An Open Source HMM-based Text-to-Speech System for Brazilian Portuguese

Igor Couto, Nelson Neto,
Vincent Tadaiesky and Aldebaro Klautau
Signal Processing Laboratory
Federal University of Pará
Belém, Brazil
{icouto,nelsonneto,vincent,aldebaro}@ufpa.br

Ranniery Maia
Toshiba Research Europe Limited
Cambridge Research Laboratory
Cambridge, UK
ranniery.maia@crl.toshiba.co.uk

Abstract— **Text-to-speech (TTS) is currently a mature technology that is used in many applications. Some modules of a TTS depend on the language and, while there are many public resources for English, the resources for some under-represented languages are still limited. This work describes the development of a complete TTS system for Brazilian Portuguese which expands the already available resources. The system uses the MARY framework and is based on the hidden Markov model (HMM) speech synthesis approach. Some of the contributions of this work consist in implementing syllabification, determination of stressed syllable and grapheme-to-phoneme (G2P) conversion. This work also describes the steps for organizing the developed resources and implementing a Brazilian Portuguese voice within the MARY. These resources are made available and facilitate the research in text analysis and HMM-based synthesis for Brazilian Portuguese.**

Keywords— ***Text-to-speech systems, HMM-based speech synthesis, text analysis.***

I. Introduction

Text-to-speech (TTS) systems are softwares that convert natural language text into synthesized speech [1]. The input text can be originated by numerous interfaces: user keyboard, scanner with character recognition (or OCR systems), etc. TTS is currently considered a more mature technology than speech recognition and has been used in many applications.

There are two guidelines for the faster dissemination of speech technologies in Brazilian Portuguese (BP):

- in the academy, to increase the synergy among research groups working in BP: availability of public domain resources for automatic speech recognition (ASR) and TTS. Both technologies are data-driven and depend on relatively large labeled corpora, which are needed for the development of state-of-art systems;
- in the software industry, to help programmers and entrepreneurs to develop speech-enabled systems: availability of *engines* (for ASR and TTS), preferably free and with licenses that promote commercialization, and tutorials and how-to's that target professionals without specific background in speech processing. In the latter case, the existence of application programming interfaces (API) is crucial because very few programmers have formal education in areas such as digital signal processing and HMMs.

In response to these two guidelines, the *FalaBrasil* project [2] was initiated in 2009. It aims at developing and deploying resources and software for BP speech processing. The public resources allow to establish baseline systems and reproducing results across different sites. With the increasing importance of reproducible research [3], the FalaBrasil project achieved good visibility and is now fomented by a very active open source community. Most of the currently available resources are for ASR, which include a complete large-vocabulary continuous speech recognition system. The current work is the first effort towards making available a TTS system. This work is also a natural follow-up of [4], which presented an HMM-based back end. Here, the emphasis is in some text analysis modules and the construction of a complete TTS system using the MARY framework [5].

This work is organized as follows. Section II provides a brief overview of some available TTS systems for BP and discusses the MARY, an open source platform that is used in this work. Section III describes the steps to implement the modules for BP TTS. Section IV discusses the developed resources and is followed by the conclusions.

II. TTS for Brazilian Portuguese

This section starts with a brief description of TTS in order to define the nomenclature. In the sequel previous research efforts and resources for BP TTS are summarized, followed by a short explanation of the MARY system.

A typical TTS system is composed by two parts: the front end and the back end. The front end is language dependent and performs text analysis to output information coded in a way that is convenient to the back end. For example, the front end performs text normalization, converting text containing symbols like numbers and abbreviations into the equivalent written-out words. It also implements the G2P conversion, which assigns phonetic transcriptions to each word and marks the text with prosodic information [6]. Phonetic transcriptions and prosody information together make up the (intermediate) symbolic linguistic representation that is output by the front end. Syllabification and syllable stress determination are also among the steps usually performed by the front end of a TTS. The back end is typically language independent and includes the synthesizer, which is the block that effectively generates sound. Fig. 1 depicts a simple functional diagram of a TTS system.

