Multilingual Voice Creation Toolkit for the MARY TTS Platform

Sathish Pammi, Marcela Charfuelan, Marc Schröder

Language Technology Laboratory, DFKI GmbH
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany and
Alt-Moabit 91c, D-10559, Berlin, Germany
{sathish.pammi, marcela.charfuelan, marc.schroeder}@dfki.de

Abstract

This paper describes an open source voice creation toolkit that supports the creation of unit selection and HMM-based voices, for the MARY (Modular Architecture for Research on speech Synthesis) TTS platform. We aim to provide the tools and generic reusable run-time system modules so that people interested in supporting a new language and creating new voices for MARY TTS can do so. The toolkit has been successfully applied to the creation of British English, Turkish, Telugu and Mandarin Chinese language components and voices. These languages are now supported by MARY TTS as well as German and US English. The toolkit can be easily employed to create voices in the languages already supported by MARY TTS. The voice creation toolkit is mainly intended to be used by research groups on speech technology throughout the world, notably those who do not have their own pre-existing technology yet. We try to provide them with a reusable technology that lowers the entrance barrier for them, making it easier to get started. The toolkit is developed in Java and includes intuitive Graphical User Interface (GUI) for most of the common tasks in the creation of a synthetic voice.

1. Introduction

The task of building synthetic voices requires not only a big amount of steps but also patience and care, as advised by the developers of Festvox (Black and Lenzo, 2007), one of the most important and popular tools for creating synthetic voices. Experience creating synthetic voices has shown that going from one step to another is not always a straightforward task, especially for users who do not have an expert knowledge of speech synthesis or when a voice should be created from scratch. In order to simplify the task of building new voices we have created a toolkit aimed to streamline the task of building a new synthesis voice from text and audio recordings. This toolkit is included in the latest version of MARY TTS[1] (version 4.0) and supports the creation of unit selection and HMM-based voices on the languages already supported by MARY: US English, British English, German, Turkish and Telugu. The toolkit also supports the creation of new voices from scratch, that is, it provides the necessary tools and generic reusable run-time system modules for adding a language not yet supported by MARY.

The toolkit is mainly intended to be used by research groups on speech technology throughout the world, notably those who do not have their own pre-existing technology yet. We try to provide them with a reusable technology that lowers the entrance barrier for them, making it easier to get started.

This paper describes the MARY multilingual voice creation toolkit and it is organised as follows. We start in Section 2 with a brief review on the available open source toolkits for creation of synthetic voices. In Section 3 the MARY multilingual voice creation toolkit is described, the support for the creation of a new language and the voice building process are explained in detail. In Section 4 the strategy for quality control of phonetic labelling is explained. The experience with the toolkit is presented in Section 5 and conclusions are made in Section 6.

2. Open source voice creation toolkits

Among the open source voice creation toolkits available nowadays by far the most used system is Festival. The Festival Speech Synthesis Systems, on which Festvox is based, was developed at the Centre for Speech Technology Research at the University of Edinburgh in the late 90's. It offers a free, portable, language independent, run-time speech synthesis engine for various platforms under various APIs. This tool has been developed in C++ and also includes some scripts in Festival's Scheme programing language, it supports unit selection and HMM-based synthesis. Full documentation for creating a new voice from scratch is provided in the Festvox project (Black and Lenzo, 2007).

Another popular, free speech synthesis tool for non-commercial use (although no open source) that appeared in the 90's is the MBROLA system. The aim of the MBROLA project, initiated by the TCTS Lab of the Faculté Polytechnique de Mons (Belgium), is to obtain a set of speech synthesisers for as many languages as possible, and provide them free for non-commercial applications. Their ultimate goal is to boost academic research on speech synthesis, and particularly on prosody generation. MBROLA is a speech synthesiser based on the concatenation of diphones, no source code is available but binaries for several platforms are provided. For creating a new voice a diphone database should be provided to the MBROLA team and they will process and adapt it to the MBROLA format for free. The resulting MBROLA diphone database is made available for non-commercial, non-military use as part of the MBROLA project (Dutoit et al., 1996).

FreeTTS is a speech synthesis system written entirely in the Java programming language. FreeTTS was written by the Sun Microsystems Laboratories Speech Team and is based on CMU's Flite engine. Flite is a lite version of Festival and Festvox and requires these two programs for generating new voices for FreeTTS. The system currently supports English and several MBROLA voices (Walker et al., 2005).

