SPEAKER-DEPENDENT SPEECH RECOGNITION BASED ON PHONE-LIKE UNITS MODELS — APPLICATION TO VOICE DIALING

Vincent Fontaine and Hervé Bourlard

Faculté Polytechnique de Mons, 31, Boulevard Dolez, B-7000 Mons, BELGIUM
Tel: ++ 32 65 374176 - Fax: ++32 65 374129
Email: fontaine,bourlard@tcts.fpms.ac.be

ABSTRACT

This paper presents a speaker dependent speech recognition with application to voice dialing. This work has been developed under the constraints imposed by voice dialing applications, i.e., low memory requirements and limited training material. Two methods for producing speaker dependent word baseforms based on Phone Like Units (PLU) are presented and compared : (1) a classical vector quantizer is used to divide the space into regions associated with PLUs; (2) a speaker independent hybrid HMM/MLP recognizer is used to generate speaker dependent PLU based models. This work shows that very low error rates can be achieved even with very simple systems, namely a DTW-based recognizer. However, best results are achieved when using the hybrid HMM/MLP system to generate the word baseforms. Finally, a realtime demonstration simulating voice dialing functions and including keyword spotting and rejection capabilities has been set up and can be tested online.

1. INTRODUCTION

Voice dialing is typically based on speaker dependent speech recognition systems in which each speaker can easily define his/her own personal repertory containing the set of commands or keywords that will be used later on to automatically dial phone numbers. The set up of such a system is usually based on two phases:

1. Enrollment phase: The user pronounces several times (in our case, twice) each of the keywords and provides the system with their associated phone number. Ideally, this enrollment should be as fast and flexible as possible.
2. Recognition phase: The user pronounces a keyword and the system automatically dials the associated phone number. Furthermore, if several speakers belongs to the same directory, the system should be able to also identify the speaker in the case of similar keywords.

One simple solution towards fast enrollment is based on standard template matching approaches (simply storing sequences of acoustic vectors associated with each utterance) and dynamic time warping (DTW). This approach however suffers from major drawbacks, namely high memory storage requirements and poor robustness against the variability of the test conditions.

Alternative solutions to the straightforward DTW approach have been proposed in the past. In [4], HMMs are automatically derived from the keywords pronounced by the user during the enrollment phase. The training of such models however require a large number of examples, which makes the system less flexible and less attractive to the user. Another solution [6] somewhat related to what will be investigated in the current paper is to use the symbolic string produced by a speaker independent speech recognizer to represent the keyword. Compared to DTW, this leads to nearly equivalent recognition rates with the advantage of a drastic reduction of the memory requirements.

In this paper, two methods for automatically generating some kind of speaker specific models based on phone-like units (PLU) are tested:

1. Section 3. makes use of a standard vector quantizer to design the PLUs.
2. Section 4. uses a speaker independent hybrid hidden Markov model (HMM) & multilayer perceptron (MLP) system to generate PLU-based speaker dependent models.

Although some of the approaches used in this paper have already been investigated in the past (see, e.g., [2] in the case of DTW with vector quantization), they are now tested in the particular framework of voice dialing application and include: (1) state-of-the-art acoustic features (see Section 2.), and (2) keyword spotting capabilities (see Section 5.).

2. ACOUSTIC FEATURES AND DATABASES

All the experiments reported in this paper used 12 rasta-plp cepstral coefficients [5] extracted from 30 ms speech frames shifted by 10 ms. The cepstral coefficients were liftered by a sine window:

$$w[i] = 1 + \frac{N}{2} sin(\frac{\pi i}{N})$$

with $N = 12$ in our case. As discussed at the end of Section 3., the use of delta features did not seem to improve performance in the case of the particular models used here.

Due to the lack of appropriate databases to study voice dialing systems, we decided to test our algorithms on the BDSONS database designed for speaker dependent speech recognition and consisting of the 10 isolated French digits pronounced 40 times by 23 speakers (400 digit utterances/speaker). For each speaker, 20 utterances (the first

two utterances of each digit) were retained to create the word models (i.e., to simulate fast enrollment of new keywords). In this paper, this database will be referred to as *enrollment database*. The 380 remaining (digit) utterances were used for testing[1] and are referred to as *test database*.