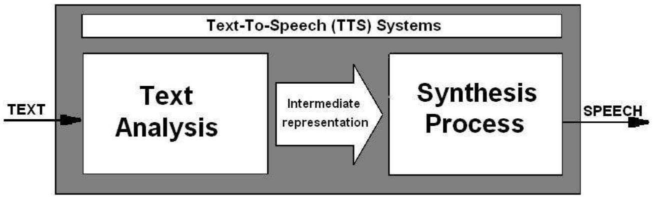


Fig. 1. Functional diagram of a TTS system showing the front and back ends, responsible by the text analysis and synthesizer, respectively

TTS systems evolved from a *knowledge-based* paradigm to a pragmatic *data-driven* approach. With respect to the technique adopted for the back end, the main categories are the formant-based, concatenative and, more recently, HMM-based [1], [7], [8]. With respect to the API, the most widely used in the industry is SAPI, the speech API from Microsoft [9]. There are other alternatives such as JSAPI (Java SAPI) from Sun Inc.

In the academy, the first complete TTS systems for BP emerged in the end of the Nineties at *Universidade de Campinas* (UNICAMP) and *Pontifícia Universidade Católica do Rio de Janeiro* (PUC-RJ), with formant-based synthesizers [10] and concatenative (diphone) synthesis [11]–[14]. Currently, informal listening tests indicate that the most mature BP TTS system is the one developed at *Universidade Federal de Santa Catarina* (UFSC) [15]. In [4], an HMM-based back end for BP was presented and the authors made available the recorded speech corpus and HTS training scripts [16].

In the speech industry, there are some companies that offer BP voices for use in specific engines. Among them one can find Raquel of Nuance [17], Fernanda, Gabriela and Felipe of Loquendo [18], and Marcia, Paola, and Carlos of Acapela [19]. Microsoft also has support for BP in the mobile and desktop platforms [20].

With the exception of [4], the previously mentioned systems were not made available, for example, for research purposes. Focusing in open source code and/or public resources, the BP engine of the MBROLA project [21] and the DOSVOX system [22] should be noted. DOSVOX corresponds to a free operating system for the visually impaired and includes its own speech synthesizer, besides offering the possibility of using other engines. Another relevant resource is the CSLU *tookit* [23], which supports BP via the diphone-based AGA voice.

A. MARY framework

The motivation for using the MARY in this work was that it is completely written in Java and supports both concatenative and HMM-based synthesis. MARY stands for "modular architecture for research on speech synthesis" and it is a modern open source framework for TTS [24]. As indicated by the name, MARY is designed to be highly modular, with a special focus on transparency and accessibility of intermediate processing steps. Currently, MARY supports German, English and Tibetan languages. The highly modular design allows one to insert new languages and create new voices. The platform aims to be a very flexible tool for research, development and teaching of text-to-speech synthesis. Internally, MARY uses an XML-based representation language, called MaryXML, to represent information inside the system. The adoption of XML facilitates the integration of modules from different origins [25].

MARY provides support to the client-server architecture. Hence, a TTS system is decomposed in a server application, which contains the components to make the synthesis, and a client application, which makes requests for the server to execute some task. A server can support many languages and waits requests from one or several clients in a port specified by user. A set of configuration files, read at system startup, defines the processing components to be used [26].

In order to create a BP TTS, some specific resources had to be developed and procedures were executed, following the tutorials in [5]. One of the final results was a file called pt_BR.config, which defined the BP processing module and is available at [2]. The next two sections describe the developed resources and adopted procedures.

III. Building a BP MARY TTS

Using the nomenclature adopted in the MARY framework, the task of supporting a new language can be split into the creation of a text processing module and the voice. The former enables the software to process BP text and, for example, perform the G2P conversion. The creation of a voice in this case corresponds to training HMMs using the HTS toolkit [16]. For HMM training, one needs a labeled corpus with transcribed speech. This work used the speech data available with the BP demo at the HTS site [16]. This is the audio data used in [4] and has a total of 221 files, corresponding to approximately 20 minutes of speech. The transcriptions were not found at the HTS site and, for convenience to other users, were made available in electronic format at [2]. Other speech corpora can be used, with the restriction that each audio file must have its corresponding orthographic transcription (at the word level).

