REDUCING COMPUTATIONAL AND MEMORY COST FOR CELLULAR PHONE EMBEDDED SPEECH RECOGNITION SYSTEM

Christophe Lévy[1,2], *Georges Linarès*[1], *Pascal Nocera*[1], *Jean-François Bonastre*[1]

[1] Laboratoire Informatique d'Avignon, Avignon, France
[2] Stepmind SA, Le Cannet, France
{christophe.levy, georges.linares, pascal.nocera, jean-francois.bonastre}@lia.univ-avignon.fr

ABSTRACT

This paper is focused on cellular phone embedded speech recognition. We present several methods able to fit speech recognition system requirements to cellular phone resource. The proposed techniques are evaluated on a digit recognition task using both French and English corpora. We investigate particularly three aspects of speech processing: acoustic parameterization, recognition algorithms and acoustic modeling.

Several parameterization algorithms (LPCC, MFCC and PLP) are compared to the Linear Predictive Coding (LPC) included in the GSM norm. The MFCC and PLP parameterization algorithms perform significantly better than the other ones. Moreover, feature vector size can be reduced until 6 PLP coefficients allowing to decrease memory and computation resources without a significant loss of performance.

In order to achieve good performance with reasonable resource needs, we develop several methods to embed classical HMM-based speech recognition system in cellular phone. We first propose an automatic on-line building of phonetic lexicon which allows a minimal but unlimited lexicon. Then we reduce the HMM model complexity by decreasing the number of (Gaussian) components per state.

Finally, we evaluate our propositions by comparing Dynamic Time Warping (DTW) with our HMM system - in the context of cellular phone - for clean conditions. The experiments show that our HMM system outperforms DTW for speaker independent task and allows more practical applications for the cellular-phone user interface.

1. INTRODUCTION

Automatic speech recognition systems obtain good results in laboratory conditions, but they still require large memory and CPU resources. State of the art speech-to-text systems are usually based on acoustic models composed of several millions of parameters. They also use large lexicon and language models. Moreover, decoding a sentence requires huge amount of computational power (more than 10 times real time on a standard workstation). Embedded speech recognition system on a cellular phone implies to reduce significantly both decoding complexity and model size.

In this paper, we study various techniques for mapping speech recognition system requirements to the limited amount of resource available in a cellular phone.

In GSM standard, the voice coding is based on LPC (Linear Predictive Coding [1]). Using directly this LPC voice coder for speech recognition saves computational costs. We compare this LPC parameterization with classical parameterization algorithms proposed in the literature: Mel Frequency Cepstral Coefficients (MFCC [2]) and Perceptual Linear Predictive coefficients (PLP [3]). At last, we evaluate the influence of acoustic vector size on speech recognition system performance, in a realistic context for cellular phone (isolated words, small vocabulary).

Secondly, we compare Dynamic Time Warping [4] (DTW) and Hidden Markov Model [5] (HMM) approaches.

Finally, we study a potential solution to reduce the HMM complexity, based on a decrease of HMM parameters. In this way, we evaluate the influence of the number of Gaussians per state on the Word Error Rate (WER).

2. CELLULAR PHONE CONTEXT

If the first generation mobile-phones provide only few user services like the recording of some phone numbers, the new generations offer a large set of functionalities. It includes powerful agenda, ring downloads, games, ..., leading to a complex phone-user interface. Due to the size of the cellphones, voice based applications like name dialing, automatic phone number recognition (and dialing) or vocal control will become the basic ones for the customer. If the last generation cellular phones provide more memory and computational resources (and power, which is linked to the previous ones), their resources are still limited compared to the needs of speech recognition engines. The cellphone embedded chip provides :

- few kB of memory (less 4 kB of ROM for our chip),
- a processor around 50 MHz, and
- a Digital Signal Processor (DSP) around 50 MHz.

Due to the application context, cellular phone speech recognition engines have mainly to deal with new names and/or family name. This adds a constraint to the lexicon; it is compulsory to extend or modify the lexicon depending on the user requests (dynamic lexicon). It also implies a phoneme-based recognition and not a global-word based recognition engine.

Finally, the ergonomic point of view leads to short training phase. Generally, the user is asked to pronounce only one repetition for each word of the lexicon.