Since voice dialing systems should ideally be independent of the language and of the training database, we decided to train the codebooks or the neural network on the TIMIT database, i.e., a database recorded in another language and designed for another speech recognition task. In this paper, this database will be referred to as *training database*.

3. DYNAMIC PROGRAMMING AND VECTOR QUANTIZATION

The first kind of phone-like units that have been tested were built up from a standard K-means vector quantizer, based on standard Euclidian distances[2], dividing the acoustic parameter space into regions representing PLUs. In this case, a K-means clustering was applied to the whole set of the TIMIT acoustic vectors (training database).

During enrollment (transforming the enrollment utterances into sequences of PLUs), PLU-based word models are built up by first replacing each vector of the enrollment utterances by the label of the closest prototype. The resulting label sequences are then further processed according to the following simple rule: sequences of the same label are reduced to sequences of length n, indicating stationary parts of speech, while transition parts are left unchanged. For example, if we suppose $n = 2$, the label sequence {2 2 2 7 5 9 4 4 4 4 1 8 2 6 6 6 1 4} will be turned into {2 2 7 5 9 4 4 1 8 2 6 6 1 4}. The parameter n will be referred to as the sequence *compression factor* in the sequel. The resulting compressed label sequence was stored as the word model, resulting in a significant reduction of the memory requirements (compared to storing the acoustic vector sequences), with an average of 30 to 50 bytes per word. Another consequence of the sequence compression procedure is the gain in CPU time for the dynamic programming that is proportionnal to the storage gain. Also, as already shown in [2], this kind of modelling also has a smoothing effect (over time and frequency) that can result in slightly better recognition performance. The enrollment procedure is illustrated in Figure 1.

The speaker dependent character of the models gives the ability to the system to discriminate keywords pronounced by different speakers. This case is typically encountered when two persons introduce the same keyword (e.g. "Mom") in the same enrollment database but with different phone numbers associated to this keyword.

During recognition, and unlike some methods proposed in the past (and unlike discrete HMMs), the input vectors are not quantized[3]. it is indeed not necessary to perform vector quantization during the recognition phase since it will only introduce unnecessary computation and distortion. Instead, local distances of the dynamic programming grid are computed between the test vectors and the centroids corresponding to the labels of the models.

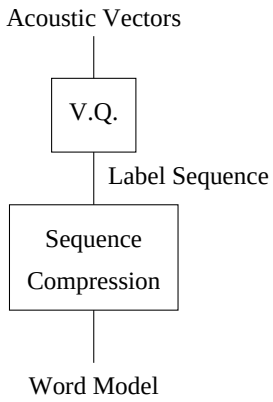

Figure 1. Enrollment procedure : Acoustic vectors are first quantized to produce label sequences that are further processed to produce word models.

The results presented in Table 1 show that very high accuracy has been obtained for this task especially when the codebook is designed with 256 centroids. We tried to improve these results by adding the first derivatives of the cepstral coefficients and the first and second derivatives of the log-energy. These additional parameters were quantized by separate codebooks (4 codebooks in total) and the training utterances were then modeled by 4 sequences of labels. The local distances of the dynamic programming grid were computed as a weighted sum of the distances between the vector components and the nearest centroid of the corresponding codebook. Several weighting configurations of the distances were tested but never lead to significant improvement of results obtained with static parameters only. A possible explanation to this is that the word models are so detailed that they implicitely include a good description of the dynamics.

Tests in a multi-speaker environment were also performed to study the ability of the system to discriminate between speakers. The models of the ten digits for 5 speakers were considered as the enrollment database and the test set was composed of the remaining utterances for these 5 speakers. Table 1 shows that discrimination between speakers (and keywords !) becomes pretty good as the size of the codebook increases. This is not surprising since more refined the PLU space becomes and more the speaker specific characteristics are captured by the system.

Table 2 presents the influence of the compression of the label sequences on the error rate. We can observe that the best results are obtained without compression but that the error rate is not quite sensitive to the compression parameter. Even for $n = 2$, the error rate is only of 2.3% (to be compared to 1.9% without compression) while storage and computation requirements are reduced by approximately 40%.

