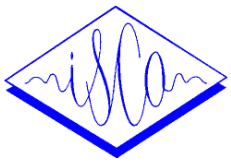

3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

REAL-TIME, NEURAL NETWORK-BASED, FRENCH ALPHABET RECOGNITION WITH TELEPHONE SPEECH

P. Schmid [†], *R. Cole* [†], *M. Fanty* [†], *H. Bourlard* [‡], *and M. Haessen* [‡]

[†] Center for Spoken Language Understanding
Oregon Graduate Institute
20000 N.W. Walker Road
P.O. Box 91000
Portland, OR 97291-1000, USA

[‡] Lernout & Hauspie Speech Products
Rozendaalstraat, 14
B-8900 Ieper, BELGIUM

ABSTRACT

We describe a real-time speaker-independent French alphabet recognizer that performs with sufficient accuracy for commercial use. The system (a) digitizes a sequence of letters separated by brief pauses and computes a RASTA-PLP spectral representation, zero-crossing rate and peak-to-peak amplitudes of the waveform; (b) uses a neural network to assign 23 phonetic category labels to successive time frames; (c) performs an initial segmentation of the speech by mapping the phonetic label scores for each frame to pronunciation models for each letter using a modified Viterbi search; (d) performs a second classification of each hypothesized letter using the segment boundaries provided by the first-pass segmentation, producing a set of 26 letter scores plus a score for the category "Not-A-Letter"; and (e) uses the letter scores (plus the score for the category "Not-A-Letter") to identify the spelled word from a data base. The system has been evaluated on calls that were not used for training either network. The system achieved 84.4% first choice letter recognition accuracy on the test set. The system has also been evaluated on 84 spelled names from different callers where it achieved 92.8% correct recognition of the 84 spelled names contained in a database of 50,000 names. The final system has been optimized to run in real-time on a PC-board based on a single DSP TMS320C30. The two passes described above are performed in real-time by the DSP while the name search (up to 50,000 names) is performed (as letters are recognized) by the PC.

1. INTRODUCTION

Recognizing the letters of the French alphabet is a challenging task for computer speech recognition, because of the acoustic similarity of many of the letter pairs (e.g., M/N, B/D V/G, F/S and P/T). This task is even more difficult for telephone speech, because of bandwidth limitations and distortions introduced by the communication channel.

Research using a segment-then-classify approach has led to accurate recognition of spoken English letters for both high quality and telephone speech [1,2]. In this approach, segment boundaries are located in a first pass, and the letters are classified in a second pass.

In the first pass classification, individual time frames are assigned phonetic category labels by a neural network classifier. Because the goal of this pass is to only perform an accurate segmentation, acoustically similar phonetic categories, such as [b]-[d], [p]-[t]-[k], [m]-[n] or [f]-[s], are combined into a single class. A dynamic programming search provides a segmentation by mapping the output scores of the first pass classification to pronunciation models for each letter. The second pass then computes segment-based features, and reclassifies the letters, using acoustic features that are designed to perform the fine phonetic distinctions needed for accurate discrimination of the letters. For example, the information needed to discriminate B from D or P is provided in the duration and spectrum of the stop release, and the formant transitions following the vowel onset. By locating the stop onset and vowel onset, it is possible to measure these features directly. Similar arguments can be found for the other letters in the alphabet.

For high quality speech, this approach achieved 4% error rate on spoken English letters, compared to 1% error rate for human listeners on the same stimuli. For telephone speech, the comparable numbers are 7% for humans and 11% for our system.

The goal of the present research is to test the generality of the segment-then-classify approach by extending it to the French alphabet. Table 1.. compares the acoustic phonetic structure of the English and French alphabets. It can be seen that although most of the letters are structurally similar (e.g. the letter "A") a few letters have a very different structure (e.g. "Y"). Moreover, some letters contain sounds that are unique to one alphabet, such as the velar fricative /KH/ in the French letter "Y". Because of these differences between the alphabets, it was necessary to modify the categories used for first and second pass classification, and to create new pronunciation models for French letters. Because of the similarities between the alphabets, we were able to use the English alphabet recognizer to help automate the transcription of French letters.

