Implementation of a Large Vocabulary Continuous Speech Recognition System for Brazilian Portuguese

Rafael Teruszkin and Fernando Gil Vianna Resende Junior

Abstract – **This work presents the implementation of a large vocabulary speech recognition system for Brazilian Portuguese. The implemented system uses tools available on HTK and ATK toolkits. Tests were conducted in order to check the correlation on the context of continuous speech recognition among the following variables: word recognition rate, perplexity, distinct language models, computational complexity and vocabulary size. A speech database was used to train the stochastic acoustic models based on continuous HMMs, and a textual database was developed to train language models based on *n*-grams. Vocabularies ranging between 3.528 and 60.000 words were tested. The best accuracy rate obtained with a dictionary size of 3.528 words was 90% when recognizing sentences with 9 to 12 words, and 81% with 60.0000 words, both of them being speaker dependent, with perplexities ranging between 250 and 350, and processing times less than one minute per sentence.**

Index Terms - **Continuous Speech Recognition, Brazilian Portuguese, Large Vocabulary, Continuous HMMs, *N*-grams.**

I. INTRODUCTION

UNIVERSITIES and industry have been attempting to solve practical problems within the area of speech recognition to make natural speech recognition feasible. Their target is to build systems which can be used without intensive user training and with minimum error rates. The last decade testified a significant progress in speech recognition technology.

Systems that use speech recognition for Brazilian Portuguese on their interface have already been studied in the past [1],[2],[3],[4]. The objective of this work is to study techniques used on continuous speech recognition systems that implement the state of art. The methodology that uses acoustic and language models for speech recognition was adapted for Brazilian Portuguese language. Tests varying the main parameters have been carried out in a way to help development of continuous speech recognition systems with large vocabulary and completely adapted to the Brazilian Portuguese language.

The next chapters of this work are organized in the following way: Chapter 2 presents the fundamentals of continuous speech recognition systems with large vocabulary. Chapter 3 describes the commonly used techniques to evaluate these systems. Chapter 4 shows how the training and test databases used in this work were developed. Chapter 5 brings the results of tests and in Chapter 6 conclusions about this work and suggestions for future research are presented.

II. CSR SYSTEMS FUNDAMENTALS

Current continuous speech recognition (CSR) systems with large vocabulary are strictly based on the principles of statistical pattern recognition [5],[6]. The basic methods where these principles are applied today still have strong influence from pioneering systems developed in the 70s [7],[8]. The architecture represented in Fig. 1 is practically a consensus in the area and is composed of the following components: front-end interface for capturing and extracting speech signal parameters, acoustic models, lexicon of words (optional), language model and, most important, the decoder. These blocks will be better explored in the remaining of this and next sections.

An unknown utterance, represented in Fig. 1, is received by the front-end interface and is converted into a sequence of acoustic vectors $X=\{x_1, x_2,..., x_T\}$, where T is the number of speech segments. This utterance is related to a sequence of words $W=\{w_1, w_2,..., w_n\}$, with n unknown, and is the responsibility of the CSR system to determine the most likely sequence, $\hat{W}$, for the observed acoustic vectors, X, in the terms defined by Equation (1):

$$\hat{W} = \arg\max_{w}\left[P(W \mid X)\right] = \arg\max_{w}\left[\frac{P(W)P(X \mid W)}{P(X)}\right] \qquad (1)$$

In Equation (1), after applying the Bayes Rule, the a posteriori distribution $P(W \mid X)$ is decomposed in $P(W)$, the a priori probability of word sequence W, and $P(X \mid W)$, which is the probability of observing the acoustic evidence X when sequence W is uttered. $P(X)$ is irrelevant on Equation (1) because it does not depend on W. The distribution $P(W)$ refers to the words that could have been uttered and is associated with a language model (LM). The probability model of an observation $P(X \mid W)$ is known as acoustic model (AM) [9].

To convert the architecture drawn in Fig. 1 into a practical system, the solution of some problems is required. First, the speech parameterization done in the front-end must extract from the speech signal all the needed acoustic information in a compact form and compatible with the acoustic models based on HMMs (Hidden Markov Models). In second place, the HMMs must represent the distributions of every sound in each one of the many contexts where they can occur. The needed parameters are estimated from speech databases that never cover all

Rafael Teruszkin and Fernando Gil Vianna Resende Junior are with Program of Electrical Engineering, Federal University of Rio de Janeiro. Fernando Gil Vianna Resende Junior is also with Department of Electronic Engineering and Computer Science, Polythechnic School, Federal University of Rio de Janeiro. E-mails: {rafaelt,gil}@lps.ufrj.br.

ISSN 1980–6604/$20.00 © 2006 SBrT/IEEE

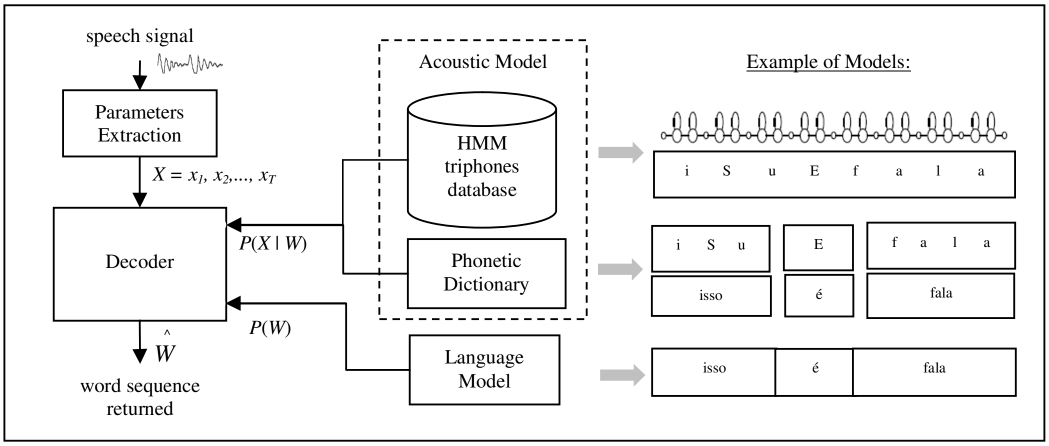


Fig. 1: Block diagram of a CSR system based on statistical modeling of sub-word units. During the speech recognition process, the word sequence *W*="*isso é fala*" which means "*this is speech*", is postulated by the decoder. The language model computes its probability *P*(*W*) and the acoustic model calculates the probability *P*(*X* | *W*). This process can be repeated for all possible word sequences and the most likely sequence is then selected as a result.

possible contexts. In third place, the language model must be designed to give precision to the prediction of words, taking into account its recent history. However, as well as for the HMMs, the variability of speech is always a present problem and the language model must be apt to deal with word sequences for which it has no occurrence found in the training database. Finally, the above delineated process for finding W, enumerating all the possible word sequences, is computationally impracticable for a large lexicon of words. However, word sequences with potential can be exploited in parallel, discarding as soon as possible hypotheses that become improbable. This process is called decoding and the design of efficient decoders is crucial for the accomplishment of practical CSR systems, capable to carry out fast and with good precision the operations on the existing computing platforms. The sections below will approach each one of these problems with greater detail.

A. SPEECH PARAMETERS EXTRACTION

An important assumption on the design of current speech recognition systems is that speech signal can be considered stationary during an interval of some milliseconds. Thus, the main function of the front-end stage is to divide the speech signal into segments and from each segment derive a smooth estimative from the spectre. The spacing between segments is typically 10ms and the analysis window length is about 25ms.

In the majority of current CSR systems, the energy of the signal together with the 12 first melcepstral coefficients are computed in each segment to form a basic acoustic vector with 13 elements. As will be seen, the acoustic modeling assumes that each acoustic vector is uncorrelated with its neighbors. This assumption is a little poor since the physical requirements of human vocal apparatus guarantees the continuity between successive spectral estimatives. However, adding the differentials of first and second order to the basic static coefficients reduces the problem enormously [10],[11].

To compensate the long duration effects in the spectre mainly caused by the use of different microphones and audio channels (including in the distance between the speaker and the microphone), as well as stationary noise, it is usual to carry out a spectral subtraction from all acoustic vectors. This operation is known as cepstral mean normalization (CMN). In practice, the calculated spectral average is updated at each segment to protect the recognition system against channel variations (the updated average is known as running average) [12].

