Creating German Unit Selection Voices for the MARY TTS Platform from the BITS Corpora

Marc Schröder and Anna Hunecke

DFKI GmbH
Saarbrücken, Germany
marc.schroeder@dfki.de, anna.hunecke@dfki.de

Abstract

The present paper reports on the creation of German unit selection voices from corpora which had been recorded and annotated previously in the BITS project. We describe the unit selection mechanism of our MARY TTS platform, as well as the tools for creating a synthesis voice from a speech corpus, and their application to the creation of German unit selection voices from the BITS corpora. Because of reservations concerning the mismatch of phonetic chains predicted by the German TTS components in MARY and the manually corrected database labels, we compared voices based on the manually corrected labels with voices based on automatic forced alignment labelling. We compute the diphone coverage for both types of voices and show that it is a reasonable approximation of the German diphone set. A preliminary evaluation confirms the expectations: while the manually corrected versions show a higher segmental accuracy, the automatically labelled versions sound more fluent.

1. Introduction

Unit selection synthesis is becoming a mature technology. Introduced in the mid-1990s [1, 2], it has matured over the last decade to the extent that now a regular competition, the Blizzard Challenge is being organised, where different data-driven synthesis algorithms are compared based on synthesis voices prepared from the same data. The vast majority of commercial TTS systems are based on unit selection technology; they are covering an increasing number of languages and voices.

Research systems, and particularly open-source systems, are less numerous. By far the most well-known system is Festival [3]; it contains two unit selection implementations, a cluster unit selection [4] and a generic unit selection [5]. The admirable Festvox toolkit provides support for creating custom synthesis voices, in the form of source code and documentation. The FreeTTS system [6] is a Java based reimplementation of code derived from Festival, and contains an implementation of the cluster unit selection algorithm. The BOSS system [7] implements a non-uniform unit selection method, which uses phrase- or word-sized units when these are found in the corpus, and reverts to smaller units otherwise. The MARY platform [8] became open source in early 2006, but until recently could generate audio only using the MBROLA [9] diphone synthesiser. A first unit selection component was added for US English [10] and released as open source.

Research on German speech synthesis, and German unit selection technology, seems to be progressing rather slowly. Indeed, there seem to be only a very limited number of German unit selection systems developed purely in Academia – we could only find two. The unit selection system BOSS [7] is available as open source; it comes with the Verbmobil database Lioba, which is somewhat tilted towards the domain of appointment negotiation. A general-domain German unit selection system based on Festival [3] has been developed at IMS Stuttgart [11] and continues to be developed in the Smartweb project [12]. However, it does not seem to be publicly available.

One important factor slowing down the development of unit selection systems in research labs is the cost associated with the creation of unit selection corpora. In order to lower that barrier, the project BITS [13] was funded to create unit selection voice databases, annotate them, and make them publicly available.

The present paper reports on the creation of publicly available German unit selection voices for the MARY TTS platform, based on the BITS corpora. The paper is organised as follows. We start by presenting the basic properties of the unit selection system developed in the framework of the MARY platform, and report on work in progress on an open-source toolkit for creating unit selection synthesis voices. We then describe the BITS corpora used as speech material for voice creation in the present paper, and report on our experiences building synthetic voices from these corpora.

2. The MARY unit selection system

2.1. The open source MARY TTS platform

MARY (Modular Architecture for Research on speech sYnthesis) is a platform for research, development and teaching on text-to-speech synthesis. Originally developed for German [8], it was extended to US English by incorporating some TTS modules from the FreeTTS project, and, as the result of a student project, to Tibetan. MARY uses an XML-based representation format for its data, which makes it possible to access intermediate processing states, and to connect it to other XML-based processing components [14].

Apart from being a research platform, MARY is also a stable Java server capable of multi-threaded handling of multiple client requests in parallel.

