A BRAZILIAN PORTUGUESE TTS BASED ON HMMS

Guilherme de Oliveira Pinto, Filipe Leandro de F. Barbosa, Fernando Gil V. Resende Jr.

Programa de Engenharia Elétrica/COPPE, DEL/EE
Universidade Federal do Rio de Janeiro
C.P. 68504, 21945-970, Rio de Janeiro, RJ, Brasil
Emails: {guilherme, filipe, gil} @lps.ufrj.br

Abstract - *This paper presents a Brazilian Portuguese TTS based on HMMs, which uses mel-cepstral coefficients as parameters of speech. We implemented an algorithm which performs a phoneme-based transcription to the Portuguese spoken in Rio de Janeiro. For a given text to be synthesized, after the phoneme transcription, static features are extracted from a sentence HMM, generated by concatenating HMMs sub-word units. An algorithm for choosing the best set of those speech units was also developed. Subjective tests show that the proposed TTS gives better results than a PSOLA based on syllabic units, with the advantage of easy speaker adaptation.*

1. INTRODUCTION

A Text-to-Speech (TTS) synthesizer is a system that should be able to read any desired text with some degree of intelligibility.

The hidden Markov model (HMMs) is a probabilistic concept which is used to characterize sequences of patterns. It has been used in state of the art technologies for speech recognition and synthesis systems [1]. For instance, it models sequences of speech spectra.

In contrast to PSOLA's (Pitch Synchronous Overlap and Add) approach, for example, with HMM-based synthesis systems it is feasible to change voice characteristics to a target speaker by changing spectral parameters of speech. With only ten spoken sentences, an HMM-based system can generate synthesized speech similar to any speaker [2].

In this work a Brazilian Portuguese TTS based on HMMs is described. A high level description of the whole system is presented. Towards a better understanding of HMM-based speech synthesis, the complete implementation is divided in two well-defined sections: the training and the synthesis part [3].

In the training part, the HMM models can be created by HTK toolkit [4] when the following items are available:

- A set of the most representative speech units.
- Phonetic transcription of the data base.
- A parametrised data base.

In the synthesis part, before using SPTK toolkit [5], the data below must be provided:

- A sentence HMM.
- Pitch information.

The way how we obtain those five itemized files is shown through the sections along this paper.

2. TTS BASED ON HMMS

HMMs have successfully been applied in speech recognition and synthesis systems. They are able to characterize pitch, duration and overall spectra information [6].

The first step in HMM-based synthesis is data parametrization. Each continuous waveform of the speech data base is converted to a set of equally spaced discrete-time parameter vectors. Besides the parametrization, a phone level transcription of all the data base sentences is also needed to perform the HMM training, as shown in Figure 1.

In this paper we discuss continuous HMMs. Each state can be defined by a mean vector, a diagonal covariance matrix and a transition matrix. Each element $a_{i,j}$ of this matrix represents the probability of having a *i* to *j* state transition. The process of calculating those triples (means, covariance matrix, transition matrix) for every unit is called HMM training.

At this point speech parameter vectors are generated from a sentence HMM, wich is constructed by concatenating HMM units. Thus by using a specific digital filter for those coefficients, speech can be synthesized. Details of these operations are presented in the next section.

3. THE IMPLEMENTED HMM-BASED BRAZILIAN PORTUGUESE TTS

3.1. The Phone Level Transcription

To start the process of synthesizing speech, a context dependent phone level transcription is needed, i.e., the same

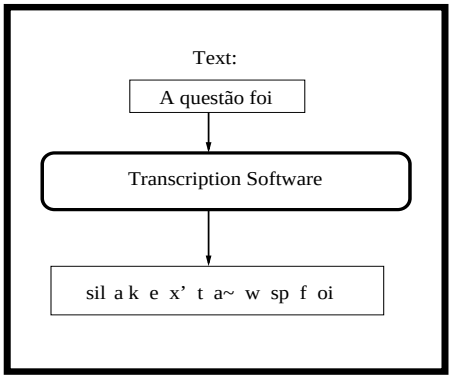

Fig. 1. Phone Level transcription of a given text.

