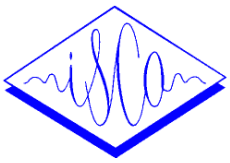

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

The automatic speech recognition engine ESPERE : experiments on telephone speech

Dominique Fohr[1], *Odile Mella*[1], *Christophe Antoine*[2]

(1) LORIA-CNRS & INRIA Lorraine, BP 239, F54506, Vandoeuvre, France
(2) MIC2, 28 Bd Kellermann, F75013, PARIS, France
{fohr, mella, antoinec} @loria.fr

ABSTRACT

This paper presents our automatic speech recognition engine ESPERE and several results obtained from experiments on telephone speech.
ESPERE (Engine for SPEech REcognition) is a HMM-based toolbox for speech recognition allowing the user to choose the modeled unit (word, phone, triphone), define the topology of every Hidden Markov Model, train the models with the Baum-Welch algorithm and evaluate the recognition accuracy on speech databases.
To validate the ESPERE toolbox, we have conducted tests on real world data: the recognition of a three-digit code to access a call center. We have investigated the influence of some parameters and some preprocessing algorithms.
Finally, combining the best parameters, the recognition score reaches 96.4% at the word level and 92.1% at the sentence level.

1 INTRODUCTION

This paper presents our automatic speech recognition engine ESPERE and several results obtained from experiments on telephone speech.
In the first section, we describe the main features of ESPERE: the acoustic front-end, the training module and the recognition engine. Then we present the telephone application used for the validation of our toolbox. Finally, we show the influence of some front-end parameters on the recognition results

2 ESPERE

ESPERE (Engine for SPEech REcognition) is a HMM-based toolbox for speech recognition which is composed of three processing stages: an acoustic front-end, a training module and a recognition engine.

2.1 The acoustic front-end

The parameterization is based on MFCC parameters. However the user can customize this parameterization: the size of the analyzing window, the shift of this one, the number of the triangular filters, the lower and upper frequency cut-offs of the filterbank and the number of the cepstral coefficients. Moreover, the user can remove C0, the first MFCC, or replace it by the energy computed as the log of the signal energy. Finally, the delta and acceleration coefficients can be added. The acoustic front-end integrates a voice activity detector (VAD).

2.2 The training module

The core of this module uses the Baum-Welch re-estimation algorithm with continuous densities. The user can define the topology of the Hidden Markov Models: the number of states and the allowed transitions. The modeled unit can be a word, a phone or a triphone. These units are trained using either an isolated training (the boundaries of every unit are specified) or an embedded training (only the string of labels is specified).
The number of probability density function (pdf) per state is determined during the training phase: the number of mixture components is repeatedly increased until the desired value by a splitting mechanism.
With regard to triphones, we build a decision tree to reduce the number of mixture pdfs and we then select the states to be tied.

2.3 The recognition engine

The recognition engine implements a one-pass time-synchron algorithm. It requires the HMMs trained as described in previous stage, the lexicon of the application and a grammar. The structure of the lexicon allows the user to give several pronunciations per word. The grammar may be a word-pair or a bigram. In order to speed up the recognition, a pruning threshold can be applied.

In conclusion, the modular architecture of ESPERE and its large facilities of customization provide a research toolkit which allows the test of new algorithms, models and acoustic parameters. Besides, we have added a labeling tool in order to supply huge labeled databases to the training module.
The implementation of ESPERE contains more than 20000 C++ lines and it runs on a PC-Linux or PC-Windows.

3 A TELEPHONE REAL-WORLD TASK

To validate the ESPERE toolbox, we have conducted tests on real world data: the recognition of a three-digit code to access a call center. The caller can utter either isolated or connected digits. The application database, MIC2 contains 165 French three-digit codes.
In the telephone speech recognition, we are faced with specific problems: limited bandwidth, different types of handsets and various transmission channels. Furthermore, the mismatch between training and test conditions yields a degradation of performances. In order to reduce this mismatch, we have trained the HMM models with the telephone speech database SPEECHDAT: 1000 speakers have been recorded through the telephone network, at 8kHz. The number of occurrences of digits was enough to train global models (one per word). We have used 800 speakers of SPEECHDAT database for the training, and, 200 speakers of SPEECHDAT database and all the speakers of MIC2 database for the test.

Using ESPERE, we have built the recognizer as follows:

- the front-end: 32ms frames with a frame shift of 8ms, each frame is passed through a set of 24 triangular band-pass filter resulting in a vector of 35 features, namely 11 static mel-cepstral coefficients (C0 was removed), 12 Δ and 12 $\Delta\Delta$ coefficients ;
- the HMM topology: every digit and the pause are modelized by continuous density mixture. Models are left-to-right with no skip state transitions ;
- the grammar: a word-pair allowing a digit sequence of any size.