The design is highly modular. A set of configuration files, read at system startup, define the processing components to use. For example, the file `german.config` defines the German processing modules, `english.config` defines the English modules, etc. If both files are present in the configuration directory, both subsystems are loaded when starting the server. Each synthesis voice is defined by a configuration file: `german-mbrola-de7.config` loads the MBROLA voice de7, `english-arctic-jmk.config` the unit selection voice built from the Arctic recordings of speaker jmk [15], etc.

Each synthesis module has an input and an output format,

which can be flexibly defined. This makes it extremely easy to define pipeline architectures for processing any given input format into one or more output formats, without explicitly stating the required chain of modules. Starting from the input format specified for the system input (e.g., plain text, SSML [16], etc.), the TTS system searches a path through the available processing components until it arrives at the requested output format (e.g., audio). Although this is a very simple mechanism for specifying a component architecture, it seems to be sufficient for the processing requirements of a TTS system.

For the generation of audio, MARY includes the concept of a collection of waveform synthesisers; these are defined in an extensible way through the MARY configuration files. Currently, the list of available waveform synthesisers includes the MBROLA diphone synthesiser; an LPC-based diphone synthesiser provided by FreeTTS; the MARY unit selection synthesiser covered in the present paper; and an experimental interpolating synthesiser, creating intermediate voices from two existing unit selection voices [17] using a spectral interpolation algorithm [18].

The architecture of the MARY platform as well as the English and Tibetan processing components are available under a liberal BSD-style license. The German processing components are available free of charge under a research license. By permission from the MBROLA team, MBROLA binaries and voices are provided with MARY under the MBROLA license.

The system runs under Windows, Linux, Solaris, and Mac OS X. A comfortable graphical installer can be downloaded from the MARY website. During installation, users can indicate which components they want to install; only these components are downloaded from the MARY page.

In order to avoid misconfigurations, the configuration files define a number of dependencies, which are checked automatically at every system startup. If a component is found to be missing, the system offers to download it from the MARY website.

2.2. Unit selection in MARY

The unit selection system in MARY implements a generic unit selection algorithm, combining the usual steps of tree-based pre-selection of candidate units, a dynamic programming phase combining weighted join costs and target costs, and a concatenation phase joining the selected units into an output audio stream.

Units to concatenate are uniform. An early version of the system [10] used phoneme units. After getting feedback at the Blizzard Challenge Workshop 2006, we switched to diphone units, because joining in the mid-section of phonemes is expected to introduce less discontinuities than joining at phoneme boundaries. For each target diphone, a set of candidate units is selected by separately retrieving candidates for each halfphone through a decision tree, and retaining only those that are part of the required diphone. When no suitable diphone can be found, the system falls back to halfphone units.

The most suitable candidate chain is obtained through dynamic programming, minimising a weighted sum of target costs and join costs. Both are themselves a weighted sum of component costs. Target costs cover the linguistic properties of units, and the way they match the linguistically defined target. In addition, acoustic target costs can be used. These are currently used for comparing a unit's duration and F0 to the ones predicted for the target utterance by means of regression trees trained on the voice data. In the future, we intend to use acoustic target costs to also cover expressivity-related acoustic measures, such as spectral tilt or other robust measures of voice quality.

Join costs are computed as a weighted sum of F0 difference and of spectral distance, computed as the absolute distance in 12-dimensional MFCC space. We had experimented with a step function for the F0 penalty, based on the reasoning that small F0 deviations can be corrected by a smoothing algorithm [10]; currently, we are using a linear cost function instead and avoid signal post-processing as it seems to degrade the overall quality.

Like all unit selection systems, we face the challenge of determining appropriate weights for the individual target and join cost components. As we have not yet developed a principled way of determining these weights, we have set a number of ad hoc values through iterative listening and adapting. The resulting weights give equal importance to join costs and to target costs, a higher importance to F0 continuity than to spectral continuity, and a higher importance to duration and F0 targets than to phonetic context.

After the chain of units minimising these costs is determined, the units are retrieved from a timeline file and concatenated using overlap-add of one pitch period at the unit boundaries. The timeline file currently contains uncompressed PCM audio data, but is designed in a way that makes it easy to use more efficient encodings in the future.