B. ACOUSTIC MODELING

The purpose of an acoustic model is providing a way to calculate the probability of any vector sequence *X*, given a word sequence *W*. In principle, the required probability distribution *P*(*X* | *W*), can be modeled through innumerable word sequences and the statistical calculation of corresponding vectors sequences. However, this method is impracticable for systems with large vocabulary and, in spite of this, HMM phoneme models are created and a word model is composed by concatenating the corresponding phoneme models.

Each phoneme, represented by a first-order HMM, typically contains three emitting states on a simple left-right topology, as illustrated in Fig. 2. Non-emitting entry and exit states are added to the model for facilitating the union of different models. The exit state of one phoneme model can be joined with the entry state of another to create a composite HMM. This allows joining phoneme models to form words and joining these word models to form complete phrases. However, the pause model, by being stationary, is normally represented by a simpler topology of only one emitting state.

An HMM can be easily understood as a generator of vector sequences. It is a finite state machine that modifies its state at each time unit. At each time *t* and state *j*, an acoustic vector x_t is generated with probability density $b_j(x_t)$. Moreover, the transition from state *i* to state *j* is also probabilistic and is governed by a discrete probability a_{ij}.

Less complex systems based on HMM use discrete output probability functions with vector quantization (VQ). On these systems, each received acoustic vector is replaced by the index of the closest vector found on a pre-computed codebook. The output probability function consists of a simple search made on a look-up table that contains all output probabilities of each possible index among all quantized vectors.

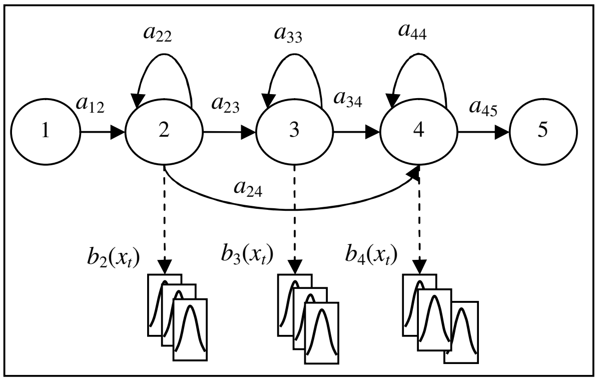


Fig. 2: Phoneme model based on HMM.

This method is efficient computationally, however, the quantization introduces a noise that limits the precision of the system. Therefore, more efficient systems use output distributions parameterized with continuous densities that model the acoustic vectors directly [13],[14]. The type of distribution normally chosen is the mixture of M multivariate Gaussian:

$$b_j(x_t) = \sum_{m=1}^{M} c_{jm} \mathcal{N}\left(x_t, \mu_{jm}, \Sigma_{jm}\right) \tag{2}$$

where c_{jm} is the weight of mixture component m, being in state j, and $\mathcal{N}\left(x,\mu,\Sigma\right)$ denotes a multivariate Gaussian with average μ, covariance Σ and l size acoustic vectors:

$$\mathcal{N}\left(x,\mu,\Sigma\right) = \frac{1}{\sqrt{(2\pi)^l |\Sigma|}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)} \tag{3}$$

So far, it was implicitly assumed that only one HMM is needed per phoneme and, as Portuguese requires approximately 40 phonemes, it could be concluded that only 40 HMMs must be trained. In practice, contextual effects cause a wide variation in the way that different sounds are produced (these phonemes that exists in different contexts are called allophones). To obtain a good phonetic discrimination, distinct HMMs must be trained for each of the different contexts of a phoneme. The usual way of solving this problem is using triphones, where each of the phonemes generates distinct HMMs for the pairs formed by phonemes situated on its right and left side. For example, assume that the notation *x-y+z* represents the phoneme *y* occurring after phoneme *x* and before phoneme *z*. Then, the sentence "*muito prazer*", which means "*nice to meet you*", could be represented by the phonetic sequence "*sil m u j~ t u p r a z e X sil*" and, for the case of triphones, the sequence could be transcribed as:

sil sil-m+u m-u+j~ u-j~+t j~-t+u t-u+p u-p+r p-r+a r-a+z a-z+e z-e+X e-X+sil sil

Notice that the contexts of triphones enclose the borders between words and the two instances of the phoneme "u" above must be represented by different HMMs, as its contexts are different. The use of this cross-word triphones results in a more precise modeling, however takes to complications in the decoder as will latter be seen. Simpler systems result of using only word-internal triphones, where the example above would become:

sil m+u m-u+j~ u-j~+t j~-t+u t-u p+r p-r+a r-a+z a-z+e z-e+X e-X sil

Here, less distinct models are necessary, simplifying in such a way the problem of parameter estimation as well as the decoder design. However, the cost is losing the ability of modeling contextual effects in the borders of words and, consequently, in modeling natural speech.

The use of Gaussian mixtures on the output distributions allows each state distribution to be modeled with good precision. In practice, it is referred that with about 10 mixture components a good performance can be achieved in large vocabulary recognition (LVR) systems [15].

C. LANGUAGE MODELING

The purpose of a language model is to provide a mechanism that estimates the probability of a word w_k, in a sentence, as a function of the words that precedes it, respectively, $w_1, w_2, ..., w_{k-1}$. This probability, represented by $P(W)$ and defined in the Equation (1), is essential to get good results in CSR systems:

$$P(W) = P(W_1^N) = P(w_1,...,w_N) \tag{4}$$

where $P(W)$ is rewritten as $P(W_1^N)$, and the indexes 1 and N represent the first and the last word of the sequence. The joint probability for the words on Equation (4) can be replaced by the product of their conditional probabilities in the following way:

$$\begin{aligned} P(W_1^N) &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1,w_2)...P(w_N \mid w_1,...,w_{N-1}) \\ &= P(w_1)\prod_{i=2}^{N} P(w_i \mid w_1,...,w_{i-1}). \end{aligned} \tag{5}$$

A simple, but effective, manner of estimating these probabilities is using the *n*-grams technique, on which it is assumed that the word w_k depends only on the *n*-1 precedent words on the sequence:

$$P(W_1^N) \approx P(w_1)\prod_{i=2}^{N} P(w_i \mid W_{i-n+1}^{i-1}) \,. \tag{6}$$

In the attempt to capture the existing correlation between neighboring words, the *n*-grams absorb simultaneously syntax, semantics and pragmatic existent in the observed sentences. This makes them extremely effective in languages as English or Portuguese where the order of the words is important since the stronger contextual effect normally comes from the closer neighbors. Moreover, the probability distributions from the *n*-grams can be directly computed from texts and, therefore, it does not have a requirement for defining explicit linguistic rules like a formal grammar of the language does.

D. DECODER

In the previous sections the main components of a LVR system had been described: speech parameters extraction at the front-end, acoustic and language modeling. So, to make speech recognition work, using these components, a search in all possible word sequences *W*, maximizing *P*(*W* | *X*), as described in (1) should take place. This is a search problem and its solution is assigned to the decoder.

Decoding is a search process where a vector sequence which corresponds to the acoustics characteristics from the speech signal is compared with word models. In a general way, the speech signal and its transformations do not supply a clear indication about the borders between words nor about the total number of words in one given utterance so the determination of these is part of the decoding process. In this process, all the word models (formed from its respective phonemes models) are compared with a sequence of acoustic vectors. The number of models grows with the vocabulary, and can generate very big search spaces, what makes the search process onerous, computationally speaking, and therefore slow. In general, this stage of recognition in modern systems is responsible for most of the computational effort in continuous speech recognition and, therefore, it is the one that determines the final speed of these systems.

During the maximization process of Equation (1), repeated hereafter for convenience, the term *P*(*X* | *W*) is expanded in accordance to its acoustic model, tying the HMM states to their output emission probabilities. *P*(*X* | *W*) is then calculated as a sum of all transition possibilities between the state sequences of the model under hypothesis:

$$\begin{aligned}\hat{W}_1^N &= \arg\max_{W_1^N}\left\{P\left(W_1^N\right)P\left(X_1^T \mid W_1^N\right)\right\}\\ &= \arg\max_{W_1^N}\left\{P\left(W_1^N\right)\sum_{S_1^T}P\left(X_1^T, S_1^T \mid W_1^N\right)\right\} \qquad (7)\\ &\approx \arg\max_{W_1^N}\left\{P\left(W_1^N\right)\max_{S_1^T}\left(P\left(X_1^T, S_1^T \mid W_1^N\right)\right)\right\}\end{aligned}$$

where $W_1^N = w_1,...,w_N$ represents the word sequence hypothesis (composed by their respective sub-word HMMs), $S_1^T = \{s_1,...,s_T\}$ is the state sequence hypothesis within the model, and $X_1^T = \{x_1,...,x_T\}$ is the observed acoustic vectors. The sum on the second Equation is replaced by a maximization, in a process referred as Viterbi Maximum Approximation [8]. Instead of summing on all the ways, we consider only the most likely way.