A. Front end: support for Brazilian Portuguese

MARY requires a set of files for each language that it supports. A block diagram of the steps that were followed to create these files for BP is presented in Fig. 2. The philosophy adopted by MARY is to provide support for the creation of a basic data-driven text pre-processing module. Alternatively, one can write its own modules, using more sophisticated techniques. This work adopted the recipe suggested by MARY for training finite state transducers (FST) [27], as described in the sequel.

As indicated in Fig. 2, the FST training procedure requires the definition of a phonetic alphabet for PB and a dictionary, with preferably all the words that should be supported. These two files are briefly described below:

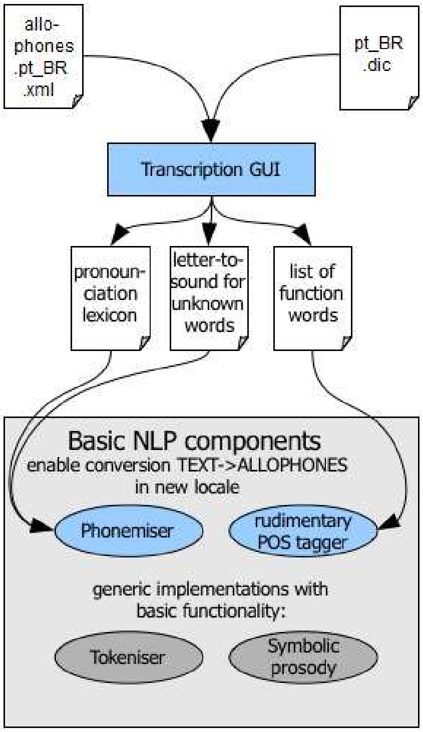


Fig. 2. Block diagram of the steps for creating the BP version of MARY. Adapted from [28].

1) allophones.pt_BR.xml: is the phonetic alphabet, which must describe the *distinctive features* [29] of each phone, such as voiced/unvoiced, vowel/consonant, tongue height, etc. A preliminary version of this file was developed by the authors using SAMPA [30] and is listed in Appendix A.

2) pt_BR.dic: is a dictionary containing all the planned words (the system will be able to synthesize out-of-vocabulary words too) with their corresponding phonetic transcriptions (based on the phonetic alphabet previously described). These transcriptions are required to be separated into syllables and the stress syllable indicated.

After creating these two files, still according to Fig. 2, the *Transcription GUI* tool reads both files and creates two FST. The first FST is responsible for converting graphemes into phonemes. The second FST is a rudimentary part-of-speech tagger and, for the developed system, tries only to distinguish functional and non-functional words. Both FSTs are based on classification and regression trees (CART) and more details can be found in [31].

At the end of the process, four files were created and compose the text pre-processing module for BP:

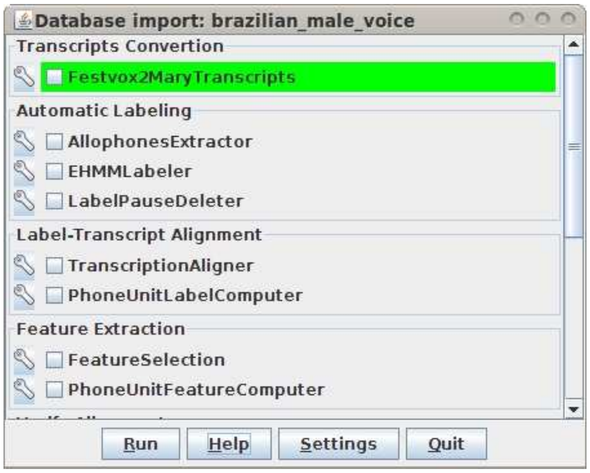
- pt_BR_lexicon.fst - grapheme-to-phoneme FST
- pt_BR.lst - letter to sound for unknown words
- pt_BR_pos.fst - functional words FST
- pt_BR_lexicon.dict - dictionary.

MARY is modular enough to allow bypassing the TranscriptionTool and allowing the user to generate some or all of the files above with other tools. But TranscriptionTool is very convenient and effective for applications such as teaching TTS techniques.