[1] We note here that, apart from the number of active words, this task could be harder than voice dialing applications since (1) keywords are quite short, and (2) some of them are quite confusable, like "cinq" and "sept".

[2] Mahalanobis distances were also tested but never led to significant improvements.

[3] This has been tested and has been shown to lead to significantly lower performance.

# Classes	Mono-speaker	Multi-speaker
64	2.7 %	15.2 %
128	1.4 %	10.5 %
256	1.1 %	10.2 %

Table 1. Influence of the number of VQ classes on the error rate. In the mono-speaker case, the tests have been performed independently for each of the 23 speakers (the enrollment set only contained models of the tested speaker) and the results have been averaged. In the multi-speaker case, models of 5 speakers were recognized simultaneously.

Compression factor	Error Rate
2	2.7 %
3	2.6 %
4	2.4 %
5	2.3 %
∞	1.9 %

Table 2. Influence of the compression factor applied to stationary parts of label sequences. The tests have been performed in the mono-speaker task. A compression factor of ∞ leaves the label sequences unchanged.

4. HYBRID HMM/MLP VOICE DIALING

The approach discussed now follows the same principle than the method presented in Section 3., with the difference that the (unsupervised) K-means clustering is replaced by the (supervised) training of a multilayer perceptron (MLP) as used in the framework of speaker independent hybrid HMM/MLP systems [3].

As in hybrid HMM/MLP systems, the MLP network is trained in a supervised way (possibly within embedded Viterbi) to yield posterior probabilities of phone classes (associated with the MLP outputs) conditioned on the input vectors presented to the network. This training is done in a speaker independent mode, on TIMIT in the current work. In the current system though, as opposed to standard HMM/MLP recognizers, the trained network is then used for two different goals:

1. To automatically infer the model topology (in terms of PLU sequence) of the voice dialing keywords.
2. To compute local DTW distances between the test utterance and the infered models.

The sequence of PLUs associated with each enrollment utterance was then generated in two steps: (1) replacing each frame by the label of the phonemic class associated with the highest posterior probability observed on the MLP outputs, and (2) applying the time compression scheme as used in Section 3.. The enrollment procedure as applied for hybrid voice dialing is illustrated in Figure 2.

Recognition was then performed by dynamic programming where the local distances between each input vector x_n of the test utterance and the PLUs composing the reference words were defined as the Euclidian distance between

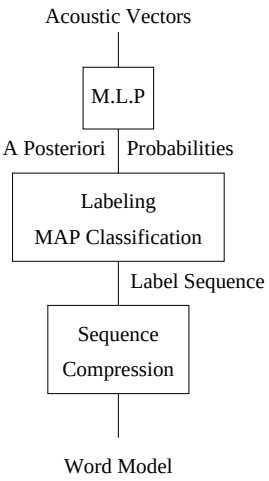

Figure 2. Enrollment procedure : Acoustic vectors are labeled according to a MAP classification criterion. The label sequences are further processed to produce word models as in the case of the VQ based recognizer.

the vector of *a posteriori* probabilities generated by the network for x_n and the vector of *ideal a posteriori* probabilities corresponding to the PLUs of the training utterances[4].

The vector of *ideal a posteriori* probabilities for the PLU q_i, noted $d(q_i)$, corresponds actually to the desired outputs as presented to the network during its training phase [3]:

$$d_k(q_i) = \delta_{k,i}, 1 \leq k \leq K$$

where K is the number of PLUs and $\delta_{k,i}$ is the usual Kronecker delta function, which is only nonzero (and equal to 1) when $k = i$. The local distance between x_n and the PLUs q_i can then be expressed as:

$$D(x_n, q_i) = \sum_{k=1}^{K} (g_k(x_n) - d_k(q_i))^2$$

where $g(x_n)$ represents the output probabilities of the MLP.

As in the previous section, the MLP was trained on the TIMIT database (English) and tested (with enrollment) on the BDSONS database (containing French digits). Recognition results are presented in Table 3 and show that hybrid systems slightly outperform the results obtained with the best configuration of the DTW based recognizer. These results also indicate that it is not necessary to divide the MLP output probabilities by the prior probabilities of the MLP output classes (as usually done in standard HMM/MLP systems). This can be explained by the fact that the word baseforms (topologies) are directly inferred from the MLP.