Table 1. Our phone list with exemples.

/a/	f**a**lha	/A/	c**a´**lice	/a /	avi**a˜**o
/b/	**b**ala	/k/	**c**aseiro	/d/	**d**antesco
/e/	aclive	/E/	h**e´**lio	/e /	**en**tediado
/f/	**f**aca	/g/	**g**alo	/h/	**h**oje
/i/	**e**studo	/j/	**g**eleía	/l/	**l**eite
/lh/	a**lh**o	/w/	unia˜**o**	/m/	**m**elhor
/nh/	ni**nh**o	/n/	**n**osso	/o/	res**o**lver
/o /	h**o**mem	/O/	**o´**timo	/p/	**p**ato
/r/	ca**r**o	/R/	ca**rr**o	/R'/	ca**r**ga
/r'/	ca**r**ta	/s/	**s**apato	/t/	**t**ato
/u/	log**o**	/u /	**u**m	/v/	**v**oltar
/x/	a**ch**ar	/x'/	cas**t**a	/j'/	a**s**ma
/z/	ca**s**a	/ai/	**ai**pim	/au/	**au**la
/ua/	q**ua**ndo	/ei/	l**ei**te	/Ei/	id**e´**ia
/oi/	**oi**to	/Oi/	her**oí**	/ou/	**ou**tro
/eu/	m**eu**	/ui/	m**ui**to		

word can have different phoneme transcriptions. Using 85 Rio de Janeiro Portuguese rules [7], a algorithm was implemented.Table 1 shows our phone set.

Figure 1 shows an example of a context dependent transcription. The text entry is: "a questão foi", which means "the question was". It is imporant to notice that there are three silence models: *sil*, *lp*, *sp*. The *sil* model is used in the beginning and ending of sentences. For the long pauses, such as commas, we use the *lp* model, otherwise, the short pause model, *sp*, is used.

3.2. Unit choosing method

The choice of the unit model has a fabulous importance on improving the quality of synthesized speech. Our system uses triphones as the main speech units. When synthesizing speech, one important part to cover is the co-articulation between phonemes, words, or even phrases. However, increasing the unit model, by choosing words as speech units

Table 2. Table of units occurences using CETEM-PUBLICO text data base.

CETEM PUBLICO	Distinct Units	Most Frequently Units 1^{st}	2^{nd}	3^{rd}
Monophones	51	sp 18.1%	a 9.04%	i 8.43%
Diphones	2131	a-sp 3.68%	u-sp 3.29%	i-sp 2.91%
Triphones	44852	sp-d+i 1.12%	d-i+sp 1.09%	a -w+sp 0.91%

Table 3. Table of units occurences using our speech data base.

OUR DATA BASE	Distinct Units	Most Frequently Units 1^{st}	2^{nd}	3^{rd}
Monophones	51	sp 15.78%	a 9.14%	i 8.42%
Diphones	612	a-sp 3.32%	u-sp 3.30%	i-sp 2.93%
Triphones	2159	sp-d+i 1.16%	d-i+sp 1.14%	a -w+sp 0.80%

for example, creates a problem of having a large amount of unseen units, during the synthesis process. By choosing triphones, we get a balance between the amount of units and the co-articulation feature.

An algorithm for choosing the best set of speech units was implemented. Initially, we obtained a table(2) estimating the triphones, diphones and monophones most frequently used in our language. To do this research we use a Portuguese data base, which has close to 900 million words. If those units HMM are not well-trained using our speech data base, we will not succeed on synthesizing speech.

A reasonable amount of speech data is necessary to generate well-trained HMMs. Fortunately, applying the transcription algorithm to our speech data base we found that the most common units extracted from the Portuguese text data base were well represented.Table 2 and table 3 show parts of the text and speech data base occurences, respectively. The best units chosen are those which appear more than a established threshold in our speech data base. All the triphones that had one or two occurences are not considered to be a good model and should be better to use diphones and monophones models instead. For the diphones, we have chosen all that appeared at least 6 times in our speech data base. To synthesize any text we have to use all data base monophones. The total phoneme number is 51, including silence models.