After having the four files, one has to specify their location in a configuration file, called pt_BR.config in our case. This file is parsed where MARY starts and enables the PB language. A complete model is provided in [5].

B. Back end: HMM-based voice creation

After having a valid BP front end, the HTS toolkit was used to create an HMM-based back end. This task can be divided in preparing the files and, after that, the HMM training itself.


Database import: brazilian_male_voice

Transcripts Convertion
□ Festvox2MaryTranscripts

Automatic Labeling
□ AllophonesExtractor
□ EHMMLabeler
□ LabelPauseDeleter

Label-Transcript Alignment
□ TranscriptionAligner
□ PhoneUnitLabelComputer

Feature Extraction
□ FeatureSelection
□ PhoneUnitFeatureComputer

Fig. 3. The GUI of the VoiceImport tool.

In order to facilitate the procedure, MARY provides the VoiceImport tool, which is a friendly interface. Fig. 3 shows a screenshot of the tool. The major advantage of this tool is that it encapsulates several commands that otherwise the user would be required to type on a console. For example, the HMMVoiceDataPreparation component verifies if all the needed programs are properly installed and can even install them, freeing the user from some manual labor. Using MARY the user can select the needed component and activate its execution by clicking a button at the provided GUI.

Because MARY supports concatenative and HMM-based synthesis, VoiceImport has routines for both. Hence, for the creation of the BP TTS not all components of the VoiceImport tool are needed. The effectively used components were the following. For the subtask of file preparation: Festvox2MaryTranscriptions, HMMVoiceDataPreparation, AllophonesExtractor, EHMMLabeller, LabelPauseDeleter, TranscriptionAligner, PhoneUnitLabelComputer, FeatureSelection, PhoneUnitFeatureComputer and PhoneLabelFeatureAligner. For the subtask of HMM training: HMMVoiceConfigure, HMMFeatureSelection, HMMVoiceMakeData, HMMVoiceMakeVoice and HMMVoiceInstaller. All these components are documented in [5].

The last component, HMMVoiceInstaller, installs the PB language support, copying the necessary files to the correct locations. After this stage the PB TTS is already supported by the MARY server.

IV. Developed resources

In spite of the possibility of using the TranscriptionTool component, the task of creating the input files can be demanding. Therefore, one of the developed resources was a Java software called TextAnalysis4BP, which is capable of creating the input file pt_BR.dic requested by TranscriptionTool. In the sequel, each of the modules of TextAnalysis4BP is described, namely: G2P converter, syllable separator and stress syllable indicator.

A. G2P conversion

In [32], the authors had described a G2P converter for BP, based on a set of rules described in [33]. One advantage of rule-based G2P converters when compared to classifiers as decision trees is that the lexical alignment is not necessary, since the software must not be trained to generate their own rules. In other words, the proposed conversion, based on pre-established phonological criteria, are supplied to the system according to the language which the application is intended. Its architecture does not rely on intermediate stages, i.e., other algorithms such as syllabic division or plural identification. There is a set of rules for each grapheme and a specific order of application is assumed. First, the more specific rules are considered until a general case rule, that ends the process. No co-articulation analysis between words was performed, and the G2P converter [32] dealt only with single words.

B. Syllabification

The phonetic dictionary generated by the G2P converter described in [32] does not perform syllabification nor stress syllable identification. These two tasks were developed in this work because they are required by MARY. The algorithm used for syllabification is described in [34]. The main idea of this algorithm is that all syllables have a vowel as a nucleus, and it can be surrounded by consonants or other (semi-)vowels. Hence, one should locate the vowels that composes the syllable nuclei and isolate consonants and semivowels.

To test the algorithm a corpus of 139,751 words of BP was obtained from a web site. A manual analysis of the obtained results indicated that the algorithm in [34] confuses hiatus. Therefore the original algorithm was modified as follows:

1) The original algorithm used to considered the structure "*i* or *u* + Vowel", found in the middle of the word, were considered rising diphthongs and were not separated, resulting in wrong syllabic division such as hiatus. It was corrected by changing the condition, so that hiatus are considered by the algorithm.
2) The original algorithm considered the structure "*i* or *u* + Vowel" at the end of words as hiatus and separated, which leads to errors for words such as "ódio" that is separated as "ó-di-o". The correction takes in account whether or not there is an accent mark in the word.