The system is reasonably efficient: it synthesises speech about ten times faster than real-time on a recent Core 2 Duo processor. Decision trees and feature vectors required for the cost computation are held in memory; audio data is retrieved from a file after selection.

2.3. The voice creation toolkit in MARY

We are in the process of developing a toolkit for creating voices for MARY. We originally used the Festvox tools [19], and we continue to be deeply grateful to their creators for making them available to the community. However, it appears that some aspects of Festvox are tightly linked to the Festival system, and we felt that in the long run, the gain in control and flexibility justifies the development of our own voice creation toolkit.

The system combines an extensible list of "voice import components" in a graphical interface which is currently still very simple (see Figure 1). The user can select a series of import components, which are run in sequence. A progress bar is shown for the component which is currently running. After successful completion, the component is coloured in green; if processing fails, it is displayed in red, and processing of subsequent components is aborted. Configuration of non-default file system paths and special settings for the components is done via command-line options.

The voice import components that are currently available include components for automatic labelling using Sphinxtrain [20]; for importing text files in Festvox format; for predicting unit features with MARY; for making sure the unit labels and the feature chain predicted by MARY are properly aligned; for pitchmarking using Praat [21]; for the conversion of data into the compact format required by the MARY unit selection runtime system; for building classification trees for candidates using the wagon tool from the Edinburgh speech tools [22]; for pruning outliers from the generated trees; and for creating regression trees for duration and F0.

One of the most time-consuming tasks is the training of classification trees for the prediction of candidate units. Similarly to [4], we use acoustic distance between units as the impurity measure, and run wagon based on distance tables. In

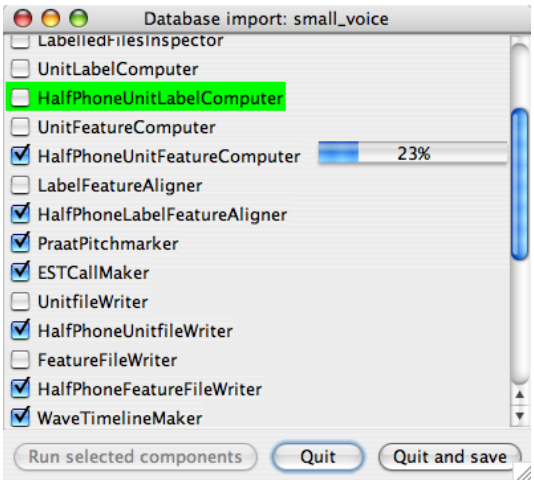

Figure 1: The MARY voice creation toolkit at work. In the situation shown, half-phone unit labels have been created successfully, unit features are being computed, and a number of components are scheduled for subsequent execution.

order to speed up the process on a multi-processor machine, the MARY CartBuilder component can run several wagon processes in parallel. Given the fact that the computation of acoustic distances is currently done in a single Java process, there is a limit to the number of wagon processes that should reasonably be started in parallel; we have experienced considerable speedup with running 3-5 wagon processes alongside one Java process on an 8-processor machine.

The MARY voice creation toolkit currently requires a considerable amount of expert knowledge in order to set paths correctly via command-line options and to select the right components for the task at hand. We intend to develop a more intuitive system providing groupings of the components that are usually required for a given task. For example, components working with halfphones are required for creating the necessary files to build classification trees for pre-selection of candidate units, but phone-sized units are needed for training regression trees for the prediction of duration and F0.

3. The BITS corpora

The BITS corpora were produced by the Bavarian Archive for Speech Signals (BAS) at Ludwig-Maximilians University, Munich, to provide a publicly available synthesis corpus for German. Two different kinds of corpora were recorded: logatome corpora for diphone synthesis and unit selection corpora. This paper only deals with the latter.

The unit selection part consists of 1683 sentences covering all German diphones and a few selected French and English diphones. A subset of the sentences was selected from a Newspaper Corpus (TAZ corpus) with a greedy algorithm. Additionally, semantically unpredictable sentences, provided by the IMS at University of Stuttgart, trade names and proverbs are contained in the set. Four speakers (two female, two male) were recorded with a close-talking microphone, a large membrane microphone and a laryngograph. The sentences were annotated with phonetic and prosodic labels automatically, then corrected by hand.