2. SYSTEM OVERVIEW

The French alphabet recognizer is modeled after an English alphabet recognizer described at Eurospeech 1991 [2]. The system accepts telephone speech and performs speaker-

Letter	English	French
A	ey	aa
B	b – iy	b – E
C	s – iy	s – E
D	d – iy	d – E
E	iy	OE , E
F	eh – f	eh – f
G	jh – iy	jh – E
H	ey – cl – ch	aa – sh
I	ay	iy
J	jh – ey	jh – iy
K	k – ey	k – aa
L	eh – l	eh – l
M	eh – m	eh – m
N	eh – n	eh – n
O	ow	OH
P	p – iy	p – eh
Q	k – y – uw	k – ux
R	aa – r	eh – rx
S	eh – s	eh – s
T	t – iy	t – eh
U	y – uw	ux
V	v – iy	v – eh
W	d – ah – cl – b – ...	d – uh – cl –b – ...
	... l – y – uw	... l – OE – v – E
X	eh – cl – k – s	iy – cl – XH
Y	w – ay	iy - cl - KR - eh - cl- KH
Z	z – iy	z – eh – cl – KH

Table 1. Comparison of Pronunciations for the English and French Alphabets using the TIMIT labels. Letter initial vowels might optionally be preceded by a glottal stop. The sounds in the French alphabet for which there is no ARPABET label are: E – long eh; OE – rounded mid vowel; OH – rounded back vowel (no off glide); KR – velar fricative; KH – aspirated k.

independent French alphabet recognition and directory name retrieval. The system modules are shown in Figure 1.

Data Capture. Telephone speech is sampled at 8 kHz at 14-bit resolution using the Gradient Technology Desklab attached to a UNIX workstation.

Signal Representation. Signal processing routines perform a seventh order RASTA–PLP (Perceptual Linear Predictive) analysis [3] every 6 msec using a 10 msec window. This analysis yields eight coefficients per frame. In addition the zero-crossing rate and the peak–to–peak amplitude are computed at this stage.

Frame-based Phonetic Classification. Classification is performed by a fully-connected three-layer feed-forward network that computes 23 phonetic category scores at each 6 msec time frame. The 23 categories provide an intermediate level of description, in which some acoustically similar phonetic categories, such as [b]-[d], [f]-[s],[v]-[z], [p]-[t]-[k] and [m]-[n] are combined; these fine phonetic distinctions are very difficult and are not needed to locate and segment the French letters. The input to the neural network classifier consists of 120 features representing RASTA–PLP coefficients in a 432 msec window centered on the frame to be classified.

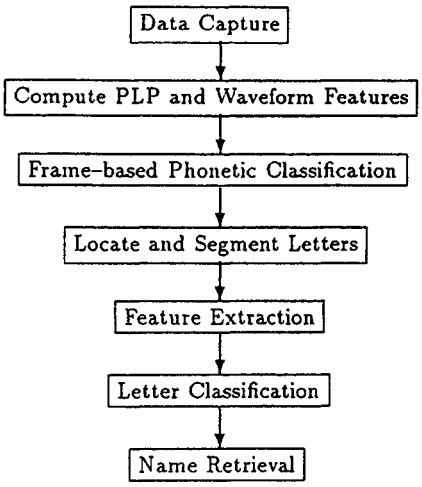

Figure 1. The Module of the Name Retrieval System

Letter Segmentation. The frame-by-frame outputs of the phonetic classifier are converted to a sequence of phonetic segments corresponding to a sequence of hypothesized letters. This is done with a Viterbi search that uses duration and phoneme sequence constraints provided by letter models.

Letter Classification. Once letter segmentation is performed, a set of 178 features is computed for each letter and used by a fully-connected feed-forward network with one hidden layer of 60 units to reclassify the letter. Feature measurements are based on the phonetic boundaries provided by the segmentation. The outputs of the classifier are the 26 letters plus the category "not a letter", which was used to classify coughs, filled pauses and other sounds.

Name Retrieval. The output of the classifier is a score between 0.0 and 1.0 for each letter. These scores are treated as probabilities and the most likely name is retrieved from the database of 50,000 last names. The data base is stored in an efficient tree structure. Letter deletions and insertions are allowed with a penalty.

3. SYSTEM DEVELOPMENT

3.1. French Letter Telephone Speech Database

The system was trained and evaluated on speech from about 600 callers from the Brussels area. Each caller recited the French alphabet with pauses between letters and provided several utterances including "Oui/Non", spelled first and last names with pauses between the letters and last names without a required pause.