In this maximization process, the search space can be described as a network where the best time alignment between the input sequence and possible state sequences is searched. The search can be carried out in two levels: in the level of states (S_1^T) and in the level of words (W_1^N). It is possible to efficiently recombine the hypotheses on these two levels using dynamic programming (DP) [16], limiting the combinatorial explosion on the number of search hypotheses.

Search strategies based on DP are successfully used in a great number of speech recognition tasks today, such as digit sequences recognition and LVR systems with almost no restriction for input speech. Many variants of search based on DP were already known in the 70s [17],[18],[19],[20],[21]. In the last three decades, these and other strategies related to DP had become surprisingly effective on dealing with vocabularies of 20k words or more.

The implemented CSR system uses the algorithm named Viterbi Beam Search as its decoder. The Viterbi decoding is a time-synchronous DP algorithm that searches the most likely HMM state sequence for some input speech. As the state space described before can be huge, even for medium size vocabulary applications, the heuristics of a beam search is normally applied for limiting the search through the pruning of unlikely search hypotheses. The combination of the search algorithm and some efficient pruning method is always referred as Viterbi Beam Search [22].

III. CSR SYSTEMS EVALUATION

A. LANGUAGE MODEL PERPLEXITY

The language model tends to minor the uncertainties (to diminish the entropy) about the content of the sentences and to facilitate its recognition. For example, if on the average there are a few words that can follow one given word in a LM, the recognition system will have fewer options to verify and the performance will be better than if many possible words existed. This example suggests that a convenient form to measure the difficulty imposed by the LM in the search process must involve the average number of words that can follow others. If the LM is seen as a graph, with terminals associated to the transitions between words, for example, then this measure would be related with the average ramification factor in all decision points of the graph. In a simplified view, this is the amount measured by the perplexity, defined as follows.

Consider a word sequence $W = \{w_1, w_2, w_3...w_N\} = W_1^N$ as a random proccess. The estimated entropy of this process can be defined as [23]:

$$\hat{H}(W) \approx -\frac{1}{N}\log_2 P(W_1^N) \qquad (8)$$

where $P(W_1^N)$ is the probability for word sequence W_1^N, estimated by the language model. Here W_1^N is considered an ergodic process and the approximation above requires this sequence to be long enough.

The perplexity *PP*(*W*) of a LM *P*(*W*) is defined as the reciprocal of the geometric average probability that the model associates to each word on the test set W_1^N. This measurement, related to the entropy as stated above, is known as test set perplexity and from Equation (8):

$$PP(W) = \frac{1}{\sqrt[N]{P(W_1^N)}} \approx 2^{\hat{H}(W)} \qquad (9)$$

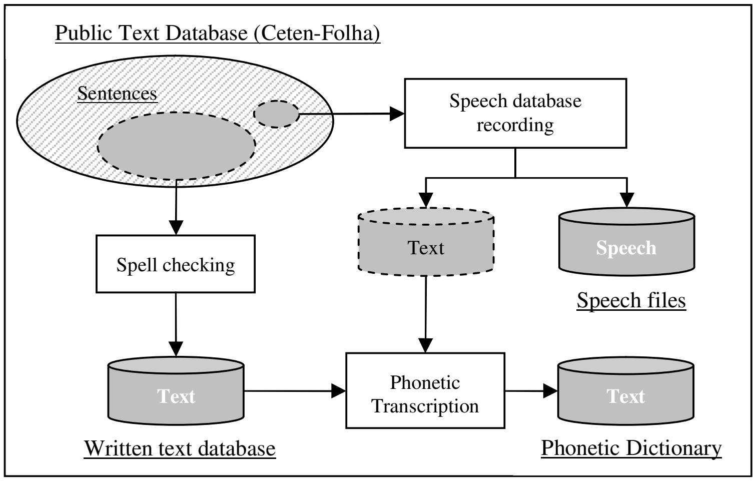


Fig. 3: Relationship between speech and written text databases used in this work.

TABLE I
LM perplexities and associated *WER* for different tasks [23].

Corpus	Vocabulary Size	Perplexity	*WER*
TI Digits	11	11	~0.0%
OGI Alphadigits	36	36	8%
Resource Management (RM)	1.000	60	4%
Air Travel Information Service (ATIS)	1.800	12	4%
Wall Street Journal	20000	200-250	15%
Broadcast News	> 80.000	200-250	20%
Conversational Speech	> 50.000	100-150	30%

The perplexity can be interpreted as the geometric average of the text ramification factor when presented to the LM. The perplexity defined in Equation (9) has two key parameters: the LM and the word sequence *W*. The test set perplexity evaluates the LM capacity of generalization. It can be said that a low perplexity is related to a possible better performance of the recognition system. This happens due to the perplexity being essentially a statistical weighed measurement for the ramification factor of the training set. As perplexity becomes higher, statistically speaking, the CSR system will need to consider more branches.

B. WORD ERROR RATE

Currently, the precision of CSR systems is typically described by the word error rate (*WER*), as defined below [5]:

$$WER = \frac{S + I + D}{N} \tag{10}$$

where *N* is the number of words on the input sequence, *S*, *I* and *D* are, respectively, the number of substitution, insertion and deletion errors on the result word sequence when compared with the input. Correspondently, the word recognition rate (*WRR*) can be calculated as:

$$WRR = 1 - WER \tag{11}$$

TABLE I shows the relationship between vocabulary size, LM perplexity and *WER* for different CSR tasks.

IV. DATABASE DEVELOPMENT FOR BRAZILIAN PORTUGUESE

The availability of databases is a conditioning factor for the development of CSR systems. Shared databases between American and European researchers had been one of the main reasons for the progresses achieved on the last decades in these regions. For the Brazilian Portuguese, however, there is still a lack of common databases and, alternatively, only individual initiatives on research centers can eventually be found [3],[24]. However, none of these initiatives completely takes care of all needed requirements for training large vocabulary continuous speech recognition (LVCSR) systems for Brazilian Portuguese language. Databases for training and testing these systems are normally subdivided in 3 parts:

- Speech files with their associated texts
- Big mass of written texts
- Phonetic dictionary

Regarding the lack of complete databases for Brazilian Portuguese and due to the fact that a speech database had been already developed, it was opted to use the advantage of this material to build a new database to support the development of LVCSR systems [25]. In the following sections it is described how each one of the three groups mentioned before were designed (Fig. 3).

A. SPEECH FILES

The speech files are used for training the acoustic models and testing the CSR system. Speech databases are normally grouped by those generated from read speech or those captured by recording spontaneous speech. For this work, 1.000 sentences recorded in studio through read speech by a single speaker are used. The texts of these sentences had been extracted from a public database [26] with the goal to be phonetically balanced on Brazilian Portuguese language [27]. The segmentation of this database into sentences was manually done, given that originally an audio file was created for each group of 20 sentences. All files have been recorded in the "wav" format, sampled with 16 bits, 48 kHz and only one channel. When necessary, the associated text files have been modified to reflect what was spoken in fact. No other manual segmentation was executed, so during the training of the acoustic model the database was segmented through the technique known as flat start [28].

B. WRITTEN TEXTS

Written texts are used to train the language model. As the used LM is statistical (*n*-gram), a great amount of texts was collected to allow the estimation of existing relationship between words during speech. On the database used for attainment of texts [26], a filtering work was done so that orthographic and grammatical errors, incorrect markings, and foreign words, among others errors found on the database did not harm the models to be created. For such, the spell checker named "br.ispell" [29] was used to validate the orthography of the words from each sentence under analysis. Whenever some word in a sentence was not found in the dictionary of "br.ispell" (which has 273.760 words), then the sentence as a whole was discarded. Before this analysis, however, a function was applied to transcribe numbers and ordinals to text, in a way that the sentence below:

<*s*> *gramado realiza de* ***20*** *a* ***23*** *de outubro o* ***6º*** *festival do turismo* </*s*>

was converted to:

<*s*> *gramado realiza de* ***vinte*** *a* ***vinte e três*** *de outubro o* ***sexto*** *festival do turismo* </*s*>

where the symbols "<*s*>" and "</*s*>" indicate, respectively, start and end of the sentences. The texts associated to the speech files had also been monitored so that they were not included in the resulting set of texts (and later not masking the perplexity measurement of the LM). With these restrictions, a total of 362.117 sentences were collected, from the original set composed by approximately 1.5 million sentences.