Even with the improvements, errors were still observed in words such as "traidor" (separated as "trai-dor" instead of "tra-i-dor"). In other words, the syllable separator fails with false diphthongs, which are those the present two possible pronunciations, as hiatus and diphthongs.

C. Syllable stress determination

Identifying the stress syllable proved to be an easier task that benefited from the fact that the G2P converter [32], in spite of not separating in syllable, was already able to identify the stressed vowel. After getting the result of the syllabification, it was then trivial to identify the syllable corresponding to the stress vowel.

V. Conclusions

This work presented the current status of the on-going project that consists in developing a state-of-art HMM-based TTS for BP. MARY, the adopted framework, has proved to be very flexible and relatively easy to use. These characteristics facilitate the use of the developed TTS system in the academy, for example, in speech processing classes. More advanced text analysis modules must be developed for achieving improved naturalness and overall quality. However, the strategy is to emphasize the creation of necessary resources even if they are not the ideal ones in terms of coverage, for example. This way the community can gradually improve aspects such asthe recorded speech corpus and prosody module.

Future works include implementing recently published algorithms for syllabification and stress determination for BP with high performance [35], interfacing a more advanced POS tagging module to MARY and improving the performance by substituting the files provided TranscriptionTool by others obtained with more accurate text pre-processing algorithms.

Acknowledgements

This work was partially supported by CNPq, Brazil, project 482148/2009-8 *Desenvolvimento de Recursos para Reconhecimento de Voz e Fomento à Indústria Brasileira de Softwares com Interface Aural.*

Appendix A
Phoneset XML file

The file below contains the distinctive features that were organized for the BP version of MARY. The phone set is SAMPA [30], which is also used by MARY for other languages.

```
<allophones name=”sampa” xml:lang=”pt–BR” features=”vlng␣
     vheight␣vfront␣vrnd␣ctype␣cplace␣cvox”>

 <silence ph=”_”/>
 <!–– Oral vowels––>
 <vowel ph=”a” vlng=”s” vheight=”3” vfront=”1” vrnd=”–”/>
 <vowel ph=”E” vlng=”s” vheight=”2” vfront=”1” vrnd=”–”/>
 <vowel ph=”e” vlng=”s” vheight=”2” vfront=”1” vrnd=”–”/>
 <vowel ph=”i” vlng=”s” vheight=”1” vfront=”1” vrnd=”–”/>
 <vowel ph=”O” vlng=”s” vheight=”2” vfront=”3” vrnd=”+”/>
 <vowel ph=”o” vlng=”s” vheight=”2” vfront=”3” vrnd=”+”/>
 <vowel ph=”u” vlng=”s” vheight=”1” vfront=”3” vrnd=”+”/>
 <!-- Nasal vowels-->
 <vowel ph=”a˜” vlng=”l” vheight=”3” vfront=”2” vrnd=”–”/
      >
 <vowel ph=”e˜” vlng=”l” vheight=”2” vfront=”2” vrnd=”–”/
      >
 <vowel ph=”i˜” vlng=”l” vheight=”1” vfront=”2” vrnd=”–”/
      >
 <vowel ph=”o˜” vlng=”l” vheight=”2” vfront=”3” vrnd=”+”/
      >
 <vowel ph=”u˜” vlng=”l” vheight=”1” vfront=”3” vrnd=”–”/
      >
 <!-- Semi-vowels-->
 <vowel ph=”w” vlng=”d” vheight=”2” vfront=”2” vrnd=”0”
      ctype=”v”/>
 <vowel ph=”j” vlng=”d” vheight=”2” vfront=”2” vrnd=”0”
      ctype=”p”/>
 <vowel ph=”w˜” vlng=”d” vheight=”2” vfront=”2” vrnd=”0”
      ctype=”v”/>
 <vowel ph=”j˜” vlng=”d” vheight=”2” vfront=”2” vrnd=”0”
      ctype=”p”/>
 <!-- Unvoiced fricatives-->
 <consonant ph=”f” ctype=”f” cplace=”b” cvox=”–”/>
 <consonant ph=”s” ctype=”f” cplace=”a” cvox=”–”/>
 <consonant ph=”S” ctype=”f” cplace=”p” cvox=”–”/>
 <!-- Voiced fricatives-->
 <consonant ph=”z” ctype=”f” cplace=”a” cvox=”+”/>
 <consonant ph=”v” ctype=”f” cplace=”b” cvox=”+”/>
 <consonant ph=”Z” ctype=”f” cplace=”p” cvox=”+”/>
 <!-- Affricatives-->
 <consonant ph=”tS” ctype=”a” cplace=”p” cvox=”–”/>
 <consonant ph=”dZ” ctype=”a” cplace=”p” cvox=”+”/>
 <!-- Plosives-->
 <consonant ph=”b” ctype=”s” cplace=”l” cvox=”+”/>
 <consonant ph=”d” ctype=”s” cplace=”l” cvox=”+”/>
 <consonant ph=”t” ctype=”s” cplace=”d” cvox=”–”/>
 <consonant ph=”k” ctype=”s” cplace=”v” cvox=”–”/>
 <consonant ph=”g” ctype=”s” cplace=”v” cvox=”+”/>
 <consonant ph=”p” ctype=”s” cplace=”l” cvox=”–”/>
 <!-- Liquids -->
 <consonant ph=”l” ctype=”l” cplace=”a” cvox=”+”/>
 <consonant ph=”L” ctype=”l” cplace=”p” cvox=”+”/>
 <consonant ph=”R” ctype=”l” cplace=”a” cvox=”+”/>
 <consonant ph=”X” ctype=”l” cplace=”a” cvox=”+”/>
 <consonant ph=”r” ctype=”l” cplace=”a” cvox=”+”/>
 <!-- Nasal consoants -->
 <consonant ph=”m” ctype=”n” cplace=”l” cvox=”+”/>
 <consonant ph=”n” ctype=”n” cplace=”a” cvox=”+”/>
 <consonant ph=”J” ctype=”n” cplace=”p” cvox=”+”/>
</allophones>
```