3.2. Training and Test Sets

The first pass segmenter was trained on 160 phonetically labeled alphabets (the letters "A" through "Z" in correct order). The speech frames used to extract the features and train the neural network were carefully selected as suggested in [4]. A cross-validation test set was used to determine the optimal amount of training for the neural network. It was made up of the remaining 40 phonetically labeled alphabets.

The second pass letter classifier was trained on 80 alphabets used to train the segmenter as well as 175 spelled first and last names for a total of 3421 letter samples. The final test set consisted of 83 calls containing spelled first or last names with pauses between the letters.

3.3. Semi-automatic Phonetic Labeling

Training the neural network used for the first pass segmentation requires the availability of phonetically labeled speech data. We used our existing English alphabet recognizer to generate an initial segmentation of speech on the training set of complete French alphabets. Human labelers then adjusted the proposed boundaries in cases where the initial segmentation was erroneous, a process that is less time consuming than labeling the entire alphabet by hand.

In order to be able to use the English recognizer, we had to build pronunciation models for the French letters out of the original 22 English phoneme categories used to represent the English alphabet. For example the French letter "A" was expressed as /q/ - /aa/ (as compared to the English pronunciation /q/ - /ey/). In order to determine the best construction of French letter models using the English letter sound categories, we ran the English segmenter on several French alphabets, using only duration constraints to guide the Viterbi search. This process is similar to the pronunciation model generation algorithm described in [5].

Once the first 20 French alphabets were automatically segmented and hand corrected, the initial English phoneme labels were remapped to the French phonetic labels. Then an initial French segmenter was trained and used to segment the next 20 alphabets, which in turn were again hand corrected. This process was repeated until 200 complete French alphabets were phonetically labeled.

3.4. Training the Letter Classifier

In order to avoid hand-segmenting training data for letter classification, an automatic procedure was used. Each utterance was listened to and transcribed manually at the "word" level. Segmentation was performed as described above, except the Viterbi search was constrained to match the transcribed letter sequence.

Then, a second Viterbi search was run without the forced alignment (i.e. any letter sequence was allowed). The two segmentation were then compared. Each occurrence of a "letter" in the unforced segmentation either lined up with a segmented letter in the forced segmentation and hence received that letter category label, or was assumed to correspond to a noise event such as breath noise or background noise and was assigned the category label "Not-A-Letter".

4. SYSTEM EVALUATION

The baseline system described above has been evaluated on 84 calls that were not used for training either network. The segmenter neural network achieved a correct frame classification rate of 67% on the cross-validation test set.

The complete system achieved 84.4% first choice letter recognition accuracy on the test set. The system has also been evaluated on 84 spelled names from different callers. The system achieved 92.8% correct name recognition of the 83 spelled names contained in a database of 50,000 names.

5. REAL-TIME IMPLEMENTATION

The final system has been redesigned and optimized to run in real-time on a PC-board based on a single DSP TMS320C30 and on a PC-board based on a DSP32C. The two passes described above are performed on the DSP while the name search (up to 50,000 names) is performed (as letters are recognized) by the PC. The modifications that were required to the baseline algorithm described above to achieve real-time operation were mostly related to the first-pass of the algorithm:

Zero-crossings. The calculation of the hysteresis band is now done in real-time based on peak-to-peak measurements. The baseline system determined the hysteresis band for the zero-crossing calculation by applying a nonlinear sorting filter to the speech samples during 30 frames of silence in front of the utterance. The fourth best peak-peak value is taken as the hysteresis band value. In the real-time system this hysteresis band is calculated by filtering the peak-to-peak values during silence in a continuous manner.

Normalization of the peak-to-peak features and the first components of the RASTA-PLP vectors. This is now done in real-time by an automatic gain control (AGC) process using a 150ms window. The baseline system used a two-pass procedure to normalize the first RASTA-PLP component and the peak-to-peak features. The first phase required searching the maximum dynamic range of the feature in the utterance, and the second phase normalized the feature by making use of the maximum dynamic range. To overcome this non-real-time operation, the normalization of the features is performed using a variable gain amplifier in which the gain is controlled by the inverse of a peak detector on the feature. The peak detector in the AGC has a decay factor of 0.999 and a limiter is included to prevent excessive gain during silence (squelch).