Epos is a language independent rule-driven text-to-speech

[1] http://mary.dfki.de/

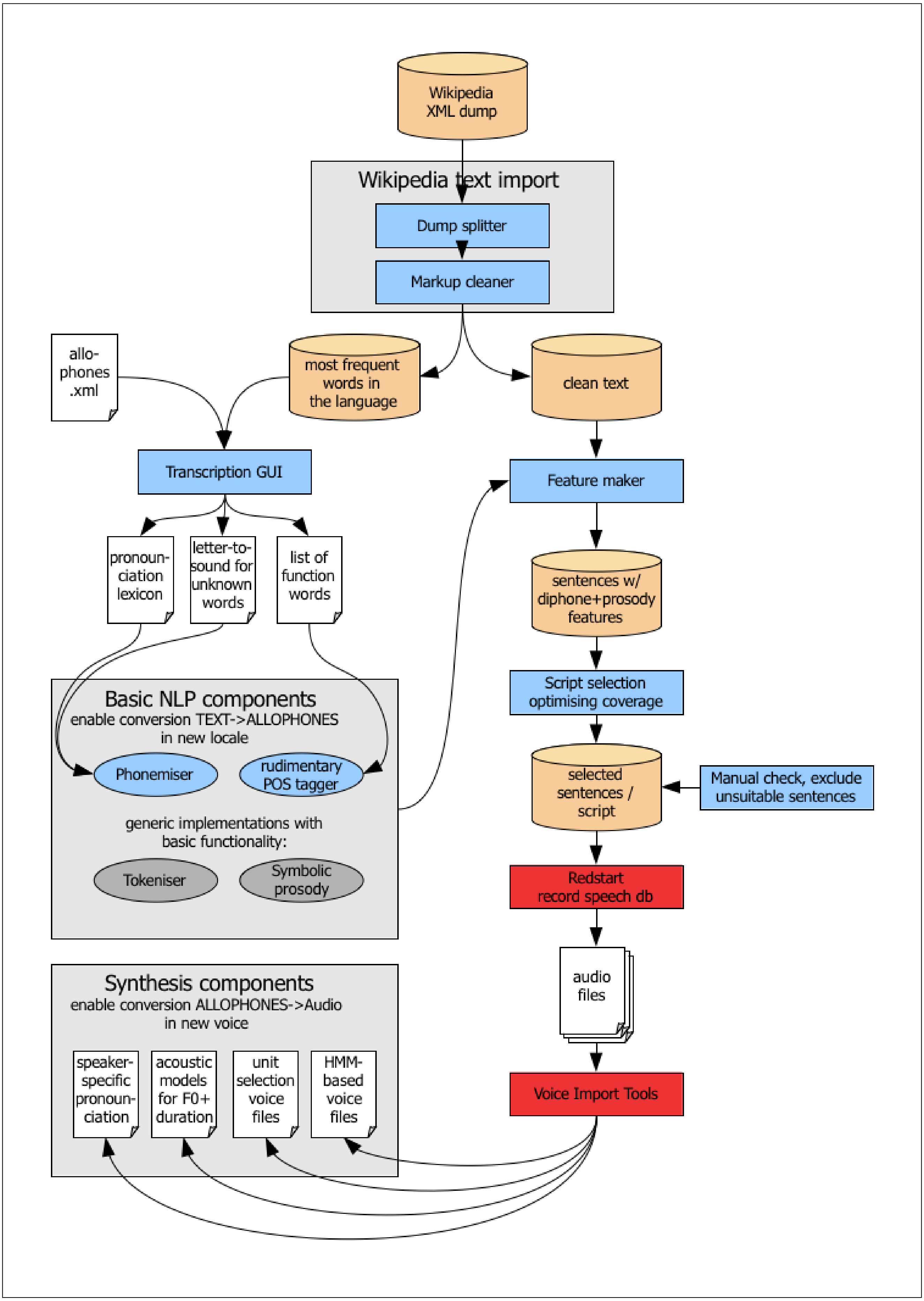

Wikipedia XML dump
Wikipedia text import
Dump splitter
Markup cleaner
allo-phones .xml
most frequent words in the language
clean text
Transcription GUI
Feature maker
pronoun-ciation lexicon
letter-to-sound for unknown words
list of function words
sentences w/ diphone+prosody features
Script selection optimising coverage
Basic NLP components
enable conversion TEXT->ALLOPHONES in new locale
Phonemiser
rudimentary POS tagger
generic implementations with basic functionality:
Tokeniser
Symbolic prosody
selected sentences / script
Manual check, exclude unsuitable sentences
Redstart record speech db
audio files
Synthesis components
enable conversion ALLOPHONES->Audio in new voice
speaker-specific pronoun-ciation
acoustic models for F0+ duration
unit selection voice files
HMM-based voice files
Voice Import Tools

Figure 1: Workflow for multilingual voice creation in MARY TTS, more information about this tool can be found in: http://mary.opendfki.de/wiki/VoiceImportToolsTutorial

system primarily designed to serve as a research tool. Epos aims to be independent of the language processed, linguistic description method, and computing environment. Epos supports Czech and Slovak text to speech synthesis, can be used as a front-end for MBROLA diphone synthesiser and uses a client-server model of operation (Hanika and Horák, 2005).

eSpeak is a compact open source formant synthesiser, that supports several languages. eSpeak was developed in C++ and provides support for adding new languages. eSpeak can be used as a front-end for MBROLA diphone synthesiser (Duddington, 2010).