3.3. Data Preparation

Firstly, using the transcription. For a 16kHz speech data base with 200 phonetically balanced unlabeled utterances. Data from one speaker is used. Using the transcription software, all the sentences are transcribed.

After the phone level transcription, for each utterance on the speech data base, mel-cepstral coefficients are obtained. Mep Cepstral analysis [8] of speech waveforms is provided by SPTK. The signal is windowed by a 32ms Hamming window with a 5ms shift.

Speech spectrum $H(e^{jw})$ is characterized by M^{th} order mel-cepstral coefficients $c_\alpha(m)$, as follows:

$$H(z) = exp \sum_{m=0}^{M} c_\alpha(m)\tilde{z}^{-m} \quad (1)$$

under the constraint

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \ \ |\alpha| < 1. \quad (2)$$

The phase is modeled by the variable α. For a sampling rate 16kHz, choosing α = 0.42 a good aproximation to the mel frequency scale is achieved.

$H(z)$ can be rewritten as,

$$H(z) = K \cdot D(z), \ \ K : const. \quad (3)$$

Mel-cepstral coefficients can be calculated as follows:

$$\epsilon = min_{D(z)} E[e^2(n)] \quad (4)$$

where $e(n)$ is the output of the inverse filter $\frac{1}{D(z)}$.

Our system works with 25 mel-cepstral coefficients, since we set M=24 and the 0^{th} coefficient is included.

3.4. HMMs training

In the HMM training part, [4] we use mel-cepstral static coefficents to train the phoneme HMMs. All HMMs used are 5-state left to right models with no skips. Each state of every model has its own mean and variance. At first, a global mean and variance is calculated and set all the Gaussians on every monophone HMM. After the Baum-Welch embedded training reestimation algorithm, the phoneme training is then finished.

Cloning the trained monophones HMM will produce diphone models for every distinct diphone in the training database. These models are then reestimated by the embedded training. During this procedure, all the diphones that have equal central phonemes will share the same transition matrix. The same cloning process is done to produce triphone models. The phoneme training and the cloning procedure are provided by HTK toolkit.

Having all the HMMs trained, it is necessary to prepare the units models to the HMM sentence. An algorithm is implemented to calculate, for each triphone, diphone and monophone model, the state duration $sd(j)$ of every state j in a given unit model. This can be done by using the expresion [9]:

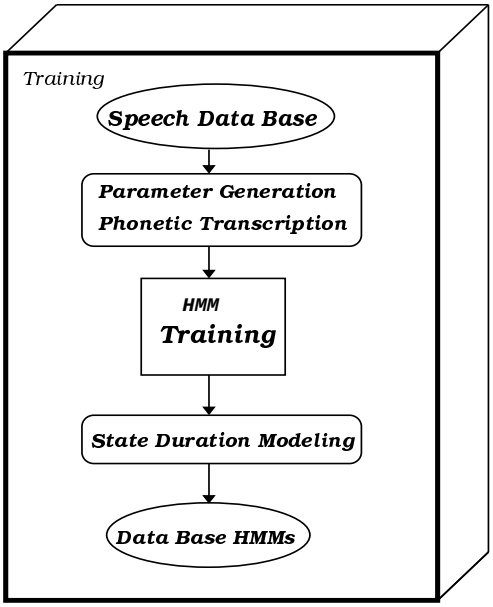

Fig. 2. The training part.

$$sd(j) = \frac{1}{1 - a_{j,j}}, \quad (5)$$

where $a_{j,j}$ represents the transition matrix element.