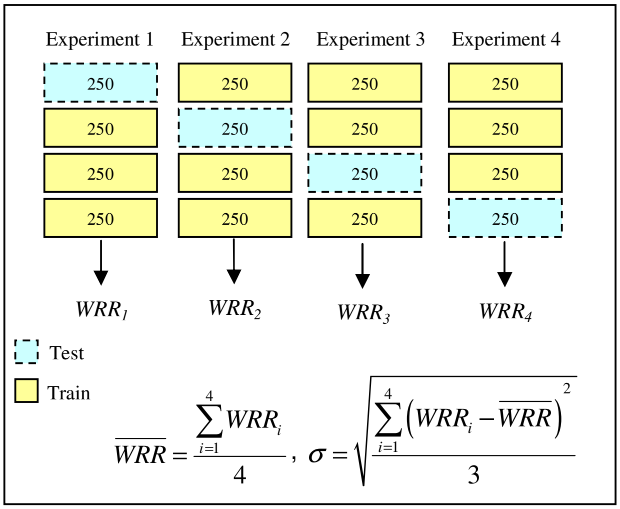
C. PHONETIC DICTIONARY

The phonetic dictionary used in this work was automatically generated using a phonetic transcription algorithm developed for Brazilian Portuguese language [30],[31]. This automation speeded up the creation of different dictionaries and their use in the tests of diverse configurations of the implemented CSR system.

TABLE II, brings the list of phonemes used by the transcriber with corresponding examples of words containing them.

V. TESTS AND RESULTS

The target for the tests presented in this section was to understand, in practice, how the main components of a LVCSR system behave: the phonetic dictionary, acoustic model, language model and the decoder. The existing correlation between these components and the performance presented by the system are analyzed in respect of the processing time, as well as the word recognition rate.



$$\overline{WRR}=\frac{\sum_{i=1}^{4}WRR_i}{4},\quad \sigma=\sqrt{\frac{\sum_{i=1}^{4}\left(WRR_i-\overline{WRR}\right)^2}{3}}$$

Fig. 4: Test and training database selections for obtaining the average *WRR* and standard deviation for each experiment.

All the tests were executed on a computer with Dual Intel® processor (Xeon^{TM} 3.0 MHz) and 2 GB of RAM. The operational system used was a Linux Red Hat, version 3.2. The software used for training the acoustic and language models was built based on the libraries distributed with HTK package, version 3.3 [28]. For the accomplishment of the tests, it was used the AVite software, available in ATK package, version 1.4.1 [12].

The configuration framework used for training and testing the system was the standard one [28], [12] within the speech recognition area. Below is a list with details about the configuration used:

- Window length: 25ms
- Time to capture speech segments: at each 10ms (also knows as shift)
- Windowing function: Hamming window
- Pre-emphasis: 0.97
- Computed coefficients for each segment: Mel Cepstral
- Number of channels of the filter bank: 26
- Coefficient for Cepstral liftering: 22
- Total of coefficients: energy + 12 Mel Cepstral coefficients + Δ (1st derivative of these 13 coefficients) + $\Delta\Delta$ (2nd derivative). On total, the computed vector for each segment has 39 coefficients.
- Energy Normalization: none (ATK doesn't support)
- Cepstral Mean Normalization: running average
- Acoustic Modeling: Continuous HMMs (with diagonal covariance matrices)
- Acoustic Units (HMMs): triphones.
- HMM Modeling: 3 emitting states and 2 non-emitting states for model concatenation
- HMM State Tying: decision trees
- Decoder: Viterbi *Beam Search*

TABLE II
List of phonemes used in this work (40) and word examples using them.

Symbol	Examples
	Oral vowels (7)
A	lápis, jatobá, ábaco, capacete, cabeça, caça, lua, pedia
E	é, médico, pajé, épico, Pelé, pele, ferro, velho
e	capacete, resolver, respeito
i	justiça, pais, saia, lápis, idiota, aqueles, ele, pele
O	ópio, cópia, jogos, docas, sozinho, forte
o	resolver, jogo, golfinho, bolo, cor
U	baiacu, Raul, culpa, baú, cururu, logo, consolo, tijolo
	Nasal vowels (5)
a~	avião, campeão, andar, tampar, canção, cama
e~	então, consciência, tempo, bem, menos, dente
i~	ninho, tinta, latina, importa
o~	onda, campeões, somos, homem, fronha
u~	um, muito, umbigo
	Semi-vowels (4)
w	natal, fácil, voltar, eu, chapéu, quase, jaula
j	fui, pai, sei, foi, caracóis, hotéis, micróbio, pátria
w~	não, cão
j~	muito, bem, parabéns, compõe
	Unvoiced fricatives (3)
f	festa, fanfarrão, afta, afluente
s	sapo, caçar, crescer, sessão, lápis, tórax, capaz, disco, casca, desço, excesso
S	chá, xaveco, cachorro
	Voiced fricatives (3)
z	casa, coisa, quase, exato
v	vovó, vamos, avião
Z	geladeira, trovejar
	Affricates (2)
tS	tia, pacote, constituinte, Tijuca
dZ	dia, cidade, disco
	Plosives (6)
b	barba, absinto
d	dados, cidade, dominar, administrar
t	todos, pato, constituinte
k	casa, casca, quero, quanto
g	guerra, gato, agüentar, agnóstico
p	papai, psicológico, apto, rapto
	Liquids (5)
l	laranja, palafita, leitão
L	calhar, colheita, melhor
R	carro, rua, rato, carga, germe
X	casar, certo, harpa, arco
r	carona, garoto, frango, graxa, por exemplo
	Nasal consonants (3)
m	mamãe, ema, emancipar, marmota
n	nome, atenuar, encanação
J	casinha, galinha
	Silences (2)
sil	silence in the beginning / end of a sentence
sp	silence in the middle of a sentence (pause)

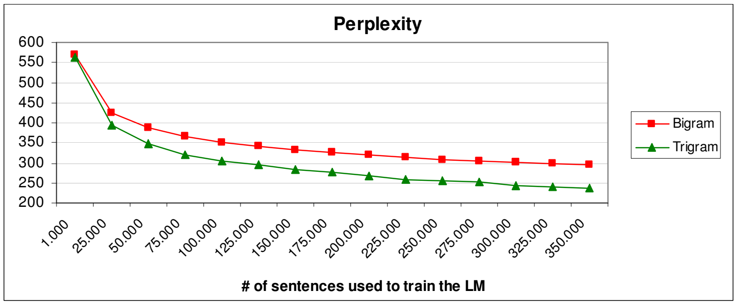


Fig. 5: Perplexity evolving against the number of sentences used to train the LM.

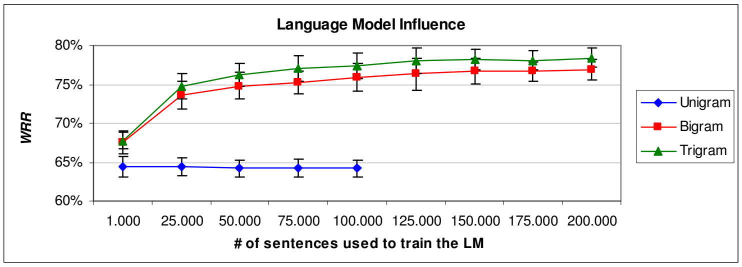


Fig. 6: $\overline{WRR}$ evolving against the number of sentences used to train the LM.

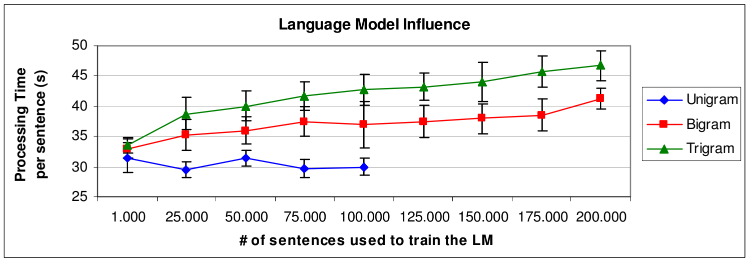


Fig. 7: $\overline{Tp}$ evolving against the number of sentences used to train the LM.