The corpus is distributed through the European Language Resources Association (ELRA) and can be ordered via the BAS website (`http://www.bas.uni-muenchen.de`).

4. German unit selection voices from the BITS corpora

4.1. Manual vs. automatic annotation

Since the BITS corpora have hand-corrected labels, they capture some phonetic detail, such as coarticulation effects and segmental reductions as they were realised by the speaker (e.g., Schwa elisions, nasal assimilations, or idiosyncratic devoicing). This poses a problem for the MARY system, since the phonemes predicted by MARY do not reflect these effects. As a result, even though a given syllable or word may be in the corpus, it may not be possible to retrieve the corresponding units. For example, one speaker frequently reduced Schwas: For the word “dunkel” (dark), the phonological form /dUNk@l/ was realised phonetically as [dUNkl]. A lookup of candidate units for /dUNk@l/ would need to find Schwa units from a different part of the corpus, even though the original word was available.

A proper solution to this problem would be a trainable postlexical phonological component, to be trained on the speech data from a given speaker in order to capture the speaker’s pronounciation rules. However, such a component is not yet realised in MARY.

In building synthetic voices for MARY from the BITS corpora, we therefore had two choices:

- use the existing manual annotation, knowing that suboptimal candidate units will be retrieved;
- use a fully automatic annotation created by forced alignment of the audio recordings with a phoneme chain created from the text using the MARY phonemisation component.

We decided to explore the trade-offs between both approaches by building two voices from each of the four databases: one with manual (M) and one with automatic (A) labels. We refer to the resulting voices as M1-4 and A1-4, respectively.

The expectation was that the A voices would show some segmental errors introduced by the uncorrected automatic labelling, but that overall the fluidity of the speech would be higher than for the M voices. In particular, it could be expected that the average length of segments joined would be higher for the A than for the M voices. The M voices, on the other hand, would be expected to have more accurate segmental pronounciations.

4.2. Voice creation

For the creation of the voices, the voice creation toolkit described in section 2.3 was used.

For the M (manually labelled) versions of the voices, additional voice import components were implemented which created labels and features based on the given labels. In the process, the phone labels of the annotation had to be mapped to the ones used by MARY, because some of the diacritics were not used by MARY, and some phone symbols were different. Also, in the BITS corpora, vowels followed by “6” (a-schwa) were annotated as diphthongs and had to be split up for MARY. For the computation of the features, first the phones and ToBI tones predicted by MARY were replaced with the actually annotated phones and tones. This modified version was then sent to MARY to compute the unit features needed for computing the target costs.

The automatic labels for the A versions of the voices were created with the components calling SphinxTrain and Sphinx2,

using the phoneme chain predicted by MARY from the text. We enriched the MARY pronounciation lexicon to make sure that the text is transcribed as accurately as possible. In the textual form of the BITS corpus, we found 459 unknown words and 123 words interpreted as English words (many of them proper names). Out of these, we manually transcribed 338 of the unknown words, and 40 of the words recognised as English; the remaining words were transcribed properly by the MARY letter-to-sound components. The unit features for the A voices were fully based on MARY predictions from text.

After the labels and features were created, the usual voice building steps were performed for all voices: First, the pitchmarks were calculated from the laryngograph files, using Praat, and with reasonable estimates of the pitch range of each speaker to minimise the risk of octave jumps. Pitch-synchronous mel-frequency cepstral coefficient (MFCC) vectors were computed using the EST tools.

The units, unit features and audio data were converted into a format suitable for the efficient use in the run-time unit selection components. In addition to the purely symbolic unit feature predicted by MARY, the unit F0 and duration were included as acoustic unit features, in view of the computation of acoustic target costs. Join cost features were computed at unit boundaries, comprising 12 MFCCs plus F0, and stored in a file allowing to access them efficiently.