3. DATABASES AND EXPERIMENTAL PROTOCOLS

Experiments are conducted on an isolated digit recognition task. In order to deal with the ergonomic constraint shown in 2, we selected

a speaker dependent mode where the models are trained with only one repetition of the different digits. We use two corpora:

- the first one is the isolated digit subset of the French corpus BDSONS [6]. We first defined an experimental set composed of 800 digit utterances pronounced by 16 speakers (5 repetitions of the ten digits by speaker). We used 1400 digits from the 14 remaining speakers (10 repetitions of the ten digits by speaker) for speaker independent HMM adaptation (section 5).
- the second is the English corpus TI_DIGITS [7]. It includes 225 speakers divided into two subsets: training (112 speakers) and testing (113 speakers). For each speaker, two utterances of eleven digits (one to nine plus "oh" and "zero") are available. The test subset, respectively the train subset, is composed of 2464 digit utterances, respectively 2486 digit utterances.

The digit z (0-9 for BDSONS or 0a, 0b, 1-9 for TI_DIGITS) of utterance y (0-4 for BDSONS or 0-1 for TI_DIGITS) pronounced by the speaker x will be noted $L_xU_yD_z$. A reference for a given digit is built using only one repetition. We define 3 different experimental protocols:

- "user" protocol: only the test utterances pronounced by the training speaker are used. This protocol corresponds to the classical operating mode of the cellphone (only the owner uses his phone). For BDSONS database, we simulate 80 different speakers by learning a reference on one among the five repetitions, by using the other repetitions (from the same speaker) for the tests and by repeating this process five times. This protocol leads to 3200 tests. To summarize, the learning uses $L_xU_{y1}D_{0-9}$ files when the testing corresponds to $L_xU_{y2}D_{0-9}$ with $y2$ different from $y1$. This protocol is not used for TI_DIGITS corpus.
- "other" protocol: only the test utterances pronounced by the other speakers (different than the training one) are used. This protocol simulates the situation where the user is not the cellphone owner. For BDSONS, we simulate 80 speakers as described bellow but we use all the utterances except the training one for testing. This protocol leads to 60000 digit tests (80 pseudo-speakers * 15 other speakers * 5 utterances * 10 digits). For TI_DIGITS, we obtain 224 virtual speakers (for 112 real speakers in the database training set and two repetitions by speaker). It leads to 556864 digit recognition tests using the 2486 digits corresponding to the testing set.
- "all" protocol: all the utterances different from the training one are used for testing. It leads to 63200 tests for BDSONS. This protocol is not used for TI_DIGITS (in this database training and testing utterances come from different speakers).

4. PARAMETERIZATION

In numeric land-line telephony, the PCM speech coding requires 56/64 Kbits/s transmission rate (8KHz x 7/8 bits). As this rate is expensive, a special codec is used for cellular phone: the Linear Predictive Coding (LPC) with a rate between 13Kbits/s for full-rate and 6.5Kbits/s for half-rate. Using directly the coefficients issued by this codec for speech recognition allows to save computational costs in the phone (and ROM memory).

In order to evaluate this solution, we compare LPC coefficients with two classical speech recognition parameterizations, MFCC and PLP, in the specific context of cellular phone embedded applications. We perform decoding with feature vectors composed of 12 static coefficients augmented by the energy (without delta and acceleration coefficients) issued from LPC, LPCC (Linear Predictive Cepstrum Coefficients), MFCC or PLP parameterization algorithms. For the PLP, we investigate also shorter dimension feature vectors (PLP with only 6 coefficients, called PLP6).

For LPCC computation, we first estimate LPC coefficients with the classical algorithm and then we apply a simple recursion (*cf.* equation 1). In this case, the additional computational cost is limited.

$$Eq.1 : LPCC_i = -LPC_i + \frac{1}{i}\sum_{k=1}^{i-1}(i-k)LPC_kLPCC_{i-1}$$

where LPC_i (respectively LPC_k) is the i^{th} coefficient (respectively k^{th} coefficient) issued from linear prediction codec and where $LPCC_{i-1}$ represents the $(i-1)^{th}$ cepstral coefficient issued from LPC coefficients.

All the experiments are conducted using a DTW-based system with and without applying Mean Subtraction and variance Reduction (MSR - each coefficient mean is set to 0 and each variance is set to 1).

4.1. Results

The results (*cf.* Table 1) show that LPCC (LPC cepstral coefficients) are more efficient than the cellular phone embedded LPC. The WER decreases from 11% to 4.8% for "user" BDSONS protocol without MSR and from 53% to 35% for "all" BDSONS protocol without MSR. The gain is similar for the TI protocol and for experiments with feature normalization (with MSR).