The speech database previously described and composed by 1.000 sentences, having each one between 9 to 12 words, was tested by breaking it up to 750 sentences for training and 250 for testing. This same separation was carried out four times, splitting the database into groups of 250 sentences and making all possible recombinations between these groups. The following results represent the average and standard deviation of the results obtained from each one of these groups, assuring the statistical quality of the experiments (Fig. 4).

The following sections present the experiments done, evolving gradually between one experiment and the next. In all the experiments, the vocabulary used contains at least the 3.528 words found on the 1.000 sentences from the speech database.

EXPERIMENT 1: EVALUATION OF THE LANGUAGE MODEL

This experiment evaluates the LM perplexity against the number of sentences used to train it. The vocabulary was kept constant during all the experiment (Fig. 5). The number of sentences used ranged between 1.000 and the 350.000 and the language models tested were the bigram and the trigram. The sentences used to measure the perplexity of each configuration were the 1.000 sentences of the speech database (remembering that there is no intercession between this database and the written texts database).

As expected, the perplexity tends to diminish as the number of sentences used in the training increases. This is related to the fact that the statistics of the trained models get improved as more occurrences of pairs and triples of words are registered in the database of written texts. The perplexities found in these experiments, seen in the steady state of the presented curves, are in accordance with values commonly found (TABLE I). It is important to observe in Fig. 5, that the difference of perplexities measured for the bigram and trigram models grows as more sentences are used in the trainings. With 350.000 sentences this difference exceeds 20%.

EXPERIMENT 2: COMPARISON BETWEEN DIFFERENT LANGUAGE MODELS

In this test, the language models trained before were used to test the CSR system, in order to evaluate the *WRR* with respect to the LM. The experiment started using 1.000 sentences from the written text database and finished with 200.000 sentences, because the $\overline{WRR}$ stopped to evolve, while the processing time increased (Fig. 6 and Fig. 7). In these experiments only word-internal triphones were used and a single Gaussian modeled the output distributions of the HMMs. Tests with unigram language models (where the probability of occurrence of a word have no relation with previous words) were included in the experiments only to state a comparison with the other models and were only tested on experiments up to 100.000 sentences, given that the results remained constant in this interval. In Fig. 7, the average processing time per sentence, $\overline{Tp}$, is shown. It is calculated in the same way as the $\overline{WRR}$. On the executed experiments, the $\overline{WRR}$ stopped to evolve after 150.000

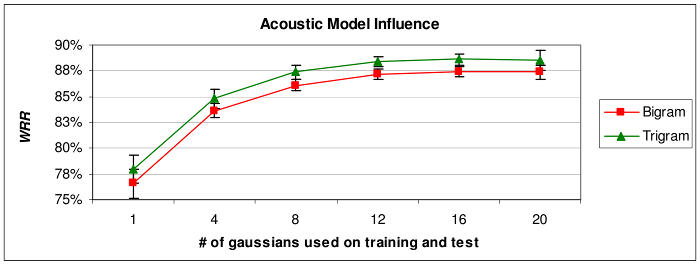


Fig. 8: $\overline{WRR}$ evolving against the number of Gaussians used for training the acoustic model and testing (using word-internal triphones).

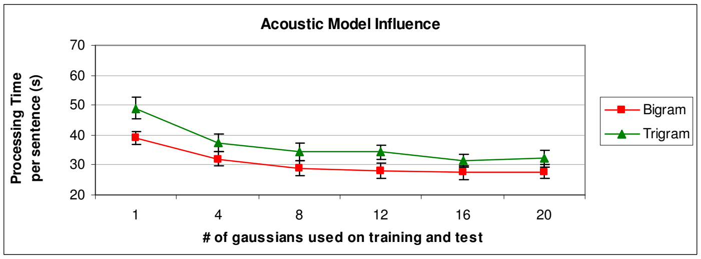


Fig. 9: $\overline{Tp}$ evolving against the number of Gaussians used for training the acoustic model and testing (using word-internal triphones).

sentences and, on the other hand, the processing time increased significantly. The reason for that could be related to a saturation of the language model, which after a certain limit, only increases itself with probabilities of sequences very infrequent in the test database, and, therefore, insignificant in the total result. For this reason, in the following experiments where the LM is fixed, only the training with 150.000 sentences will be considered.

EXPERIMENT 3: USING DIFFERENT NUMBER OF GAUSSIANS WITHIN THE ACOUSTIC MODEL

In these tests, it is considered the same configuration as before, where the LM was trained with 150.000 sentences. However, the number of Gaussians used in the state output distributions is varied, from a single Gaussian, as in the previous example, up to 20 Gaussians. The language models considered on the experiment are the bigram and trigram. Initially the test was done only using word-internal triphones (3-A) and in a second experiment (3-B), it is also considered cross-word triphones for training and testing the system.

EXPERIMENT 3-A: TESTS USING ONLY WORD-INTERNAL TRIPHONES

In Fig. 8 and Fig. 9, it can be seen that trigrams present higher performance in comparison to bigrams, always above 1%. This relation diminishes when the number of Gaussians used in the model increases, moving from 1,9% to 1,3%. A better trained acoustic model seems to alleviate the influence of the LM during the search.

EXPERIMENT 3-B: TESTS CONSIDERING ALSO CROSS-WORD TRIPHONES

In Fig. 10 and Fig. 11 and on previous experiments, the processing time decreases as more Gaussians are added to the model. With more Gaussians the acoustic model starts having its distributions better characterized. As a consequence, the decreasing observed on the processing time can be linked to a more effective pruning of the decoder at the acoustic level and consequent reduction of the search space. As the involved computational costs in the decoding process is much superior to any another computational cost of the CSR system, the complexity added for increasing the number of Gaussians is compensated by a possible improvement in the decoder performance.

When moving the mixture of Gaussians from 1 to 16, the $\overline{WRR}$ of the system was increased in almost 10%. The use of cross-word triphones improved the $\overline{WRR}$ in only 1%, however the processing time per sentence increased in approximately 10 seconds (+36%). The best obtained result with these configurations was a 90% *WRR* in the 3° test group of the speech database. As within these experiments the $\overline{WRR}$ remained constant above 16 Gaussians, the next experiments will consider this number as a default.

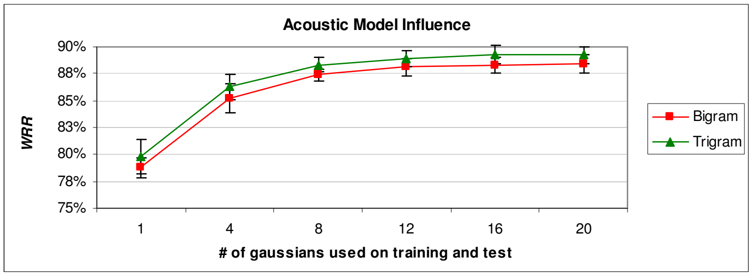


Fig. 10: $\overline{WRR}$ evolving against the number of Gaussians used for training the acoustic model and testing (using cross-word triphones).

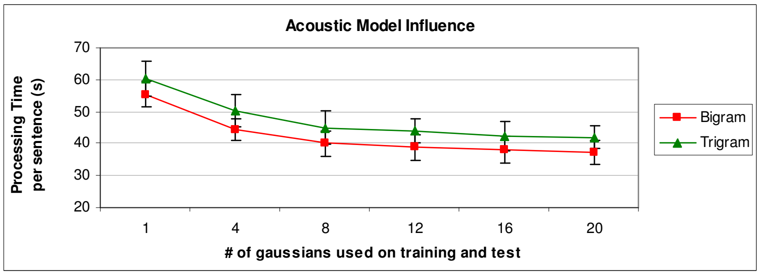


Fig. 11: $\overline{Tp}$ evolving against the number of Gaussians used for training the acoustic model and testing (using cross-word triphones).