For each voice, regression trees were built to predict phone duration and initial, medial and final log F0 in each syllable, to be used as acoustic targets and potentially for signal post-processing.

For the pre-selection of candidate units, classification trees were built using acoustic similarity as the impurity measure. Acoustic similarity was computed as a combined measure consisting of duration, F0, and linearly time-stretched average Mahalanobis distance between MFCC frames. This tree-building approach is similar to the cluster unit selection algorithm proposed by Black and Taylor [4]; however, our leaves contain between 50 and 100 candidates, for which full target costs are computed at run-time. The classification trees contain half-phone units; for generating diphone candidates, candidates are looked up for both halfphones, and only those that belong to the needed diphone are retained. This method makes it simple to fall back to halfphones: when no instance of a given diphone is found, the two sets of halfphone candidates are retained.

A pruning algorithm was implemented to remove outliers from the leaves of the pre-selection tree. This is particularly useful with fully automatic labelled data, as it can identify some of the most obvious labelling errors. One important kind of outliers are units labelled as silence which are not actually silence; we apply an energy criterion to identify these, based on a silence cutoff value determined from an energy histogram. A second kind of outlier are units that are too long, e.g. because a long portion of silence was labelled to be part of the unit, or because of wrongly predicted phoneme chains, leading to several phonemes to be labelled as a single one. We use a cutoff of 200 ms maximum duration for a halfphone: every non-silence unit that is longer than this threshold is removed. A third kind of outlier are units that have extreme values in the probability ratings generated by wagon during tree training. These are also removed from the pre-selection tree.

We have observed that some of the problems arising from automatic labelling could be filtered out using this pruning step. This is reflected in the amount of data pruned: it lies between 0.9 and 1.2% of the units for the M voices, and between 1.5 and 2.1% of the units for the A voices. However, more subtle pronounciation deviations could not be identified using this approach.

In the runtime system, weights were fine-tuned to reach a balance between linguistic and acoustic target costs on the one hand, and join costs on the other hand. Even though the weights are normalised so that all target cost weights and all join cost weights sum to one, the fact that duration and F0 are currently not normalised makes it necessary to manually adjust the weights for each voice. We did this so as to make sure that target costs are about as high join costs on average, and acoustic target costs (duration + F0) are slightly higher than symbolic target costs (mainly phonetic context).

4.3. Phonetic coverage

One objective measure of the expected quality of a voice is the coverage of diphones as they occur in the language. Therefore, the phonetic coverage of the voices was measured both for the annotated phonemes and the phonemes predicted by MARY. To get an idea of how the coverage of the BITS corpora relates to the German language in general, the results were compared with the coverage of a large German corpus. For this purpose, we collected a textual corpus consisting of 978,269 sentences extracted from German ebooks from Project Gutenberg (`http://www.gutenberg.org`), and transcribed it fully automatically using the MARY phonemisation component.

For a phoneme set of 56 German phonemes, including some English and French xenophones, the phoneme coverage is 100% for the M voices, and 98% for the A voices, where the /T/ (voiceless English "th") is missing.

The diphone coverage varies slightly between the different voices, because for each voice, some of the sentences in the corpus could not be used for building the voice. Overall, the diphone coverage for the A voices, using the automatically predicted phonemes, is slightly worse (around 1690 diphones) than the coverage for the M voices (1770 diphones). Both figure are considerably lower than the number of different diphones found in the Gutenberg corpus (2306 diphones). It strikes the eye that these figures are substantially smaller than the number of $56 * 56 = 3136$ theoretically possible diphones – apparently, only around $2306/3136 = 73\%$ of these actually occur in German. Taking the Gutenberg figure as the reference, rather than the theoretically possible number of diphones, we can thus compute a diphone coverage of $1690/2306 = 73\%$ for the A voices and $1770/2306 = 77\%$ for the M voices.