3.5. Parameter Generation

From a given HMM, λ, a speech parameter vector sequence is extracted [10] [9] [3].

$$O = [o_1, o_2, o_3, ..., o_T] \quad (6)$$

In such a way that

$$logP[Q, O|\lambda] = \alpha \sum_{k=1}^{K} logp_{q_k}(d_{q_k}) + \sum_{t=1}^{T} logc_{q_t,i_t} - \frac{1}{2}(O - \mu)'U^{-1}(O - \mu) - log|U| - \frac{3MT}{2} log2\pi \quad (7)$$

is maximized when the speech parameter vetor sequence is equal to the mean vectors, i.e. , O = μ , where

$$\mu = [\mu_{q_1,i_1}, \mu_{q_2,i_2}, \ldots, \mu_{q_T,i_T},] \quad (8)$$

$$U = [U_{q_1,i_1}, U_{q_2,i_2}, \ldots, U_{q_T,i_T},] \quad (9)$$

and $c_{q,i}$, $\mu_{q_t i_t}$ and Uq_t, i_t are the mixture weight, the mean vector and the covariance matrix respectively. K is the total of states that have been visited during T frames

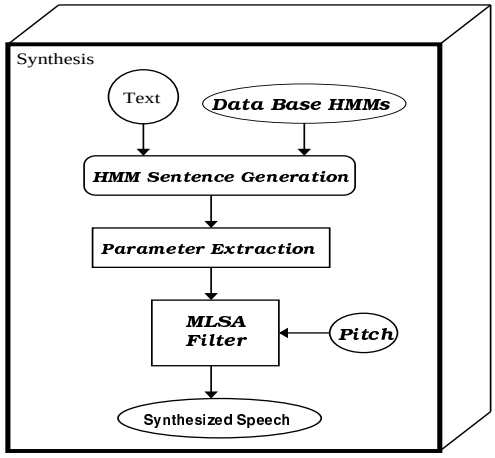

Fig. 3. The synthesis part.

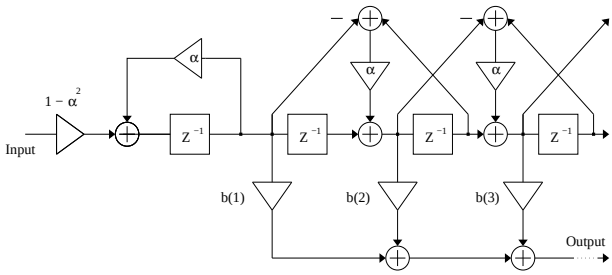

Fig. 4. Basic Filter F(z).

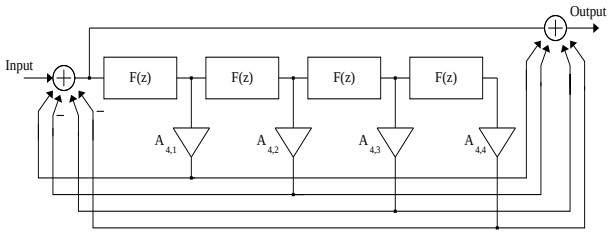

Fig. 5. R_L**(-F(z))** $\simeq \frac{1}{D(z)}$ **L =4.**

and $p_{q_k}(d_{q_k})$ is the probability of consecutives observations generated in state q_k. M represents the order of mel-cepstral coefficients.

3.6. MLSA Filter

The MLSA (Mel Log Spectrum Aproximation) filter [8] [11] inputs mel-cepstral coefficients and pitch information and outputs speech. It performs a highly accurate aproximation. The transfer function $\frac{1}{D(z)}$is exponential, howeve, it is possible to aproximate it with sufficient accuracy by a rational transfer function. The complex exponential is aproximated by a rational function as follows :

$$exp(w) \simeq R_L(w) = \frac{1 + \sum_{l=1}^{L} A_{L,l} w^l}{1 + \sum_{l=1}^{L} A_{L,l} - w^l}. \qquad (10)$$

Thus $\frac{1}{D(z)}$ can be aproximated as follows :

$$R_L(-F(z)) \simeq exp(-F(z)) = \frac{1}{D(z)} \qquad (11)$$

under the constraint

$$F(z) = \sum_{m=1}^{M} b(m)\Phi_m(z). \qquad (12)$$

where

$$\Phi_m(z) = \begin{cases} 1, & m = 0 \\ \frac{(1-\alpha^2)z^{-1}}{1-\alpha z^{-1}} \tilde{z}^{-(m-1)}, & m \geq 1 \end{cases}$$

where $b(m)$ can be obtained through a linear transformation of $c_\alpha(m)$. The coefficients $A_{L,l}$ have the same value as the LMA filter.