The filter-bank based parameterizations (MFCC and PLP) outperform drastically the LPC based parameterization. The WER (for MFCC and PLP) is always less than 0.3% for "user" protocol compared with the 11% or 4.8% obtained with LPC based parameterizations. Nevertheless, these methods require more CPU resource. The feature normalization (mean subtraction and variance reduction) improves significantly the recognition performance for "all" and "other" protocols, especially using MFCC parameters (from 41% to 27% of WER for TI_DIGIT corpus and from 36% to 15,6% for BDSONS corpus). Nevertheless, we also observe a slight recognition rate degradation using "user" protocol.

The last interesting point to note is the small loss (around 0.60% of absolute WER for "user" protocol) observed using compact feature vectors (PLP6, composed of the first 5 PLP coefficients associated with the energy) compared to the best parameterization.

5. REDUCING HMM RESOURCE NEEDS

In the literature, the two main algorithms used for isolated digit recognition are DTW and HMM. If DTW is well-known for its good performance/resource ratio, it is not able to perform complex tasks, based for example on continuous speech recognition. HMM systems are well known for their performance, especially for continuous speech recognition. Unfortunately, HMM systems use generally very large acoustic models composed of several thousands of parameters.

	BD		TI
	”user”	”all”	”other”
LPC	11.00%	53.84%	65.81%
LPC MSR	14.56%	58.90%	67.51%
LPCC	4.88%	35.29%	42.54%
LPCC MSR	5.19%	25.27%	34.35%
MFCC	0.19%	36.03%	41.66%
MFCC MSR	0.31%	15.60%	27.24%
PLP	0.12%	26.08%	36.29%
PLP MSR	0.28%	23.09%	29.34%
PLP6	0.69%	23.29%	28.09%
PLP6 MSR	0.91%	17.40%	24.83%

Table 1. WORD ERROR RATE OF DTW BASED RECOGNIZER USING SEVERAL ACOUSTIC FEATURES: lpc, lpc with Mean Subtraction and variance Reduction (lpc MSR), lpcc, lpcc MSR, mfcc, mfcc MSR, plp, plp MSR, compact plp (plp6) and compact plp with MSR (plp6 MSR)

Before comparing HMM and DTW systems in the focus of cellular phone embedded application, we propose two solutions to improve HMM technique.

5.1. Experimental conditions

In order to evaluate the performance of the HMM system with the dynamic lexicon (memory resource reduction) and model-size reduction (computation resource reduction and memory save), described in section 5.2.1 and 5.2.2 respectively, we use the protocols defined in section (3) and BDSONS corpus. The signal is parameterized on 12 MFCC coefficients and energy for each frame (20 ms) and MSR normalization (mean subtraction and variance reduction) is applied. The decoding strategy is based on a classical Viterbi algorithm.

The HMM system uses classical left-right context independent phoneme models, composed of 3 emitting states.

The context-independent phoneme models are learned using a three step method:

- Firstly, the models are learned using the French database BREF120 ([8]). This database includes about 40 hours of speech, pronounced by 120 speakers. This training phase is done using EM algorithm and optimizing a Maximum Likelihood (ML) criterion;
- after this first step, the models are adapted to the corpus and the task (BDSONS and digit recognition) using the set of 1400 digits defined in 3[1]. This adaptation is based on MAP (Maximization a posteriori [9]). It gives speaker-independent models.
- lastly, we perform a second model adaptation (still using MAP) in order to obtain speaker-dependent models. This second adaptation phase is achieved using only one training repetition (the ten digits).

5.2. HMM: memory and computational reduction

To embed a HMM speech recognition engine in a mobile phone leads to a drastic reduction in terms of memory and computation resource consumption. To deal with this problem, we propose in this paper two techniques:

[1]speaker involved during this adaptation phase will not be present in sets used for speaker dependent model training nor for testing

- In order to fit cellular phone memory requirements without limiting the potential applications, we propose a dynamic building of lexicon. A dynamic lexicon allows the user to use specific words, like family names (which is compulsory, for example, in name dialing application), without leading to a huge lexicon. Here, the dynamic lexicon is automatically built thanks to an acoustic decoding pass, which produces the phonetic transcription of each new word.
- Due mainly to the embedded processor limits, we reduce the model complexity. We study the correlation of the number of Gaussians per state and the WER (from 128 to 1 Gaussian per state).