To get an idea not only of the quantity but also of the quality of the diphone coverage in the BITS voices, we also looked at the distribution of the diphones. Figure 2a shows the distribution of the diphones in the Gutenberg corpus. It can be seen that the distribution follows Zipf's law, according to which the frequency of a word (or in this case, a diphone) is roughly inversely proportional to its rank in the frequency table.

Figures 2b and 2c show the relative frequencies of diphones in the BITS voices A1 and M1, respectively. The distribution curves for the other BITS voices look similar. Whereas the distribution of A1 is highly similar to the distribution of the Gutenberg corpus, M1 has substantially more outliers. Most of these are related to the Schwa elisions annotated in the BITS corpora: For example, the diphone "t_n" (arising by a reduction of /t@n/) occurs far more frequently in M1 than in the Gutenberg corpus, which is transcribed without phonological reduction.

Figure 3 shows a different way of comparing the diphone distribution in A1 and M1 to the Gutenberg corpus. The coverage ratio v shown in the figure is computed for each diphone

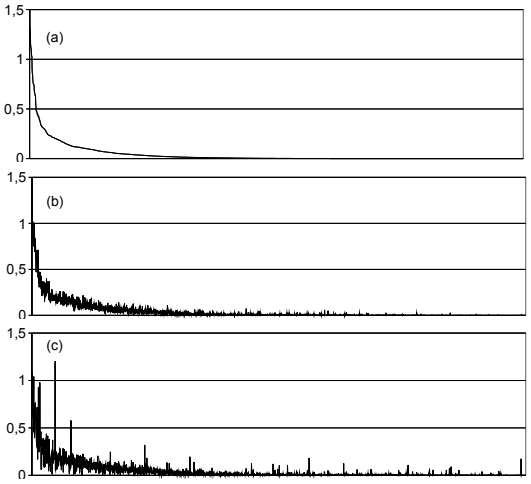

Figure 2: Relative frequency of diphones (a) in the Gutenberg corpus, (b) in voice A1, and (c) in voice M1. In all three, diphones are sorted on the X axis according to their frequency of occurrence in the Gutenberg corpus.

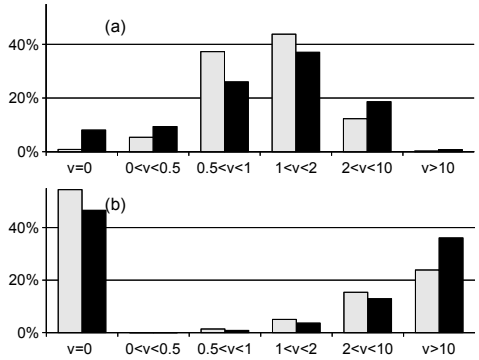

Figure 3: Coverage ratio of Gutenberg diphones for voices A1 (light) and M1 (dark), for (a) the frequent and (b) the rare half of Gutenberg diphones.

as the ratio of the relative frequency of the diphone in the voice and the relative frequency in the Gutenberg corpus. The graph shows the percentage of diphones with a given coverage ratio. Figure 3a represents the most frequent half of the Gutenberg diphones (the left half of Figure 2a), Figure 3b represents the least frequent half of the Gutenberg diphones (the right half of Figure 2a).

It can be seen that for the frequent German diphones, v values around 1 dominate, i.e. the coverage is close to the Gutenberg distribution. This is true for both voices, A1 and M1, with a slight advantage for the automatic labelling method which was also used for transcribing the text corpus. For the rare diphones, on the other hand, we see a clear dichotomy between diphones which are missing and diphones which are over-represented. Over-representation seems generally inevitable when trying to approximate a Zipf distribution with a much smaller corpus: even by occurring only once in the voice, a rare diphone already has a much higher relative frequency than the very low relative frequency in the large corpus.

4.4. Initial assessment of quality

Informal listening tests were performed to compare the quality of the voices, using ten example sentences for each of two text styles. First, news sentences were extracted from the web page of the German newspaper TAZ. Given the fact that the recording script was based on text material from the TAZ newspaper corpus, this can be considered a "within-domain" condition, which can be expected to lead to a relatively good synthesis quality. As a second text style, we used ten sentences from the fairy tale "Däumelieschen" available from the Gutenberg collection (`http://www.gutenberg.org`). This domain being different from the recording script, it can be considered a priori more challenging.