EXPERIMENT 4: LM PERPLEXITY ANALYSIS AGAINST DICTIONARY GROWTH

The method used to increase the system dictionary was to gradually select sentences from the written text database and adding the new words that appears in these sentences to the dictionary (which had initially 3.528 words). Doing this way, the curve presented in Fig. 12 could be prepared, where the number of sentences varies from 0 to 350.000. In this last case, the size of dictionary reached 67.505 words. To an established goal of 60k words, it can be verified with the presented curve that this dictionary size is reached when 240.000 sentences are used. Fixing that number of sentences for training the language model, its perplexity against the test set (for the trigrams case) was measured for language models built with different sizes of dictionary, as seen in Fig. 13. As in Experiment 1, the sentences used to measure the perplexity of each configuration were the same 1.000 sentences of the speech database. Knowing that in the written text database, sentences have in average 10 words each, the amount of sentences selected compounds a database of approximately 2.4 billion words.

As seen in Fig. 13, the perplexity of the language model against the test set increases as the dictionary becomes bigger and the language model is retrained. A possible reason is that with bigger dictionaries the probability estimated to each n-gram trained is decreased as a consequence of using techniques to optimize sparse database training such as discounting and backing-off [32],[33]. As a result, the perplexity tends to increase. It can be also observed in Fig. 13 that the perplexity had not varied so much on the second half of the curve. That behavior can be related with the way the dictionary was expanded. In the beginning of the expansion, words with higher frequency on the database had more probability to be added to the dictionary and this reason also contributes to make these words relevant to the LM. Later, the dictionary was expanded mostly with less frequent words and whose smaller probabilities practically do not count on the LM decisions.

EXPERIMENT 5: TESTS WITH DIFFERENT DICTIONARY SIZES

The goal of this experiment was to test the behavior of a CSR system when its vocabulary is extended. In an initial configuration, it was used a trigram LM built with 240.000 sentences, 60k words, 16 Gaussians and the internal-word and cross-word triphones. The target was to measure the average processing time per sentence for this dictionary size, since with the original dictionary (containing 3.528 words) the system already presented times close to one minute per sentence. The time measured in this task rose to approximately 8 minutes per sentence, so the mass of tests that was planned to be made, as on previous experiments, became impracticable, besides having its practical applicability sufficiently reduced.

As an option, the parameters of CSR system which indicate the size of the beam used on Viterbi beam search had been empirically reduced in 20% intending to discard the less probable sequences and hence speed up the processing time, even knowing that the *WRR* probably would be decreased. The observed decrease on *WRR* was approximately 8%, with values around 81%, however, the

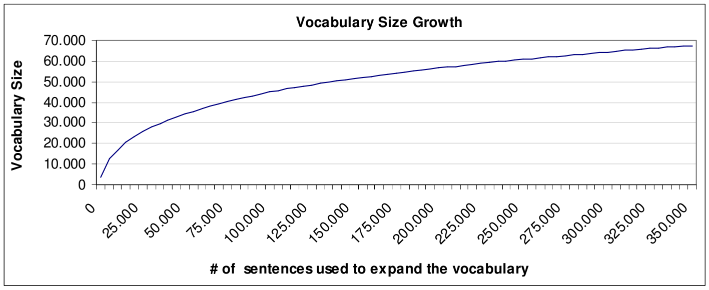


Fig. 12: Dictionary growth against the number of sentences used to expand it.

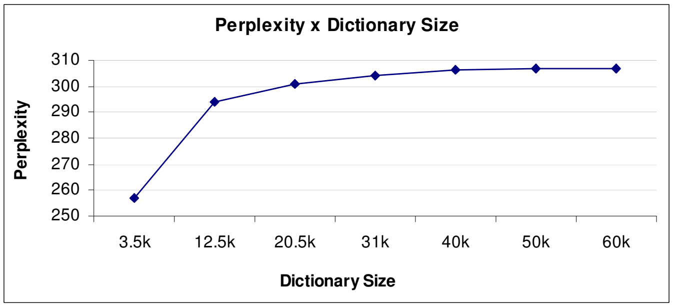


Fig. 13: Perplexity evolving with the different language models built during the dictionary growth.

TABLE III

$\overline{Tp}$ measurement for different dictionary sizes. The results are compared with those which used trigram LM and cross-word triphones.

LM / Dictionary	Bigram		$\overline{Tp}$ comparison with previous configuration
	$\overline{Tp}$	σ	
3.5k	8	1	-52,24%
31k	44	4	-59,25%
60k	64	5	-64,48%

measured processing time for the task of 60k words was close to 2 minutes per sentence and, for the task using the original dictionary (3.528 words), was approximately 15 seconds. This way, the effect of the beam to the performance of the decoder could be observed. It was also perceived that the initialization time of the decoder grows considerably when the vocabulary is extended, going from an almost imperceptible time (using the original vocabulary) to practically 8 minutes (with the 60k words vocabulary). The curves in Fig. 14 and Fig. 15 were obtained with the new configuration of beam applied to the decoder.

The $\overline{WRR}$ measurement practically remained constant during all the experiment, while the processing time continued growing, going from 17 seconds per sentence to about 3 minutes (Fig. 16 and Fig. 17). A possible reason for this behavior is the computational cost added to the search algorithm in allowing the decoder to use trigrams. To verify this fact, the previous test was repeated, however using bigram and only internal-word triphones, given that, as previously seen, considering cross-word triphones almost did not have effect on the $\overline{WRR}$ and represented a high computational cost to the system (in average, +30%).

As foreseen, it can be observed in TABLE III that the $\overline{Tp}$ effectively decreased (more then 50%), while the $\overline{WRR}$ remained approximately on the same level when compared to the tests that used a trigram LM and also considered cross-word triphones (only a light decrease was observed, inferior to 1%). In the task of 60k words, the decrease of $\overline{Tp}$ achieved almost 65%. In TABLE III, the term variation means the percentage change between a result and its previous correspondent.

The reduction observed on $\overline{Tp}$ perhaps explains the fact that many LVCSR systems use multi-pass algorithms, where, on the first pass, only bigrams and internal-word triphones are normally considered [22]. Examples of sentences recognized by the system on previous related experiments and their associated *WRR* can be observed in TABLE IV.

Previously published results with tasks of the same size applied to Brazilian Portuguese (60k words), having speaker independence, however with less phonetic variability on the database, presented *WRR* close to 63% [4]. A comparison between this number and the numbers presented before (with *WRR* around 81%) is difficult to be carried out, given the total incompatibility between these databases [24] and thus, remaining as a brief reference.

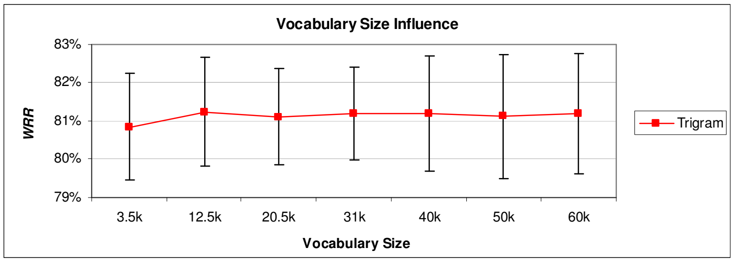


Fig. 14: $\overline{WRR}$ evolving against dictionary growth (with beam size parameters decreased in 20%).

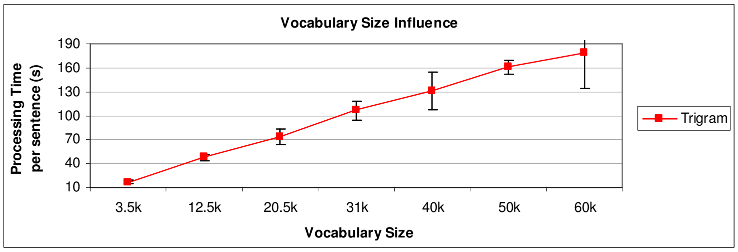


Fig. 15: $\overline{Tp}$ evolving against dictionary growth (with beam size parameters decreased in 20%).

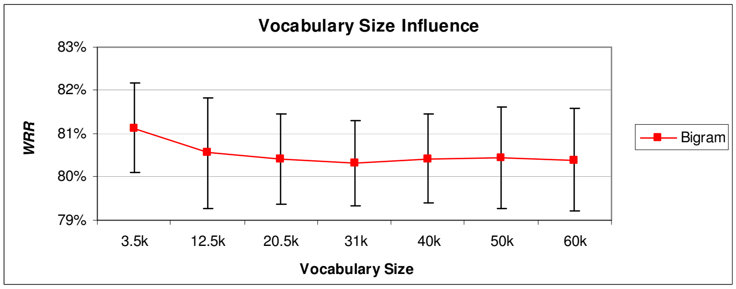


Fig. 16: $\overline{WRR}$ evolving against dictionary growth (considering only bigrams and internal-word triphones).