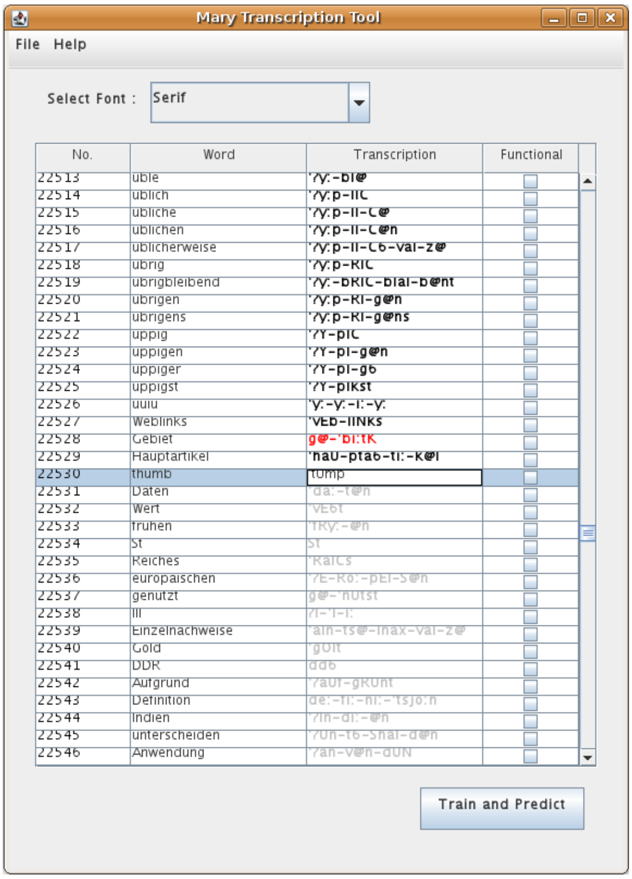
Gnuspeech is an extensible, text-to-speech package, based on real-time, articulatory, speech synthesis by rules. It includes Monet (Hill et al., 1995) a GUI-based database creation and editing system that allows the phonetic data and dynamic rules to be set up and modified for arbitrary languages. Gnuspeech is currently available as a NextStep 3.x version, and partly available (specifically Monet) as a version that compiles for both Mac OS/X and GNU/Linux under GNUStep (Hill, 2008).

One of the latest open source systems for developing statistic fully parametric speech synthesis is the HMM-based Speech Synthesis System (HTS), first version released in 2002. HTS and hts_engine API do not include any text analyser but the Festival system, the MARY TTS system, or other text analyser can be used with HTS. HTS and hts_engine are developed in C++. For the MARY TTS system the hts_engine has been ported to Java (Schröder et al., 2008). The development of a new voice with the HTS system is fully documented including training demos (Zen et al., 2006).

Most of the systems described above provide guidelines and recipe lists of the steps to follow in order to create a new voice, in most of the cases it is expected that the developer has a basic knowledge of the TTS system and speech signal processing and in some cases minimal programming skills. In MARY TTS, we have developed, in Java, a voice creation toolkit, that includes graphical user interfaces (GUIs) for most of the common tasks in the creation of a synthetic voice. This facilitates the understanding of the whole process and streamline the steps. All the steps are documented in the MARY wiki pages (http://mary.opendfki.de/wiki) and there is also the possibility to get support subscribing to the MARY mailing list.

3. Toolkit workflow

The steps required to add support for a new language from scratch are illustrated in Figure 1. Two main tasks can be distinguished: (i) building at least a basic set of natural language processing (NLP) components for the new language, carrying out tasks such as tokenisation and phonemic transcription (left branch in Figure 1); (ii) and the creation of a voice in the new language (right branch in Figure 1). Whereas high-quality support of a language will usually require language-specific processing components, it is often possible to reach at least a basic support for a language using generic methods (Black and Lenzo, 2007). In the following two sections we briefly explain these two branches.



Select Font : Serif

No.	Word	Transcription	Functional
22513	uble	'?y:-bl@	☐
22514	ublich	'?y:p-lIC	☐
22515	ubliche	'?y:p-lI-C@	☐
22516	ublichen	'?y:p-lI-C@n	☐
22517	ublicherweise	'?y:p-lI-C6-vaI-z@	☐
22518	ubrig	'?y:p-RIC	☐
22519	ubrigbleibend	'?y:-bRIC-blaI-b@nt	☐
22520	ubrigen	'?y:p-RI-g@n	☐
22521	ubrigens	'?y:p-RI-g@ns	☐
22522	uppig	'?Y-pIC	☐
22523	uppigen	'?Y-pI-g@n	☐
22524	uppiger	'?Y-pI-g6	☐
22525	uppigst	'?Y-pIkst	☐
22526	uuiu	'y:-y:-i:-y:	☐
22527	Weblinks	'vEb-lINks	☐
22528	Gebiet	g@-'bi:tK	☐
22529	Hauptartikel	'haU-pta6-ti:-k@l	☐
22530	thumb	'tUmp	☐
22531	Daten	'da:-t@n	☐
22532	Wert	'vE6t	☐
22533	fruhen	'fRy:-@n	☐
22534	St	St	☐
22535	Reiches	'RaICs	☐
22536	europaischen	'?E-Ro:-pEI-S@n	☐
22537	genutzt	g@-'nUtst	☐
22538	III	?I-'I-i:	☐
22539	Einzelnachweise	'aIn-ts@-lnax-vaI-z@	☐
22540	Gold	'gOlt	☐
22541	DDR	dd6	☐
22542	Aufgrund	'?aUf-gRUnt	☐
22543	Definition	de:-fi:-ni:-'tsjo:n	☐
22544	Indien	'?In-di:-@n	☐
22545	unterscheiden	'?Un-t6-ShaI-d@n	☐
22546	Anwendung	'?an-v@n-dUN	☐