Removal of the DC-component of the signal. This is now done on a frame to frame basis. The baseline system used a two pass operation to remove the DC component from the signal. The first pass calculated the mean over the complete utterance, the second phase subtracted this mean from the whole utterance. In the real-time system the mean is removed on a frame by frame basis; for the RASTA-PLP calculation this is achieved by setting the first FFT component to zero, for the zero crossing computation the mean calculated over the frame is subtracted, and for the peak-to-peak feature no DC removal is necessary.

RASTA-PLP vector extraction. The RASTA-PLP processing has been modified to use a dual-FFT procedure to reduce calculation time (2 vectors are calculated at once). In the normal RASTA-PLP algorithm an FFT is used on a real input sample buffer. The imaginary input is then set to zero. This involves a number of operations proportional to $N \log N$. The dual-FFT algorithm is capable of calculating the DFT on 2 real input buffers at the same time at the cost of one complex FFT of half the size. By using such a dual-FFT algorithm the number of operations are reduced to be proportional to $N/2 \log(N/2)$.

Pruning strategy. This has been removed from the dynamic programming algorithm used in the first pass. Indeed,

the first pass uses only a limited number of states (approx. 80). For such a small number of states pruning is not actually effective since the overhead is larger than the gain that can be expected from a lower number of active states.

Dynamic programming backtrace node cleanup procedure. The dynamic programming implemented in the baseline system had a backtrace strategy based on the dynamic allocation of backtrack nodes. Nodes are allocated for each state transition encountered in the dynamic programming beam-search. This leads to excessive memory requirements for backtracking (3000 nodes per second). In the real-time version a backtrace node cleanup procedure is added to recover backtrace nodes of the paths that become obsolete during dynamic programming algorithm.

Premature backtracking procedure for dynamic programming. The baseline system consists of two passes that are executed one after the other. This makes the baseline system a non-real-time algorithm because pass 1 must be finished before pass 2 can start. This leads to excessive memory requirements for intermediate buffers for the RASTA-PLP coefficients, the zero-crossing rated and the peak-to-peaks features, because the whole utterance has to be stored for the second pass at a 6kByte per second rate. However, due to the free format syntax used in the dynamic programming algorithm in the first phase, one can design a premature backtrace strategy based on the fact that the backtrace path for all states during dynamic programming are likely to overlap from a certain time. When this occurs (typically after a letter has been spelled), a backtrace procedure can be started and the segmentation which results from this can be used to start the second phase. This means that the intermediate feature buffers can be implemented as ring-buffers containing only the features that are not yet used by the second phase algorithm instead of storing the whole utterance.

Letter probabilities. The baseline system used a name search procedure based on a floating point dynamic programming procedure. Doing the name search on a PC requires a fixed point dynamic programming procedure since floating point operations are very time consuming (even with a mathematical co-processor). Therefore the log probabilities for the letters coming from the second pass neural network are quantized on 8 bit precision.

The system is now being integrated into a commercial board able to handle eight telephone lines in parallel to be evaluated in a real world application.

6. CONCLUSION

We have verified the generality of the segment-then-classify approach by building a letter recognizer for the French alphabet. We used an existing English letter recognizer to semi-automatically label the speech data used to train the French system.

We plan to extend this research by building letter recognizers for other languages that are acoustically similar to English and French such as German and Spanish.

7. ACKNOWLEDGEMENTS

Research supported by Lernout & Hauspie Speech Products, US WEST Advanced Technologies, and grants from the National Science Foundation and the Office of Naval Research.

REFERENCES

[1] Mark Fanty & Ron Cole, "Spoken letter recognition", *Advances in Neural Information Processing Systems 3*, San Mateo, CA: Morgan Kaufmann Publ., 1990.

[2] Ron Cole, Krist Roginsky, & Mark Fanty, "English Alphabet Recognition with Telephone Speech," *Proceedings of EUROSPEECH'91*, pp. 479-482, Genova, Italy.

[3] H. Hermansky, N. Morgan, A. Bayya, & P. Kohn, "Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP)" *Proceedings of EUROSPEECH'91*, pp. 1367-4670, Genova, Italy.

[4] Mark Fanty, Ron Cole & Krist Roginsky, "English Alphabet Recognition with Telephone Speech," *Advances in Neural Information Processing Systems 4*, pp. San Mateo, CA: Morgan Kaufmann Publ., 1991.

[5] Philipp Schmid, Mark Fanty, & Ron Cole, "Automatically generated word pronunciations from phoneme classifier output", *Proceedings of ICASSP 93*, pp. 223-226, Minneapolis, Minnesota.