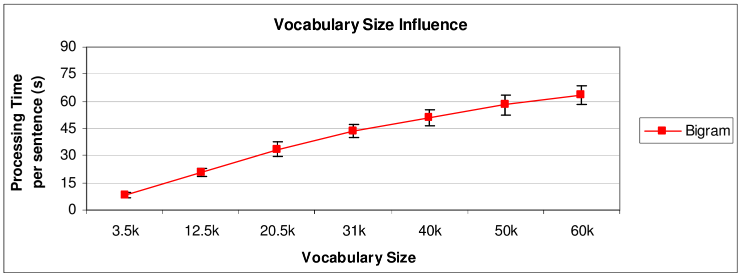


Fig. 17: $\overline{Tp}$ evolving against dictionary growth (considering only bigrams and internal-word triphones).

TABLE IV
Examples of speech recognition outputs using bigrams and dictionary size of 60k words. Inside the brackets of each case the number of deletion(D), substitution(S) and insertion(I) errors are shown. For each example, "LAB" means the original sentence and "REC" is the decoder output.

```
M012602:   60.00% [H=6, D=2, S=2, I=0, N=10]
 LAB: DÁ    VONTADE DE EU METER O MICROFONE NA SUA CABEÇA
 REC: DAVAM TARDE   DE    METER   MICROFONE NA SUA CABEÇA
M012605:   88.89% [H=8, D=0, S=1, I=0, N=9]
 LAB: UMA PÁGINA INTEIRA SERÁ DEDICADA A NOTAS E CURIOSIDADES
 REC: UMA PAZ    INTEIRA SERÁ DEDICADA A NOTAS E CURIOSIDADES
M012610:   75.00% [H=6, D=1, S=1, I=0, N=8]
 LAB: O PRESIDENTE MANDOU CHAMÁ-LO SEGUNDO A        IMPRENSA INTERNACIONAL
 REC: O PRESIDENTE MANDOU CHAMÁ-LO          SEGUNDA IMPRENSA INTERNACIONAL
M012619:   77.78% [H=7, D=1, S=1, I=0, N=9]
 LAB: SEM A CIRURGIA HUMORISTA PODERIA MORRER EM       SEIS MESES
 REC: SEM A CIRURGIA HUMORISTA PODERIA          MORREREM SEIS MESES
M012704:   90.91% [H=10, D=0, S=1, I=0, N=11]
 LAB: A ÚNICA CHANCE DO BAHIA ACONTECEU AOS QUARENTA E TRÊS MINUTOS
 REC: A ÚNICA CHANCE DO BAHIA ACONTECEU OS  QUARENTA E TRÊS MINUTOS
```

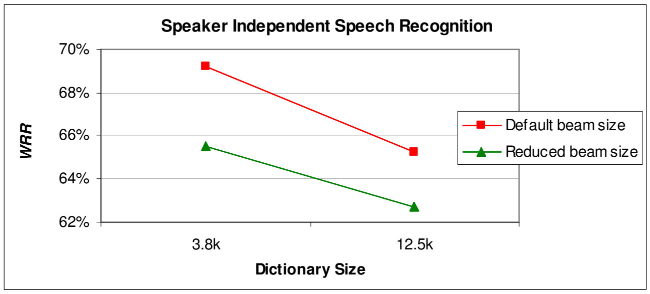


Fig. 18: *WRR* observed on preliminary tests of speaker independent speech recognition.

EXPERIMENT 6: SPEAKER INDEPENDENT SPEECH RECOGNITION

Preliminary tests with speaker independency were made possible by joining the database with 1.000 utterances, used in the tests presented before, to another database composed of 1.600 utterances [3], recorded by 40 distinct speakers (20 male and 20 female) and based on 200 phonetic balanced sentences [34]. The 1.600 utterances were split into training and testing sets, respecting simultaneously the criteria defined below (the 1.000's database was fully assigned to the training set):

- 1280 utterances for the training set, 320 for the testing set
- 32 speakers for the training set (16 M, 16 F), 8 for the testing set (4 M e 4 F)
- The 200 sentences were split in groups of 10: 16 groups for the training set, 4 for the testing set.

With the built databases, it was possible to observe the *WRR* of the speaker independent CSR system tested with dictionaries sizes of 3.8k and 12.5k words, bigram and only internal-word triphones. The dictionary of 3.8K words was built using the existing dictionary of 3.5k words (generated from the 1.000 sentences) and adding to itself the new words found on the 200 sentences. The *WRRs* obtained on the tests with these dictionaries and also with different configurations for the search beam size (the default one and the one reduced in 20%) are presented in Fig. 18.

VI. CONCLUSION

Because there is still much deficiency in databases available for speech recognition systems in Brazilian Portuguese, it was opted within this work to create a database that fulfilled the requirements for testing a LVCSR system. The speech database used in this work is composed by 1.000 utterances and was constructed to be phonetically balanced for Brazilian Portuguese language [27]. The written text database developed for training the LM has 350.000 sentences processed (approximately 3.5 billion words) and with their orthography verified. The phonetic transcription of the words was automatically done by a tool developed for this task [30],[31]. Although being small when compared to shared databases for the English language, the database composed by the 1.000 utterances presented good results on a speaker dependent LVCSR system (achieving 90% *WRR*), and its consistency was surveyed with small standard deviations (< 2%) measured on the experiments with different segments of the database.

The LVCSR implemented in this work is based on a one-pass Vitebi Beam Search decoder using a tree-structured lexicon, Gaussian multivariate continuous HMMs modeling triphones and *n*-gram language models (bigrams and trigrams). The tests made with the developed database were planned in a way to improve the understanding about the existent relation between some of the variables that are relevant to the design of acoustic and language models and the effect that these variables have on the performance of the decoder regarding both *WRR* and processing time.

In the initial tests, the language models based on unigrams, bigrams and trigrams had been compared, and a huge influence of these models over the performance of the decoder could be verified. The smaller measured perplexity of a language model (*PP*=236) was using trigrams trained with the 350.000 sentences and with the dictionary containing only the words found in the 1.000 sentences of the speech database (3.528 words). This number could possibly decrease more with the growth of the written text database, however on a very low rate, as could be verified with the trend of the presented curve (Fig. 5). For this size of vocabulary, an improvement in the system *WRR* for perplexities smaller then 283 could not be experimentally verified (this perplexity refers to the language models built with 150.000 sentences). In contrast with that, an increase in the processing time was observed.

Regarding the acoustic model, it could be observed an improvement in the *WRR* when considering cross-word triphones, beyond word-internal triphones, always with results superior in 1%. However, the processing time on this case had shown, in average, to be 30% greater. About the increase on the number of Gaussians used in the mixture of output distributions of continuous HMMs, besides improving the system *WRR*, also contributed to reduce the time expended by the decoder. For the obtained results, 16 Gaussians is shown to be an optimum number to compose the mixture.

During the expansion of the dictionaries, the effect of the size of the beam, used on search, over the performance of the system could be observed. It was also verified that the size of the dictionary has little influence in the *WRR*, however it is strongly attached to the computational complexity. The trigram language model, used in the one-pass decoder, revealed to be impracticable for vocabularies with more than 20k words: it achieved processing times greater then three minutes per sentence for a dictionary with 60k words (time expended with sentences ranging between 9 to 12 words). The bigram use, in these cases, reduced the computational complexity in practically 60%, almost keeping the same *WRR* (80%).

As subject to future works, the following topics are suggested:

- Study the relevance of words added to the dictionary which have very small frequencies in the database
- Improve speaker independent tests (with greater and more diverse databases)
- Tests with noisy databases
- Use segmented databases to initialize the acoustic models
- Use word class *n*-grams
- Add variants of pronunciation to words that present higher error rates
- Test other speech recognition toolkits (and other decoding techniques)
- Use multi-pass decoders
- Use speaker adaptation

ACKNOWLEDGEMENT

The authors would like to thank the reviewers for their comments and suggestions, which resulted in a significant improvement of this paper.