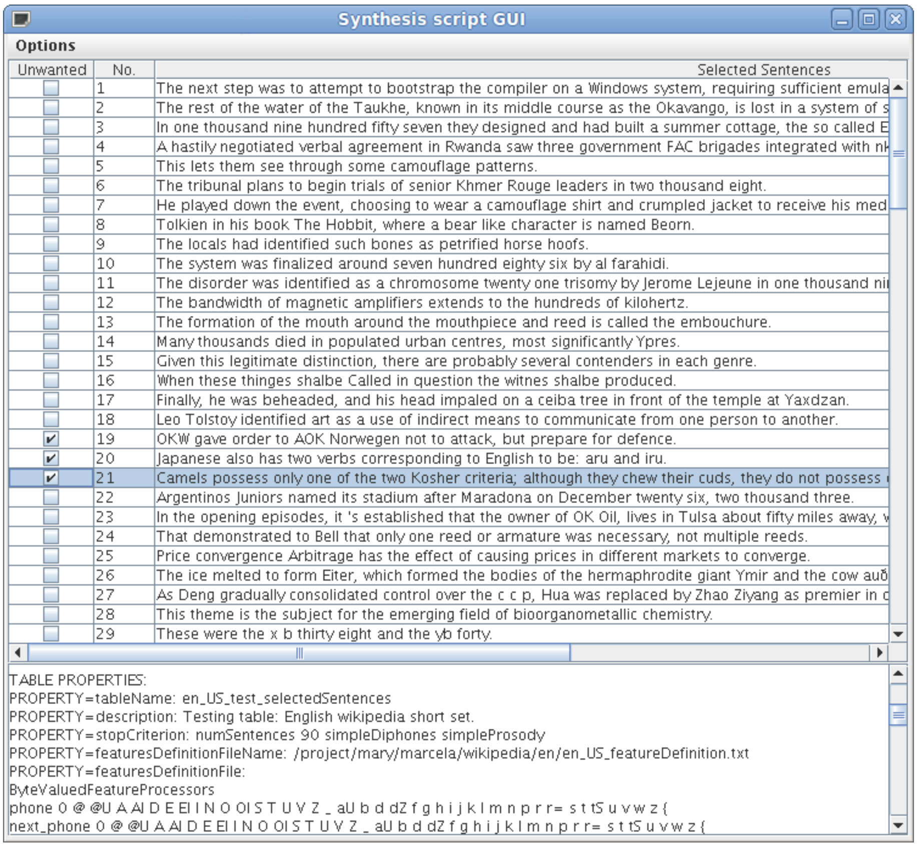
Figure 2: Transcription GUI: MARY semi-automatic word transcription tool, more information about this tool can be found in: http://mary.opendfki.de/wiki/TranscriptionTool.

3.1. New language support

For creating a voice from scratch or creating a voice for a language already supported by MARY TTS, our workflow starts with a substantial body of UTF-8 encoded text in the target language, such as a dump of Wikipedia in the target language. After automatically extracting text without markup, as well as the most frequent words, the first step is to build up a pronunciation lexicon, letter-to-sound rules for unknown words and a list of function words. In the following the extraction of this information with the transcription GUI is explained.

Transcription GUI

Using the `Transcription GUI`, Figure 2, a language expert can transcribe as many of the most frequent words as possible using the allophone inventory. We use an XML file, `allophones.xml`, describing the allophones of the target language that can be used for transcription, providing for each allophone symbol phonetic features that are going to be used for characterising the phone later. The phonetic features described in the XML file are: vowel length, vowel height, vowel frontness and lip rounding for vowels, consonant type, place of articulation and consonant voicing for consonants; therefore this XML file has to be prepared manually by a language expert. Once the allophones.xml file is available for a target language the transcription GUI can be used in the following way. After the transcription