Multi-speaker tests have also been performed for the hybrid systems. The tests have been conducted in the same way as for the DTW based recognizer and indicate that the hybrid system is not able to discriminate the speakers as

[4]Kullback-Leibler distance has also been tried but never outperformed results obtained with an Euclidean distance.

Network size (a-b-c)	1-speaker Div. by priors	1-speaker No Div. by priors	5-speaker No Div. by priors
182-243-64	1.7 %	1.4 %	-
117-500-64	1.3 %	0.8 %	25.3 %

Table 3. Speaker dependent recognition error rates with neural networks. (a-b-c) represent the number of input nodes, hidden nodes and output nodes, respectively. Feature vectors of the first network included dynamic parameters (7*26: 12 rasta - 12 Δ rasta - Δ log-energy - $\Delta\Delta$ log-energy) while only static parameters (9*13: 12 rasta - log-energy) were used for the second network. In the mono-speaker case (1-speaker), the tests have been performed independently for each of the 23 speakers (the enrollment set only contained models of the tested speaker) and the results have been averaged. In the multi-speaker case (5-speakers), models of 5 speakers were recognized simultaneously.

Compression factor	Error Rate
2	0.8 %
3	0.9 %
4	0.8 %
5	0.8 %
∞	1.4 %

Table 4. Influence of the compression factor applied to stationary parts of label sequences. The tests have been performed using the hybrid HMM/MLP system in the same conditions as for the DTW based recognizer.

good as VQ derived PLU. This can be explained by the fact that the MLP is trained in a supervised way to learn speaker independent realisations of phonemes. Therefore, the speaker dependent models of a keyword will be close to each other for all speakers.

Table 4 shows the influence of the label sequence compression factor on the error rate. Here, it is quite remarkable (and particularly interesting) to observe that, unlike the previous DTW based system, sequence compression always results in a significant reduction of the error rate. This allows to reduce significantly memory storage (about 60% storage gain) without any degradation of the error rate.

5. KEYWORD SPOTTING

The two algorithms discussed in this paper have been adapted to accommodate keyword spotting by using a slight adaptation of the method presented in [1] and referred to as "on-line garbage". In this case, a fictitious "garbage" unit is introduced in the dynamic programming for which the local score is computed as the average of the N-best distances between each of the test frame and the reference labels. Keyword spotting is then performed simply by adding this garbage unit at the beginning and at the end of each word model (syntax allowing "garbage-keyword-garbage", or "garbage-garbage" in the case of rejection).

6. CONCLUSION

In this paper, two approaches for speaker dependent speech recognition tasks based on generation of phone-like units sequences have been tested and compared.

In both cases, very high accuracies can be achieved. A realtime demonstration of a voice dialing system based on the technology discussed in this paper and including rejection and keyword spotting capabilities has been implemented and can be tested at +32-65-37.41.77 (Belgian site). Information on the use of the demonstration system is available on our web site at the address: http://tcts.fpms.ac.be/speech/softdial.html

REFERENCES

[1] J.-M. Boite, H. Bourlard, B. D'hoore, and M. Haesen, "A new approach towards keyword spotting," in *Proceedings of EUROSPEECH93*, pp. 1273-1276, 1993.

[2] H. Bourlard, H. Ney, and C.J. Wellekens, "Connected Digit Recognition using Vector Quantization," *IEEE Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. 16.10.1-4, 1984.

[3] H. Bourlard and N. Morgan, *Connectionist Speech Recognition*, Kluwer Academic Publishers, 1994.

[4] D. Geller, R. Haeb-Umbach, and H. Ney. "Improvements in speech recognition for voice dialing in the car environment," in *Proceedings of Speech Processing in Adverse Conditions*, pp. 203-206, 1992.

[5] H. Hermansky, and N. Morgan, "RASTA processing of speech," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 4 pp. 578-589, 1994.

[6] N. Jain, R. Cole, and E. Barnard. "Creating speaker-specific phonetic templates with a speaker-independent phonetic recognizer: Implications for voice dialing," in *Proceedings of ICASSP96*, pp. 881-884, 1996.