REFERENCES

[1] J. Allen, M. S. Hunnicutt, D. H. Klatt, R. C. Armstrong, and D. B. Pisoni, *From text to speech: The MITalk system*. Cambridge University Press, 1987.

[2] “http://www.laps.ufpa.br/falabrasil,” Visited in May, 2010.

[3] P. Vandewalle, J. Kovacevic, and M. Vetterli, “Reproducible research in signal processing - what, why, and how,” *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 37–47, 2009.

[4] R. Maia, H. Zen, K. Tokuda, T. Kitamura, and F. Resende, “An HMM-based Brazilian Portuguese speech synthetiser and its characteristics,” *Journal of Communication and Information Systems, v. 21, p. 58-71*, 2006.

[5] “http://mary.opendfki.de/,” Visited in May, 2010.

[6] J. van Santen, J. Hirschberg, J. Olive, and R. Sproat, Eds., *Progress in Speech Synthesis*. New York: Springer-Verlag, 1996.

[7] T. Dutoit, *An Introduction to Text-To-Speech Synthesis*. Kluwer, 2001.

[8] R. I. Damper, *Data-Driven Methods in Speech synthesis*, 2001.

[9] “http://www.microsoft.com/speech/,” Visited in March, 2010.

[10] L. De C.T. Gomes, E. Nagle, and J. Chiquito, “Text-to-speech conversion system for Brazilian Portuguese using a formant-based synthesis technique,” in *SBT/IEEE International Telecommunications Symposium*, 1998, pp. 219–224.

[11] J. Solewicz, A. Alcaim, and J. Moraes, “Text-to-speech system for Brazilian Portuguese using a reduced set of synthesis units,” in *ISSIPNN*, 1994, pp. 579–582.

[12] F. Egashira and F. Violaro, “Conversor Texto-Fala para a Língua Portuguesa,” in *13o Simposio Brasileiro de Telecomunicacoes*, 1995, pp. 71–76.