The first author, a trained phonetician, listened to the eight versions of each sentence, generated with the A and the M voice created from each of the four BITS corpora. Labels "+", "0" and "-" were assigned to each utterance, where "+" indicated that only minor problems could be heard, "0" indicated audible prosodic deviations or minor segmental deviations, and "-" indicated clearly wrong segments. While the individual ratings are certainly subjective, and therefore are not reported in detail, some relatively clear patterns seem to emerge from this preliminary assessment.

Globally, more discontinuities can be heard in the M voices than in the A voices. This is reflected in the average length of consecutive unit stretches selected – 3.5 halfphones for M voices, and 4.0 halfphones for A voices. Furthermore, the M voices tend to sound a bit over-articulated. The A voices generally have a more natural prosody, but occasionally labelling errors are very prominent.

In the preliminary assessment, the A voices received better overall ratings than the M voices, reflecting the fact that prosodic naturalness and continuity were better for many of the sentences, and bad segments occurred only in a few sentences.

The news style sentences received better scores than the fairy tale sentences, lending support to the hypothesis that it is easier to synthesise within-domain material at good quality than material from a different type of text.

"-" labels, indicating segmental errors, occurred mostly for the A voices, but occasionally also for the M voices.

These first impressions provide an indication regarding the trade-off between the M and A voices which motivated the creation of both voices (see Section 4.1). Manual labelling leads to a considerable reduction of wrong segments in the output, and therefore remains a requirement for the professional creation of voice databases which cannot be replaced with filtering methods at the stage of tree pruning; however, when the predicted chain of target units does not reflect the kinds of postlexical phonological effects exhibited by the speaker, the continuity of the generated speech is reduced.

These findings suggest that it is not easy to choose between the M and the A version of a voice. Instead, it seems that the effort to develop a postlexical phonological component which can learn to map lexical-phonemic transcriptions to speaker-dependent surface-phonetic transcriptions would be well justified, because it could be expected to combine the benefits of both methods.

5. Conclusion

We have described the creation of unit selection synthesis voices in the MARY TTS platform, using the German corpora recorded for this purpose in the BITS project. Comparing voices created from the manually corrected labels in the database with voices created from fully automatic forced-alignment, we found systematic differences: higher segmental accuracy for the manually labelled voices, but more natural prosody and higher continuity for automatically labelled voices.

The resulting synthesis voices are work in progress, and can certainly be improved; but they are already quite intelligible German unit selection voices. Given the sparsity of publicly available unit selection systems for German, we will make the resulting voices available for download as soon as possible, under the same research license as the existing German MARY TTS components.

Future work will address various aspects of the current system. In the context of the present paper, the most obviously needed improvement is a trainable postlexical component. In addition, the general voice-building and unit selection methods will be improved, as time permits, along the following lines. Acoustic target and join costs should be computed in a normalised acoustic space, i.e. in z-scores. This will make it easier to set the weights for various target and join cost components. It will also allow us to reuse one speaker's prosody model with another speaker's voice, simply by setting the denormalisation coefficients to the new speaker's mean and standard deviation. Pooling training data from several voices for more robust prosody prediction is another option.

These developments are also in line with our mid-term goals of making progress towards parametrisable expressive speech synthesis. In this context, a major issue in view of high-quality signal modification and efficiency is the representation of the audio signal, e.g. as line spectrum pairs (LSP).

6. Acknowledgements

The work reported here was supported by the EU project HUMAINE (IST-507422), by the DFG project PAVOQUE, and by the ProFIT project IDEAS4Games. The authors would like to thank the BITS team for making their voice databases available, and for allowing us to distribute synthesis voices created from them; Sacha Krstulović for helping with the definition and implementation of file formats and voice import components; Mat Wilson for the design and implementation of a GUI for recording new voices; and Maximilian Kwapil for implementing the pruning algorithm for outliers in the classification tree.