First, we have tried to find the optimal number of states per word (with 4 pdf per state).

Number of states per word	Test database : SPEECHDAT	
	Word accuracy	Correct sentence
3	85.7 %	40.9 %
6	92.4 %	60.5 %
9	94.3 %	69.4 %
12	95.3 %	73.2 %
15	95.3 %	71.8 %

In the above table, we have reported the word accuracy and the sentence recognition rate for the SPEECHDAT test. The average number of digits per sentence is 7.2 .
Thus, we have decided to use 12 states for every word in the following experiments,
Then we investigate the influence of the number of states for the silence:

Number of states for silence	Test database : SPEECHDAT	
	Word accuracy	Correct sentence
1	92.6 %	61.2 %
3	94.0 %	65.6 %
6	94.7 %	70.5 %
9	95.0 %	72.5 %
12	95.3 %	73.2 %
15	95.2 %	72.5 %

As shown in the above table, 12 states for silence gives best results.
Finally, we want to optimize the number of mixture per state:

Number of pdf per state	Test database : SPEECHDAT	
	Word accuracy	Correct sentence
1	90.1 %	54.6 %
2	94.0 %	69.4 %
4	95.3 %	73.2 %
8	96.3 %	79.4 %
16	96.2 %	78.7 %
32	96.0 %	77.0 %
64	96.3 %	80.1 %

So, we decide to use 8 pdf per state because it is the best compromise between accuracy and computation load. In the following experiments, we use 12 states for each HMM word model and 8 pdfs per state

With a mixture of 8 pdfs per state, the recognition accuracy at the sentence level was 79.4% on the testing part of SPEECHDAT database and 18% on the MIC2 database[1]. Despite that we have carefully chosen a telephone speech database for the training, a huge degradation of the recognition accuracy can be noticed on the application database.
The gap between the two scores could be explained by the mismatch between the recording conditions of the both databases. Three main causes can be listed:

- the acquisition hardware was different,
- the files of the SPEECHDAT database were correctly segmented whereas the quality of the speech/non speech detection was bad and kept a long portion of non speech signal after the three-digit code in the MIC2 database,
- the speech signal of the MIC2 database had a low signal to noise ratio due to background noise as music, breath, clicks, tapping, babble,...

[1] There are only three digits per sentence in the MIC2 database

In order to reduce the bad effects of these facts and to increase the accuracy on the MIC2 database, we have investigated the influence of some parameters and some preprocessing algorithms. It will be presented in the next section.

4 EXPERIMENTS

We have considered the recognizer described in the previous section as our "baseline" system. We have then tuned or modified some parameters of this recognizer and we have tested these modifications on both databases.

4.1 Influence of the training

First, we have studied the influence of the quality of the labeling used by the training module. The training portion of SPEECHDATA is used as follows:

(a) only the orthographic transcription of every digit sequence is provided to the training module. Thus an optional pause is inserted between two digits. It is the baseline;
(b) the true sequence of the words uttered (pauses included) is provided ;
(c) all the boundaries of the digits and of the pauses are provided.

Training	Test database			
	SPEECHDAT		MIC2	
	Word	Sentence	Word	Sentence
(a)	93.4 %	66.0 %	36 %	10 %
(b)	95.5 %	75.9 %	45 %	22 %
(c)	96.3 %	79.4 %	44 %	18 %

The above table shows the recognition accuracy on both databases at the word and sentence level. The labeling (b and c) of the databases improves the recognition scores. More accurate is the labeling (c) better is the result.
In the following experiments, we have decided to train the models with the most accurate labeling (c).

4.2 Influence of the parameterization

To increase the recognition accuracy on the application database MIC2, we have found out the acoustic parameters which are less influenced by the acquisition hardware.

Filter frequencies

To avoid to take into account information located outside of the telephone bandwidth, we have moved the upper and lower cutoff frequencies of the MFCC triangular filters: 300 Hz for lower cutoff frequency of the first filter and 3400 Hz for the upper cutoff frequency of the last one.

Bandwidth	Test database			
	SPEECHDAT		MIC2	
	Word	Sentence	Word	Sentence
[0 – 4000 Hz] (baseline)	96.3 %	79.4 %	44 %	18 %
[300 – 3400Hz]	95.0 %	73.2 %	71 %	38 %

As it could be expected, the modification of the filter frequencies, result in a slight difference for the SPEECHDAT scores, in comparison, we notice a dramatic improvement for the MIC2 scores. It may be due to the presence of an artifact during the data acquisition of one of the two databases.