[13] E. Albano and P. Aquino, “Linguistic criteria for building and recording units for concatenative speech synthesis in Brazilian Portuguese,” in *in Proceedings EuroSpeech, Rhodes, Grecia*, 1997, pp. 725–728.

[14] P. Barbosa, F. Violaro, E. Albano, F. Simes, P. Aquino, S. Madureira, and E. Franozo, “Aiuruete: a high-quality concatenative text-to-speech system for Brazilian Portuguese with demisyllabic analysis-based units and hierarchical model of rhythm production,” in *Proceedings of the Eurospeech'99, Budapest, Hungary*, 1999, pp. 2059–2062.

[15] I. Seara, M. Nicodem, R. Seara, and R. S. Junior, “Classificação Sintagmática Focalizando a Síntese de Fala: Regras para o Português Brasileiro,” in *SBrT*, 2007, pp. 1–6.

[16] “http://hts.ics.nitech.ac.jp/,” Visited in May, 2010.

[17] “http://www.nuance.com/realspeak/languages/,” Visited in May, 2010.

[18] “http://www.loquendo.com/en/demos/demo_tts.htm,” Visited in May, 2010.

[19] “http://www.acapela-group.com/portuguese-brazil-46-text-to-voice.html,” Visited in May, 2010.

[20] D. Braga, P. Silva, M. Ribeiro, M. S. Dias, F. Campillo, and C. García-Mateo, “Hélia, Heloísa and Helena: new HTS systems in European Portuguese,Brazilian Portuguese and Galician,” in *PROPOR 2010 - International Conference on Computational Processing of the Portuguese Language*, 2010, pp. 27–30.

[21] “http://tcts.fpms.ac.be/synthesis,” Visited in May, 2005.

[22] “http://intervox.nce.ufrj.br/dosvox/,” Visited in March, 2005.

[23] “http://cslu.cse.ogi.edu/toolkit/,” Visited in March, 2010.

[24] M. Schröder and J. Trouvain, “The German Text-to-Speech synthesis system MARY: A tool for research, development and teaching,” in *International Journal of Speech Technology*, 2001, pp. 365–377.

[25] M. Schröder and S. Breuer, “XML Representation Languages as a Way of Interconnecting TTS Modules,” in *in Proc. ICSLP, Jeju, Korea*, 2004.

[26] M. Schröder and A. Hunecke, “MARY TTS participation in the Blizzard Challenge 2007,” 2007.

[27] A. Kornai, Ed., *Extended Finite State Models of Language*. Cambridge University Press, 1999.

[28] “http://mary.opendfki.de/wiki/newlanguagesupport,” Visited in May, 2010.

[29] R. Jakobson, C. G. M. Fant, and M. Halle, *Preliminaries to speech analysis: the distinctive features and their correlates*. Cambridge : Acoustics Laboratory, Massachusetts Institute of Technology, 1952.

[30] “http://www.phon.ucl.ac.uk/home/sampa/home.htm,” Visited in May, 2010.

[31] “http://mary.opendfki.de/wiki/transcriptiontool,” Visited in May, 2010.

[32] A. Siravenha, N. Neto, V. Macedo, and A. Klautau, “Uso de Regras Fonológicas com Determinação de Vogal Tônica para Conversão Grafema-Fone em Português Brasileiro,” *7th International Information and Telecommunication Technologies Symposium*, 2008.

[33] D. C. Silva, A. de Lima, R. Maia, D. Braga, J. F. de Morais, J. A. de Morais, and F. G. V. R. Jr., “A rule-based grapheme-phone converter and stress determination for Brazilian Portuguese natural language processing,” in *Proc. of IEEE Int. Telecomm. Symposium (ITS)*, 2006.

[34] L. Gomes, “Sistema de conversão texto-fala para a língua portuguesa utilizando a abordagem de síntese por regras,” Master's thesis, Universidade Estadual de Campinas, 1998.

[35] D. C. Silva, D. Braga, and F. G. V. R. Jr, “Separação das Sílabas e Determinação da Tonicidade no Português Brasileiro,” *XXVI Simpósio Brasileiro de Telecomunicações (SBrT'08)*, 2008.