REFERENCES

[1] TERUSZKIN, R., RESENDE JR., F.G.V, VILLAS-BOAS, S.B., LIZARRALDE, F., "Object Oriented Library for Speech Recognition and Application on Robot Controlling ", *Automatic Control Brazilian Congress, Natal, Brazil*, September, 2002. (In Portuguese)

[2] SANTOS, S., ALCAIM, A., "A Task Dependent Continuous Speech Recognition System for the Portuguese Language", *Journal of the Telecommunications Brazilian Society*, 17(2):135–147, 2002. (In Portuguese)

[3] YNOGUTI, C. A., "Continuous Speech Recognition using Hidden Markov Models", *Electric Engineering College – UNICAMP*, D.Sc. Thesis, May, 1999. (In Portuguese)

[4] SILVA, Ê., BAPTISTA, L., FERNANDES, H., KLAUTAU, A., "Development of a Continuous Speech Automatic Recognition System with Large Vocabulary for Brazilian Portuguese", *Congress of the Brazilian Computation Society – Workshop TIL*, São Leopoldo, 2005. (In Portuguese)

[5] RABINER, L.R., JUANG, B., *Fundamentals on Speech Recognition*, New Jersey, Prentice Hall, 1996.

[6] HUANG, X., ACERO, A., HON, H.W., *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, New Jersey, Prentice Hall, chapter 11, 2001.

[7] BAKER JK, "The Dragon System – an overview", *IEEE Trans. ASSP*, Vol. 23, No 1, pp24-29, 1975.

[8] JELINEK, F., "Continuous Speech Recognition by Statistical Methods", *Proc. IEEE*, 64(4), pp532-556, 1976.

[9] COLE, R. A., MARIANI, J., USZKOREIT, H., ZAENEN, A. and ZUE, V., "Survey of the State of the Art in Human Language Technology", *Cambridge University Press, Cambridge*, UK, 1997 (http://cslu.cse.ogi.edu/HLTsurvey).

[10] FURUI, S., "Speaker-Independent Isolated Word Recognition using Dynamic Features of Speech Spectrum", *IEEE Trans. ASSP*, Vol. 34, No 1, pp52-59, 1986.

[11] APPLEBAUM T., HANSON B., "Regression Features for Recognition of Speech in Noise", *Proc. ICASSP*, S14.26, Toronto, 1991.

[12] ATK - API for HTK (http://htk.eng.cam.ac.uk/develop/atk.shtml)

[13] JUANG, B.H., "Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains", *AT&T Technical J*, Vol. 64, No 6, pp1235-1249, 1985.

[14] BAHL, L.R., BROWN, P.F., de SOUZA, P.V., MERCER, R.L., "Speech recognition with continuous parameters hidden Markov models", *Computer Speech and Language*, Vol. 2, No 3/4, pp219-234, 1987.

[15] WOODLAND, P.C, ODELL, J.J., VALTCHEV, V., YOUNG, S.J., "Large vocabulary continuous speech recognition using HTK", *Proc. ICASSP*, Vol. 19, pp. 125 - 128, April 1994.

[16] VINTSYUK, T.K., "Speech Discrimination by Dynamic Programming", *Kibernetika (Cybernetics)*, vol. 4, No. 1, pp. 81-88, Jan.-Feb. 1968.

[17] ITAKURA, F., "Maximum prediction residual principal applied to speech recognition", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-23, pp67-72, Feb. 1975.

[18] LOWERRE, B.T., REDDY, R., "The HARPY speech understanding system", *Trends in Speech Recognition*, pp340-360, W.A. Lea, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[19] SAKOE, H., CHIBA, S., "A dynamic programming approach to continuous speech recognition", *Proc. 7th Int. Congress on Acoustics*, Budapest, Hungary, Paper 20 C 13, pp65-68, August 1971.

[20] VINTSYUK, T.K., "Element wise recognition of continuous speech composed of words from a specified dictionary", *Kibernetika (Cybernetics)*, vol. 7, pp133-143, March-April 1971.

[21] MYERS, C.S. and RABINER, L.R., "A level building dynamic time warping algorithm for connected word recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP* - 29:284-297. April, 1981.

[22] NEY, H. J. and ORTMANNS, S., "Dynamic Programming Search for Continuous Speech Recognition", *IEEE Signal Processing Magazine*, 1999. pp. 64-83.

[23] PICONE, J., "ECE 8463: Fundamentals of Speech Recognition", *Department of Electrical and Computer Engineering Mississippi State University*, On-line available material: http://www.cavs.msstate.edu/hse/ies/publications/courses/ece_8463/lectures/current/.

[24] SPOLTECH Brazilian Portuguese Corpus, February, 2002. Available at: http://www.cslu.ogi.edu/corpora/spoltech/ (18/05/2006).

[25] TEVAH, R.T., "Implementation of a Large Vocabulary Continuous Speech Recognition System for Brazilian Portuguese", *Electric Engineering Program, COPPE, UFRJ*, M.Sc. Thesis, June, 2006. (In Portuguese)

[26] "Corpus de Extractos de Textos Electrónicos NILC / Folha de S. Paulo (Ceten-Folha)". Available at: http://acdc.linguateca.pt/cetenfolha/ (14/11/2005). (In Portuguese)

[27] CIRIGLIANO, R.J.R., MONTEIRO, C., BARBOSA, F.L.F., RESENDE JR., F.G.V, COUTO, L.R., DE MORAES, J.A., "A Set of 1.000 Phonetically Balanced Sentences for Brazilian Portuguese Obtained Using Genetic Algorithms Method", *XXII Telecommunications Brazilian Symposium – SBrT*, 2005. (In Portuguese)

[28] YOUNG, S., "The HTK hidden Markov model toolkit: Design and philosophy", *Cambridge University Engineering Department*, UK, Tech. Rep. CUED/FINFENG/TR152, Sept., 1994. (http://htk.eng.cam.ac.uk).

[29] "Br.ispell Dictionary for Portuguese Language Spoken in Brazil, version 3.0". Available at: http://www.ime.usp.br/~ueda/br.ispell/ (03/01/2006). (In Portuguese)

[30] BARBOSA, F.L.F., PINTO, G., RESENDE JR., F.G.V, GONÇALVES, C.A., MONSERRAT, R., ROSA, M.C., "Grapheme-phone transcription algorithm for a Brazilian Portuguese TTS", *Proceedings of PROPOR*, Faro, Portugal, 2003.

[31] SILVA, D.C., DE LIMA, A.A., Maia, R., BRAGA, D., DE MORAES, J.F., DE MORAES, J.A., RESENDE JR., F.G.V, "A rule-based grapheme-phone converter and stress determination for Brazilian Portuguese natural language processing*", Proceedings of the VI International Telecommunications Symposium (ITS)*, Fortaleza, Brazil, September 2006.

[32] KATZ, S.M., "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", *IEEE Trans ASSP*, Vol. 35, No 3, pp400-401, 1987.

[33] NEY, H., ESSEN, U., KNESER, R., "On Structuring Probabilistic Dependences in Stochastic Language Modeling", *Computer Speech and Language*, Vol. 8, No 1, pp1-38, 1994.

[34] ALCAIM, A., SOLEWICZ, J.A., MORAES, J.A., "Occurrence Frequency of phones and lists of phonetically balanced sentences with the Portuguese spoken in Rio de Janeiro", *Journal of the Telecommunications Brazilian Society (SBrT)*, Rio de Janeiro, v. 7, n. 1, p. 23-41, 1992. (In Portuguese)

Rafael Teruszkin was born in Porto Alegre, Brazil in 1978. In 2002 he graduated in Electronic and Computing Engineering at the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil. In 2006 he received the M.Sc. degree in Electrical Engineering at the Federal University of Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brazil. He is currently leading the Information Technology Engineer Team at Sicpa Product Security S.A. and still holds the position of researcher at the Signal Processing Laboratory in COPPE/UFRJ, both in Rio de Janeiro, Brazil. His research concerns large vocabulary speech recognition.

Fernando Gil Vianna Resende Junior received the B.Sc. degree from Military Institute of Engineering (IME), Brazil, in 1990, and the M.Sc. and Ph.D. degrees from Tokyo Institute of Technology (TIT), Japan, in 1994 and 1997, respectively, all in electrical engineering. Since 1998 he has been with the Department of Electronic Engineering and Computer Science, Polytechnic School, Federal University of Rio de Janeiro (UFRJ), as Associate Professor. Also, since 2003 he has been with the Program of Electrical Engineering, COPPE/UFRJ. His research interests are in the areas of natural language processing, speech synthesis, speech and speaker recognition, and speech coding.