The basic block, $F(z)$, is shown in Figure 4. Figure 5 shows the MLSA filter $R_L(-F(z))$ for the case of $L = 4$.

3.7. Speech Synthesis

Hitherto, all the training part has already been done. For a given text to be synthesized, after the phonetic transcription, we will have a triphone sequence. An algorithm is implemented to search for each triphone on the data base. If this speech unit is not found, diphone HMMs, if they exist, are concatenated instead, otherwise, monophones are used. Figure 3 explicits the procedures described.

By concatenating those HMM units, a sentence HMM is constructed. An algorithm was implemented to obtain the static parameters from this sentence. Using these mel-cepstral static coefficients and constant pitch information as MLSA [10] filter's input, speech signal is synthesized [10]. The MLSA filter is provided by SPTK.

4. RESULTS

A Comparative listening test was conducted for evaluation of our HMM-based speech synthesizer against the TalkActie sinthesizer. The TalkActive performs concatenative speech synthesis using the PSOLA's algorithm.This intelligibility test was performed with 10 people (9 men and one woman).

A sound speaker was used to listen to the TalkActive sinthesizer , on the other hand, head-phones were used to listen to the HMM-based sinthesizer.

Synthesizing speech with the HMM technology we obtain better results, as can be seen on table 4.

Table 4. Table of the comparative listening test.

Using PSOLA's algorithm	With HMM-based
88.1%	90%

5. CONCLUSIONS

In this paper we have presented a Brazilian Portuguese TTS based on HMMs. One advantage of HMM-based TTS systems is that speaker adaptation can be easily performed. The usage of SPTK and HTK toolkits are fundamental in our system. Preliminary subjective results show that the proposed system outperforms a PSOLA TTS based on syllabic units. Our further work will be improving the quality of synthesized speech through the bootstrap initialization and pitch modeling, which can take account of the prosody.

6. ACKNOWLEDGMENT

Special thanks to professor Keiichi Tokuda, for the useful discussions.

7. REFERENCES

[1] M. TAMURA, S. KONDO, T. MASUKO, AND T. KOBAYASHI. Text-to-audio-visual speech synthesis based on parameter generation from hmm. In "Proc. EUROSPEECH", pages 959–962 (1999).

[2] T. MASUKO, K. TOKUDA, T. KOBAYASHI, AND S. IMAI. Voice characteristics conversion for hmm-based speech synthesis system. In "Proc. ICASSP", pages 1611–1614 (1997).

[3] K. TOKUDA, T. MASUKO, T. YAMADA, T. KOBAYASHI, AND S. IMAI. Speech parameter generation from hmm using dynamic features. In "Proc. EUROSPEECH", pages 757–760 (1995).

[4] S. YOUNG, D. KERSHAW, J. ODELL, D. OLLASON, V. VALTCHEV, AND P. WOODLAND, editors. "The HTK book".

[5] "The SPTK Reference Guide".

[6] T. YOSHIMURA, K. TOKUDA, T. MASUKO, T. KOBAYASHI, AND TADASHI KITAMURA. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In "Proc. EUROSPEECH", pages 2347–2350 (1999).

[7] J.A. SOLEWIC. Sińtese de voz a partir de texto para o portugues do brasil. Master's thesis, PUC-RJ (1993).

[8] T. FUKADA, K. TOKUDA, T. KOBAYASHI, AND S. IMAI. An adaptative algorithm for mel-cepstral analysis of speech. In "Proc. ICASSP", pages 137–140 (1992).

[9] K. TOKUDA, T. KOBAYASHI, AND S. IMAI. Speech parameter generation from hmm using dynamic features. In "Proc. ICASSP", pages 660–663 (1995).

[10] T. MASUKO, K. TOKUDA, T. KOBAYASHI, AND S. IMAI. Speech synthesis using hmms with dynamic features. In "Proc. ICASSP", pages 389–392 (1996).

[11] S. IMAI. Cepstral analysis synthesis on the mel frequency scale. In "Proc. ICASSP", pages 93–96 (1983).