Unwanted	No.	Selected Sentences
☐	1	The next step was to attempt to bootstrap the compiler on a Windows system, requiring sufficient emula
☐	2	The rest of the water of the Taukhe, known in its middle course as the Okavango, is lost in a system of s
☐	3	In one thousand nine hundred fifty seven they designed and had built a summer cottage, the so called E
☐	4	A hastily negotiated verbal agreement in Rwanda saw three government FAC brigades integrated with nk
☐	5	This lets them see through some camouflage patterns.
☐	6	The tribunal plans to begin trials of senior Khmer Rouge leaders in two thousand eight.
☐	7	He played down the event, choosing to wear a camouflage shirt and crumpled jacket to receive his med
☐	8	Tolkien in his book The Hobbit, where a bear like character is named Beorn.
☐	9	The locals had identified such bones as petrified horse hoofs.
☐	10	The system was finalized around seven hundred eighty six by al farahidi.
☐	11	The disorder was identified as a chromosome twenty one trisomy by Jerome Lejeune in one thousand ni
☐	12	The bandwidth of magnetic amplifiers extends to the hundreds of kilohertz.
☐	13	The formation of the mouth around the mouthpiece and reed is called the embouchure.
☐	14	Many thousands died in populated urban centres, most significantly Ypres.
☐	15	Given this legitimate distinction, there are probably several contenders in each genre.
☐	16	When these thinges shalbe Called in question the witnes shalbe produced.
☐	17	Finally, he was beheaded, and his head impaled on a ceiba tree in front of the temple at Yaxdzan.
☐	18	Leo Tolstoy identified art as a use of indirect means to communicate from one person to another.
☑	19	OKW gave order to AOK Norwegen not to attack, but prepare for defence.
☑	20	Japanese also has two verbs corresponding to English to be: aru and iru.
☑	21	Camels possess only one of the two Kosher criteria; although they chew their cuds, they do not possess
☐	22	Argentinos Juniors named its stadium after Maradona on December twenty six, two thousand three.
☐	23	In the opening episodes, it 's established that the owner of OK Oil, lives in Tulsa about fifty miles away, w
☐	24	That demonstrated to Bell that only one reed or armature was necessary, not multiple reeds.
☐	25	Price convergence Arbitrage has the effect of causing prices in different markets to converge.
☐	26	The ice melted to form Eiter, which formed the bodies of the hermaphrodite giant Ymir and the cow auð
☐	27	As Deng gradually consolidated control over the c c p, Hua was replaced by Zhao Ziyang as premier in c
☐	28	This theme is the subject for the emerging field of bioorganometallic chemistry.
☐	29	These were the x b thirty eight and the yb forty.

```
TABLE PROPERTIES:
PROPERTY=tableName: en_US_test_selectedSentences
PROPERTY=description: Testing table: English wikipedia short set.
PROPERTY=stopCriterion: numSentences 90 simpleDiphones simpleProsody
PROPERTY=featuresDefinitionFileName: /project/mary/marcela/wikipedia/en/en_US_featureDefinition.txt
PROPERTY=featuresDefinitionFile:
ByteValuedFeatureProcessors
phone 0 @ @U A AI D E EI I N O OI S T U V Z _ aU b d dZ f g h i j k l m n p r r= s t tS u v w z {
next_phone 0 @ @U A AI D E EI I N O OI S T U V Z _ aU b d dZ f g h i j k l m n p r r= s t tS u v w z {
```

Figure 3: Synthesis script GUI: MARY optimal text selection tool, more information about this tool can be found in: http://mary.opendfki.de/wiki/NewLanguageSupport

GUI loaded the most frequent words and a language expert has manually transcribed a number of words, a Train and Predict button can be used to automatically train a simple letter-to-sound algorithm, based on decision trees, and predict pronunciations for the untranscribed words in the list. Furthermore, as a simplistic approximation of a part-of-speech tagger (POS), it is possible to mark function words in the list. This rudimentary POS tagger works based on simple function word annotations. Although the POS tagger works for many languages, for morphological languages like Telugu there is the possibility to plug in instead a rule-based POS tagger or a rule-based phonemiser. This is possible thanks to the modular architecture in the MARY TTS system.

With this minimal manual input for a new language, a simple NLP system can be built for a new language, using a generic tokeniser and a rule-based prediction of symbolic prosody.