Cepstral Mean Subtraction

It is well known that Cepstral Mean Subtraction (CMS) is one of the methods to solve the problem of channel distortion. As shown in the following table, it was successful !

CMS	Test database			
	SPEECHDAT		MIC2	
	Word	Sentence	Word	Sentence
Without CMS	96.3 %	79.4 %	44 %	18 %
With CMS	96.5 %	80.8 %	85 %	74 %

Reduction of the number of parameters

As for a real-time application, it could be worth to speed up the recognition process and to save memory. Thus we have tried to reduce the number of the parameters, firstly decreasing the number of static MFCC coefficients form 12 to 8, secondly removing the $\Delta\Delta$ coefficients.

MFCC coefficients	Test database			
	SPEECHDAT		MIC2	
	Word	Sentence	Word	Sentence
11 static[2] + 12 Δ + 12 $\Delta\Delta$	96.3 %	79.4 %	44 %	18 %
7 static[2] + 8 Δ + 8 $\Delta\Delta$	95.9 %	75.9 %	41 %	15 %
11 static[2] + 12 Δ	95.0 %	72.8 %	33 %	9 %

The above Table shows that taking 8 coefficients instead of 12 yields to a slight degradation of the recognition score. Therefore, it could be a good alternative for embedded applications.
Another way to speed up the recognition process is to reduce the frame rate, *ie* to increase the frame shift.

[2] C_O was removed from the static coefficients

Frame shift	Test database			
	SPEECHDAT		MIC2	
	World	Sentence	Word	Sentence
64 ms (baseline)	96.3 %	79.4 %	44 %	18 %
96 ms	95.3 %	74.6 %	54 %	20 %
128 ms	92.7 %	64.3 %	64 %	34 %

We can notice that the results differ according to the test database. Regarding SPEECHDAT, the scores decrease because we suppose that the Viterbi algorithm has less information and can also omit some short digits. On the contrary, for MIC2 database, the less the number of frames there is, the less the number of insertions we get. Indeed, the low baseline scores on MIC2 database are due to a lot of insertions of short digits (like six) in place of some noise.

4.3 Speech/non speech detection

The table below shows the great importance of having a good voice activity detector (VAD). This detector consists of a five-state automaton (silence, speech presumption, speech, plosive, speech continuation). The transitions are controlled by a combination of some energy thresholds and some duration constraints [1].

	Test database :MIC2	
	World	Sentence
baseline	44 %	18 %
Baseline + VAD	56 %	30 %

The improvement of the recognition accuracy is due to the reduction of insertions mainly at the end of the utterance.

4.4 Extra speech models

Another way to reduce the insertions in the noisy part of the signal is to add several specific models for noise: a model for short noises like tapping, one for babble, one for breath and one for all the other noises.

	Test database: MIC2	
	Word	Sentence
10 digits + silence (baseline)	44 %	18 %
10 digits + silence + 4 noise models	83 %	64 %

4.5 The adapted grammar

During all the previous experiments, we have used a word-pair grammar allowing a digit sequence of any size whereas the application task requires only sequences of three digits. As it can be expected, using an adapted grammar allowing only three digits with optional pauses gives better results.

	Test database: MIC2	
	Word	Sentence
Word-pair grammar	44 %	18 %
Adapted grammar	62 %	42 %

4.6 The final recognizer

Following these experiments, we have decided to combine:

- the best acoustic parameters: 12 static MFCC, 12 Δ and 12 $\Delta\Delta$ coefficients, the [300-3400Hz] band-pass filter, the Cepstral Mean Subtraction and a 128 ms frame shift;
- the voice activity detector;
- the extra speech models;
- the adapted grammar.

	Test database: MIC2	
	Word level	Sentence level
Baseline	44 %	18 %
Final recognizer	96 %	92 %

5 CONCLUSION

This paper describes our new speech recognition toolbox ESPERE. With this toolkit we have built a recognizer for a telephone application and we have optimized the parameters of the front-end. The main conclusion of this paper is the following: some modification in the parameterization (like CMS or filter bank) gives small improvements when the train and test database are recorded in the same conditions but gives dramatic improvements when there is a mismatch between train and test conditions.

REFERENCES

1. Karray L. and Monné J. “Robust speech/non-speech detection in adverse conditions based on noise and speech statistics” , ICSLP 98, Sydney, Dec 1998