7. References

[1] A. W. Black and N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis," in *Proc. Eurospeech*, vol. 1, Madrid, Spain, 1995, pp. 581–584.

[2] A. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, Atlanta, Georgia, 1996.

[3] A. W. Black, P. Taylor, and R. Caley, "Festival speech synthesis system, edition 1.4," Centre for Speech Technology Research, University of Edinburgh, UK, Tech. Rep., 1999. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival

[4] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proc. Eurospeech*, Rhodes/Athens, Greece, 1997.

[5] R. A. J. Clark, K. Richmond, and S. King, "Festival 2 – build your own general purpose unit selection speech synthesiser," in *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, USA, 2004, pp. 173–178.

[6] "Freetts 1.2," http://freetts.sourceforge.net, 2005.

[7] E. Klabbers, K. Stöber, R. Veldhuis, P. Wagner, and S. Breuer, "Speech synthesis development made easy: The Bonn Open Synthesis System," in *Proc. Eurospeech*, Aalborg, Denmark, 2001, pp. 521–524.

[8] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, pp. 365–377, 2003. [Online]. Available: http://mary.dfki.de

[9] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. van der Vrecken, "The MBROLA project: Towards a set of high quality speech synthesisers free of use for non commercial purposes," in *Proc. ICSLP*, Philadelphia, USA, 1996.

[10] M. Schröder, A. Hunecke, and S. Krstulović, "OpenMary – open source unit selection as the basis for research on expressive synthesis," in *Proc. Blizzard Challenge'06*, 2006.

[11] A. Schweitzer, N. Braunschweiler, T. Klankert, B. Möbius, and B. Säuberlich, "Restricted unlimited domain synthesis," in *Proc. Eurospeech*, Geneva, Switzerland, 2003.

[12] D. Sonntag, R. Engel, G. Herzog, A. Pfalzgraf, N. Pfleger, M. Romanelli, and N. Reithinger, "Smartweb handheld – multimodal interaction with ontological knowledge bases and semantic web services," in *Proc. Intl Workshop on AI for Human Computing (AI4HC) at IJCAI*, Hyderabad, India, 2007.

[13] T. Ellbogen, F. Schiel, and A. Steffen, "The BITS speech synthesis corpus for German," in *Proc. LREC*, Lisbon, Portugal, 2004, pp. 2091–2094.

[14] M. Schröder and S. Breuer, "XML representation languages as a way of interconnecting TTS modules," in *Proc. ICSLP*, Jeju, Korea, 2004.

[15] J. Kominek and A. W. Black, "CMU ARCTIC speech databases," in *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, USA, 2004, pp. 223–224.

[16] M. R. Walker and A. Hunt, *Speech Synthesis Markup Language Specification*, W3C, 2001. [Online]. Available: http://www.w3.org/TR/speech-synthesis

[17] M. Schröder, "Interpolating expressions in unit selection," in *Proc. 2nd International Conference on Affective Computing and Intelligent Interaction (ACII'2007)*, Lisbon, Portugal, to appear.

[18] O. Turk, M. Schröder, B. Bozkurt, and L. Arslan, "Voice quality interpolation for emotional text-to-speech synthesis," in *Proc. Interspeech*, Lisbon, Portugal, 2005.

[19] A. W. Black and K. Lenzo, "Festvox: Building synthetic voices, edition 1.6," Language Technologies Institute, Carnegie Mellon University, PA, USA, Tech. Rep., 2002. [Online]. Available: http://www.festvox.org

[20] R. Mosur and K. A. Lenzo, *Sphinx-II User Guide*, CMU, http://cmusphinx.sourceforge.net/sphinx2/doc/sphinx2.html.

[21] P. Boersma and D. Weenink, "Praat: doing phonetics by computer." http://www.praat.org, 2007.

[22] S. King, A. W. Black, P. Taylor, R. Caley, and R. Clark, "Edinburgh speech tools library," http://www.cstr.ed.ac.uk/projects/speech_tools, 2003.