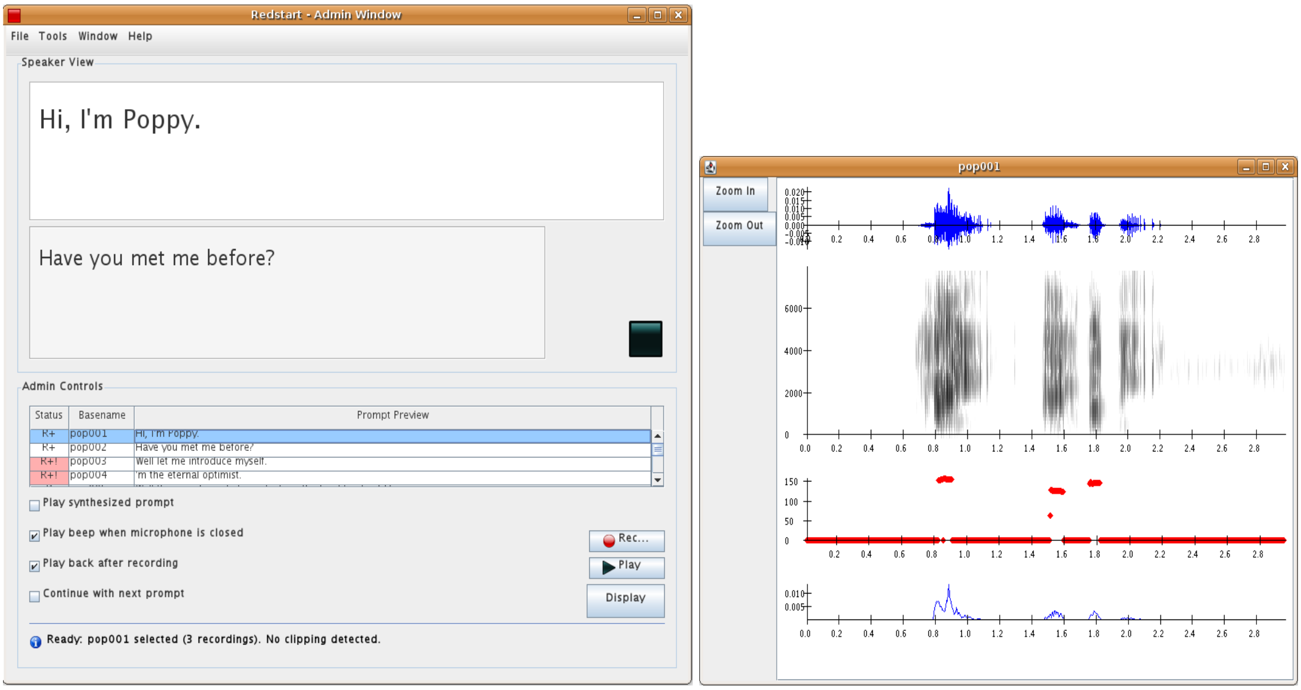
3.2. Voice-building process

Once the NLP component has been developed, or they are already available in MARY TTS (which is the case for English, German, Turkish and Telugu), the task of creating a voice can be pursued (right branch in Figure 1).

First, a recording script providing good diphone and prosodic coverage is selected from the text collection (Hunecke, 2007). Using the NLP components a Feature maker component annotates each sentence in the clean text database with diphone and prosody features to be used in a greedy selection. The resulting collection of sentences can be used as the recording script (selected sentences / script) for voice recordings with our tool Redstart. The recorded audio files can then be processed by our voice import tools which generate a unit selection and/or an HMM-based voice, as well as speaker-specific prediction components for acoustic parameters. If, during the voice-building process, force-aligned transcriptions were manually corrected, it is also possible to predict *speaker-specific pronunciations*. In the following these steps are explained in more detail.

3.2.1. Optimal text selection

Creating a recording script that provides a good diphone and prosodic coverage is not a trivial task. In the MARY voice creation toolkit an algorithm using greedy methods is used for selection of sentences optimising coverage. Three parameters are taken into account: the units, coverage definition and stop criteria. Units are defined as vectors consisting of three features: phone, next phone and prosody property. The definition of coverage fixes what kind of units are wanted in the final set, in the current version all diphones and their prosodic variation are used. Other aspects like frequency weights, sentence length, features weight, etc can be set for optimising the coverage. The stop criteria is a combination of number of sentences, maximum diphone coverage and maximum prosody coverage (Hu-



Hi, I'm Poppy.

Have you met me before?

Status	Basename	Prompt Preview
R+	pop001	Hi, I'm Poppy.
R+	pop002	Have you met me before?
R+!	pop003	Well let me introduce myself.
R+!	pop004	'm the eternal optimist.

☐ Play synthesized prompt

☑ Play beep when microphone is closed

☑ Play back after recording

☐ Continue with next prompt

Ready: pop001 selected (3 recordings). No clipping detected.

Figure 4: Redstart: MARY voice recording tool, more information about this tool can be found in: http://mary.opendfki.de/wiki/RedStart

necke, 2007). Additionally the selection of sentences using the previous algorithm can be combined with manual check. A synthesis script GUI, Figure 3, allows a user to check the sentences already selected, discard some (or all) and select and add more sentences. As said, the procedure start with the big wikipedia dump that includes all the articles available in a particular language. If the case is to design a domain specific TTS, at the moment, the only way to select sentences in a specific domain or jargon is manually through the SynthesisScriptGUI. However another possibility (not implemented yet) could be to restrict the Wikipedia articles to a particular topic like sports, chemistry, etc.

3.2.2. Voice recording: Redstart

The MARY Redstart is a voice recording tool that includes a GUI shown in the left part of Figure 4. Once the recording script is ready by the optimal text selection, the user can import the script into Redstart. The tool displays sentences one by one and the speaker can read the visible sentence on the Redstart screen after a beep. The user is not only able to play the recorded waveform, but also able to display the speech signal and the corresponding spectrogram, pitch, and energy contours, right part of Figure 4. The tool automatically prompts if it detects temporal clipping on the recording. So the user can record again the sample if he is not satisfied.

3.2.3. Voice import components

The `voice import tools` combines an extensible list of components in an intuitive Graphical User Interface (GUI), Figure 5. The GUI design is primarily intended to facilitate the creation of new voices by users without expert knowledge on speech synthesis. The voice import components normally execute high quality freely available components specialised in a particular task, for example for automatic labelling we can use Festvox EHMM or for training HMM models we use the scripts provided by HTS adapted to the MARY TTS architecture. The user does not have to care about the tool's configuration because the MARY voice building tool will suggest one (which the user can change if needed). In the following the voice import components are described in more detail.

Feature extraction components

For pitch extraction we use Praat (Boersma, 2001) and for pitch synchronous MFCC extraction we use the Edinburgh Speech tools (EST) (King et al., 2003).

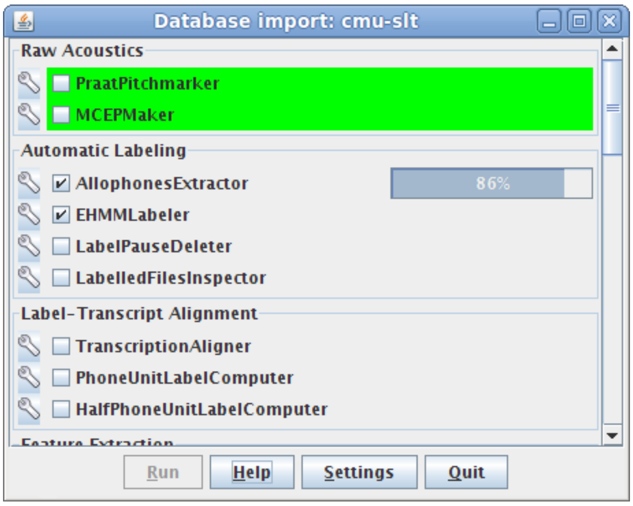
Automatic labelling components

For automatic labelling we can use the Festvox EHMM (Black and Lenzo, 2007) or the Sphinx (Huang et al., 1993; Walker et al., 2004) labelling tool.

