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

[5] 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.

[6] J. Černocký, "Speech processing using automatically derived segmental units: Applications to very low rate coding and speaker verification," Ph.D. dissertation, Universite Paris-Sud, Dec 1998.

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

[8] P. Mermelstein, "Automatic segmentation of speech into syllabic units," *The Journal of the Acoustical Society of America*, vol. 58, no. 4, pp. 880–883, 1975.

[9] Z. Xie and P. Niyogi, "Robust acoustic-based syllable detection," in *Proceedings of the ICSLP, International Conference on Spoken Language Processing*, 2006.

[10] J. Kominek, "Tts from zero: Building synthetic voices for new languages," Ph.D. dissertation, Carnegie Mellon University, 2009.

[11] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Trans. Speech Lang. Process.*, vol. 4, no. 1, pp. 3:1–3:34, Feb. 2007.

[12] C. Christodoulopoulos, S. Goldwater, and M. Steedman, "Two decades of unsupervised POS induction: How far have we come?" in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, October 2010, pp. 575–584.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, March 2003.

[14] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, 1990.

[15] 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.

[16] 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.

[17] 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.

[18] N. Braunschweiler and S. Buchholz, "Automatic sentence selection from speech corpora including diverse speech for improved HMM-TTS synthesis quality," in *Proc. Interspeech*, Florence, Italy, Aug. 2011, pp. 1821–1824.

[19] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*.

[20] G. Murray, S. Renals, and M. Taboada, "Prosodic correlates of rhetorical relations," in *Proceedings of HLT/NAACL ACTS Workshop, 2006, New York City, USA*, Jun. 2006.

[21] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

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

[23] 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*, 2012.

[24] 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.

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

[26] 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, Jan. 2011.

[27] 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.

[28] C. Benoit, M. Grice, and V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences," *Speech Communication*, vol. 18, no. 4, pp. 381 – 392, 1996.

[29] S. King and V. Karaiskos, "The blizzard challenge 2011," in *Proc. Blizzard Challenge 2011*, sep 2011.

[30] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *Proc. ICASSP '04*, vol. 1, 2004, pp. 409–12.