5.2.1. Dynamic lexicon results

Table 2 presents the results obtained using our dynamic lexicon. The performance, in terms of WER, is compared to a classical static lexicon and is given for the three protocols (”user”, ”other” and ”all”). Static and dynamic lexicons achieve very similar performance. Nevertheless, the dynamic lexicon leads to a significative gain in terms of memory and functionalities. A different application, like name dialing, will certainly highlight the interest of the dynamic lexicon compared to a digit recognition task (which relies on a small lexical complexity).

	user	other	all
Static lexicon	1.38%	5.38%	5.18%
Dynamic lexicon	1.88%	6.71%	6.46%

Table 2. WORD ERROR RATE ON BDSONS DATABASE USING STATIC AND DYNAMIC LEXICON: 16 components by mixture. 3200 tests for ”user”, 60000 for ”other” and 63200 for ”all”

5.2.2. Model size reduction results

In order to evaluate the influence of HMM model size on WER, we test 5 configurations from 128 to 1 Gaussian per state. The results (*cf.* Table 3) show, as expected, that reducing the number of components in the model increases the WER. Nevertheless, a small loss (less than 0.2% of WER) occurs when using the higher order models compared to middle size ones. It comes from the small amount of adaptation data available for the two adaptation steps (1400 digits for the first one and only one repetition of the ten digits for the speaker adaptation).

HMM models with 16 components by state obtain the more interesting performance/resource ratio for the targeted application.

	user	other	all	Model size
128g/state	1.88%	5.88%	5.68%	360 kB
64g/state	1.66%	6.84%	6.58%	180kB
16g/state	1.88%	6.71%	6.46%	45kB
4g/state	12.41%	35.60%	34.43%	11kB
1g/state	21.09%	74.56%	71.85%	3kB

Table 3. HMM SYSTEM WER WITH DIFFERENT NUMBERS OF COMPONENTS BY MIXTURE/STATE:. All tests are performed with a dynamic lexicon.

5.3. DTW *vs.* HMM

For "user" protocol, DTW obtains around 0.31% of WER (*cf.* Table 1) using MFCC MSR parameterization when HMM gets between 1.66% to 21.09%, depending on the model size (*cf.* Table 3).

For "all" protocol, HMM outperforms DTW with a WER around 5%, to compare with around 15% of WER for DTW.

Looking at the amount of computational time needed (*cf.* Table 4), both approaches seem relatively close when model size reduction is used. HMM with less than 16 components by mixture need less than 0.4 times the real time for the decoding and DTW around 0.31 times.

	HMM sys.					DTW
	1g	4g	16g	64g	128g	sys.
time (s.)	0.30	0.31	0.40	0.79	1.33	0.31

Table 4. COMPUTING TIME NEEDED FOR DECODING 1 SECOND OF SIGNAL: results given for DTW and HMM with 1 to 128 Gaussians per state

6. CONCLUSION

In this paper, we focused on embedding a speech recognition application in a mobile phone. In this context, the speech recognition engine must respect some constraints in terms of computational and memory costs.

We showed that LPC cepstral coefficients (LPCC) issued from the cellular phone embedded LPC codec allow a good performance/cost ratio. Nevertheless, filter-bank based parameterizations (MFCC and PLP) outperform significantly both LPC and LPCC parameterizations. We proposed a dynamic lexicon which allows smaller but unlimited lexicon without a significant loss in terms of WER. Then, we showed that reducing the number of components by state until 16 Gaussians maintains a good perfomance level and allows to save significantly memory and computational costs.

Finally, we presented a very compact HMM system which used around 45kB of memory and no more computational resource than DTW approach. This HMM system obtained a satisfactory level of performance and authorizes a large set of applications, in the context of ergonomic cellular-phone vocal interfaces.

7. REFERENCES

[1] Tremain T.E., "The government standard linear predictive coding algorithm : Lpc10," *Speech Technology*, vol. 1, no. 2, pp. 40–49, april 1982.

[2] Davis S.B. and Mermelstein P., "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, August 1980.

[3] Hermansky H., "Perceptual linear predictive (plp) analysis of speech," *Journal of Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, April 1990.

[4] Bellman R., *Dynamic Programming*, Princeton University Press, 1957.

[5] Rabiner L.R., "A tutorial on hidden markov models and selected applications in speech recognition," *IEEE transactions Speech Audio Processing*, vol. 77, no. 2, pp. 257–285, february 1989.

[6] Esknazi M. Mariani J. Carr R., Descout R. and Rossi M., "The french language database : defining, planning and recording a large database," in *proceeding of ICASSP*, San Diego, 1984.

[7] Leonard R.G., "A database for speaker-independent digit recognition," in *proceedings of ICASSP*, San Diego, 1984, vol. 3.

[8] Gauvain J.L. Lamel L.F. and Esknazi L., "Bref, a large vocabulary spoken corpus for french," *EUROSPEECH*, 1991.

[9] Gauvain J.L. and Lee C.H., "Maximum a posteriori estimation for multi-variante gaussian mixture observations of markov chains," *IEEE transactions on speech and audio processing*, vol. 2, pp. 291–298, 1994.