Unit-selection synthesis components

Classification and Regression Tree (CART) models used by the unit selection run-time system are trained with the wagon tool from EST. The CARTDurationModeller[2] and the CARTF0Modeller components are used for building the *acoustic models for F0 and duration*. The MARY unit selection algorithm combines the usual steps of pre-selecting 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. In the current version, a very small pre-selection tree is manually specified and can pre-select units, e.g., by their phone or diphone identity (Schröder et al., 2009). A beam search is used in the dynamic programming step to keep processing time low. After successful execution of the unit

[2] http://mary.opendfki.de/wiki/VoiceImportComponents


Database import: cmu-slt

Raw Acoustics

- □ PraatPitchmarker
- □ MCEPMaker

Automatic Labeling

- ☑ AllophonesExtractor 86%
- ☑ EHMMLabeler
- □ LabelPauseDeleter
- □ LabelledFilesInspector

Label-Transcript Alignment

- □ TranscriptionAligner
- □ PhoneUnitLabelComputer
- □ HalfPhoneUnitLabelComputer

Feature Extraction

Figure 5: MARY voice import tool components, more information about this tool can be found in: http://mary.opendfki.de/wiki/VoiceImportComponents

selection voice building components, an installation component will copy all the necessary *unit selection voice files* of the new voice to the MARY TTS platform.

HMM-based synthesis components

For creating HMM-based voices we use a version of the speaker dependent training scripts provided by HTS (Zen et al., 2006) that was adapted to the MARY TTS platform. For the MARY TTS platform these training scripts have been modified to use (i) context features predicted by the MARY text analyser, (ii) include global variance and (iii) include features for generation of mixed excitation. As in the original HTS scripts we also use here the SPTK (Imai et al., 2009) and Snack (KTH, 2006) tools for extracting features with the HMMVoiceMakeData[3] component. In the same way, a HMMVoiceMakeVoice component use the HTK tool patched with the code provided by HTS to train HMMs. After successful execution of the HMM-based voice building components, an installation component will copy all the necessary *HMM-based voice files* of the new voice to the MARY TTS platform.

4. Quality control

In unit selection and HMM-based speech synthesis systems, accurate phonetic segmentation (labelling) is required to ensure quality of speech. The quality of labelling determines the quality of units, which might be affected by a range of problems including misaligned phone boundaries, mismatches between the phones that are labelled and that are pronounced, and the presence of background noise. Estimating the quality of individual units in the database is a key issue in order to reduce the amount of manual correction effort or as a criteria to apply when choosing a unit during synthesis. This toolkit provides a component to estimate the quality of labelling using a statistical model cost measure, comparing recorded phones to "average" acoustics as generated by an HMM synthesis model trained on the same data (Pammi et al., 2009). This component estimates quality of individual phonetic segments. When a human inspects the labels in the order given by this component, more errors can be found in a given time than with simple linear inspection.

5. Experience with the toolkit

The toolkit has been successfully applied to the creation, from scratch, of Turkish, Mandarin Chinese, British English and Telugu text-to-speech systems at DFKI. In the case of the creation of the Turkish voice the selection of sentences (1170 sentences, approximately 1 hour and 38 minutes of recording) and the semi-automatic transcription of word pronunciations (approximately 500 words) took around 2 weeks, the recordings were done in two days and the creation of a Turkish unit selection voice took between one and two days to a member of the MARY team. The creation of a Turkish HMM-based voice took approximately two days, also to a member of the MARY team.

The performance of the MARY unit selection voices in the 2009's Blizzard Challenge participation is significantly better than average and only two systems were performing better than MARY (Schröder et al., 2009). The toolkit has also served as the basis for the support of Mandarin Chinese in the Blizzard Challenge participation. For Mandarin Chinese, we have managed to create a relatively intelligible voice that sounds reasonably natural. This demonstrates the multilingual support of the MARY voice creation toolkit. We believe that having an up and running system there is a lot of room for improvement specially for experts on the language (none of the MARY team that developed this voice speaks any Mandarin Chinese).

The toolkit has also been used on the creation of voices for the MARY TTS platform in the languages already supported by MARY, English (US and British) and German. The unit selection and HMM-based voices available in MARY 4.0 are presented in Table 1.

The MARY voice creation toolkit was released with MARY TTS version 4.0 in December 2009. Since its release the number of downloads is already more than 6700. We have also registered more activity on the MARY users mailing list on questions regarding the creation of new voices. The number of registered members in the MARY users mailing list is around 150, several people are already using the voice creation toolkit.

6. Conclusions

We have presented a multilingual voice creation toolkit that supports the user in building voices for the open source MARY TTS platform, for two state of art speech synthesis technologies: unit selection and HMM-based synthesis. For languages not yet supported by MARY TTS, the toolkit provides the necessary tools and generic reusable run-time system modules for adding support for a new language.

The languages already supported by the toolkit are German, US English, GB English, Turkish and Telugu for which the latest version of MARY TTS (Version 4.0) provide unit selection and HMM-based voices freely available.

[3] http://mary.opendfki.de/wiki/HMMVoiceCreationMary4.0

[5] http://www.semaine-project.eu/

Language	Gen.	Name	Unit select.	HMM-based
German	M	bits3	X	X
	M	dfki-pavoque-neutral	X	
US English	F	cmu-slt	X	X
GB English	M	dfki-obadiah	X	
	F	dfki-poppy	X	
	F	dfki-prudence	X	
	M	dfki-spike	X	
Turkish	M	dfki-ot	X	X
Telugu	F	cmu-nk	X	

Table 1: Voices freely available in the MARY TTS system version 4.0. Gen.: gender (M) male, (F) female. The GB English voices are expressive voices build for the SEMAINE[5] project and all the other voices are normal/neutral speaking style.

The toolkit is mainly intended to be used by research groups on speech technology throughout the world, notably those who do not have their own pre-existing technology yet. We try to provide them with a reusable technology that lowers the entrance barrier for them, making it easier to get started. The whole process of creating a synthetic voice is fully documented in the MARY wiki pages and there is also the possibility to get support subscribing to the MARY mailing list.

Our experience with the toolkit has demonstrated that it enables rapid development of new voices with good quality. The MARY TTS results obtained in the latest Blizzard Challenge are encouraging.

Future improvements are planned on the level of individual component technologies. For example, signal processing techniques can be used to reduce the amount of audible concatenation artifacts in unit selection voices. More sophisticated prosody models may be able to improve the naturalness of the synthetic speech, especially for HMM-based voices. As core technologies, these improvements will benefit the quality of MARY voices independently of their language.

7. Acknowledgements

This work is supported by the DFG project PAVOQUE and the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 211486 (SEMAINE). The authors would like to thank the reviewers of this paper for their helpful comments.

8. References

A. W. Black and K Lenzo. 2007. Festvox: Building synthetic voices, Version 2.1. http://www.festvox.org/bsv/. (accessed March 2010).

P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345.

J. Duddington. 2010. eSpeak text to speech Version 1.43.12. http://espeak.sourceforge.net/. (accessed March 2010).

T. Dutoit, F. Bataille, V. Pagel, O. Pierret, and O. Van der Vreken. 1996. The MBROLA project: Towards a set of high-quality speech synthesizers free of use for non-commercial purposes. In *Proc. ICSLP*, Philadelphia, USA.

J. Hanika and P. Horák. 2005. The Epos speech system: User documentation Version 2.5.37. http://epos.ure.cas.cz/. (accessed March 2010).

D. Hill, L. Manzara, and C. Schock. 1995. Real-time articulatory speech-synthesis-by-rules. In *AVIOS 14th Annual International Voice technology Conference*, San Jose, CA, USA.

D. Hill. 2008. Gnuspeech: Articulatory Speech Synthesis. http://www.gnu.org/software/gnuspeech/. (accessed March 2010).

X. Huang, F. Alleva, H.W. Hon, M.Y. Hwang, K.F. Lee, and R. Rosenfeld. 1993. The SPHINX-II speech recognition system: an overview. *Computer Speech & Language*, 7(2):137–148.

A. Hunecke. 2007. Optimal design of a speech database for unit selection synthesis. Master's thesis, Fachrichtung 4.7 Allgemeine Linguistik, Universität des Saarlandes.

S. Imai, T. Kobayashi, K. Tokuda, T. Masuko, K. Koishida, S. Sako, and H. Zen. 2009. Speech signal processing toolkit (SPTK), Version 3.3. http://sp-tk.sourceforge.net. (accessed March 2010).

S. King, A. W. Black, P. Taylor, R. Caley, and R. Clark. 2003. The Edinburgh Speech Tools Library, system documentation edition 1.2, for 1.2.3. http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0/. (accessed March 2010).

KTH. 2006. The snack sound toolkit. http://www.speech.kth.se/snack. (accessed March 2010).

S. Pammi, M. Charfuelan, and M. Schröder. 2009. Quality control of automatic labelling using HMM-based synthesis. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing-Volume 00*, pages 4277–4280. IEEE Computer Society.

M. Schröder, M. Charfuelan, S. Pammi, and O. Türk. 2008. The MARY TTS entry in the Blizzard Challenge 2008. In *Proc. of the Blizzard Challenge 2008*.

M. Schröder, S. Pammi, and O. Türk. 2009. Multilingual MARY TTS participation in the Blizzard Challenge 2009. In *Proc. of the Blizzard Challenge 2009*.

W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. *Sphinx Whitepaper, Sun Microsystems INC.*

W. Walker, P. Lamere, and P. Kwok. 2005. FreeTTS 1.2. http://freetts.sourceforge.net/docs/index.php. (accessed March 2010).

H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda. 2006. The HMM-based speech synthesis system (HTS) Version 2.0. In *The 6th International Workshop on Speech Synthesis*.