An N-best strategy, dynamic grammars and selectively trained neural networks for real-time recognition of...

Conference Paper *in* Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on · June 1995

DOI: 10.1109/ICASSP.1995.479828 · Source: IEEE Xplore

Stéphane Valente
STMicroelectronics

Jean Mari
National Institute for Research in Computer Sci...

An N-Best Strategy, Dynamic Grammars and Selectively Trained Neural Networks for Real-Time Recognition of Continuously Spelled Names over the Telephone

Jean-Claude Junqua[1], Stephane Valente[1, 3], Dominique Fohr[2], and Jean-François Mari[2]

[1]Speech Technology Laboratory, Panasonic Technologies Inc., 3888 State Street, Santa Barbara, California, 93105, U.S.A. E-mail: jcj@STL.Research.Panasonic.Com.

[2]CRIN-CNRS & INRIA Lorraine, BP 239-F54506 Vandoeuvre lès Nancy, France.

[3]Institut Eurécom, BP 193-06904 Sophia Antipolis cédex, France.

ABSTRACT

In this paper, we introduce **SmarTspelL**, a new speaker-independent algorithm to recognize continuously spelled names over the telephone. Our method is based on an N-best multi-pass recognition strategy applying costly constraints when the number of possible candidates is low. This strategy outperforms an HMM recognizer using a grammar containing all the possible names. It is also more suitable to real-time. For a 3,388 name dictionary, a 95.3% name recognition rate is obtained. A real-time prototype has been implemented on a workstation. We also present comparisons of different feature sets for speech representation, and two speech recognition approaches based on first- and second-order HMMs.

I. INTRODUCTION

Automatic speech recognition of spelled names is a difficult task because of the confusable letters contained in the alphabet, the distortions introduced by the telephone channel and the variability due to an undefined telephone handset. However, in an application, the names generally belong to a fixed list and the knowledge of this list can be used to apply constraints on the sequence of letters. One way to use this knowledge is to define a grammar containing the name list, and to constrain the recognition with this grammar. While reasonable recognition accuracy can be obtained with this method, response time increases very rapidly with the size of the dictionary. As our concern was to develop a real-time recognizer, we investigated a different strategy. In this paper, we present our recognition procedure, its evaluation on the OGI speech telephone corpus, the performance of various analysis techniques for telephone speech and a comparison between two recognition approaches based on a first- and second-order HMMs (called hereafter HMM1 and HMM2).

II. THE RECOGNITION STRATEGY

The central idea of our method is to propagate N-best hypotheses through different processing modules and to apply costly constraints, if needed, at the end of the processing when the number of remaining candidates is low. As shown in Figure 1, our recognition strategy consists of at most four passes. The first pass, which produces the N-best sequences of letters (N=20 in our experiments) given acoustic hidden Markov models of the letters, is the most time consuming. After the first pass, selectively trained neural networks (STNN) [1] focus on the discriminative segments of speech, where the distinct acoustic information is localized. The discriminative speech segments are determined using the segmentation given by the first pass and an energy criterion. This second pass is activated each time an hypothesized letter belongs to one of the confusable subsets. The third pass consists of a DTW alignment procedure taking into account confusions made by the previous two passes as well as insertion, deletion and substitution penalties. This third pass provides the decision strategy module with N-best candidates. Based on the scores of the third pass candidates, the decision strategy module decides if the first candidate should be considered as the recognized name or if the fourth pass should be invoked. If the fourth pass is invoked, a dynamic grammar is built with the N-best candidates provided by the DTW alignment and the HMM recognizer used in the first pass is re-run with this highly constrained grammar (typically 10 or 20 names). When the fourth pass is invoked, its output is the recognized name.

III. DATABASE

The database used in our experiments is a subset of the speech telephone corpus collected at Oregon Graduate Institute (OGI) [3]. Over four thousand people called in response to public requests. They were prompted by a recorded voice to say their first and last names, with and without pauses, together with other information. 225 repetitions of the alphabet and more than 1300 different calls were selected for the training, 558 calls for the validation and 491 calls for the test. The purpose of the validation experiments was to optimally tune the system's parameters before running it on the test set. As every speaker belongs only to one set (training, validation or test) the experiments conducted are speaker-

0-7803-2431-5/95 $4.00 © 1995 IEEE

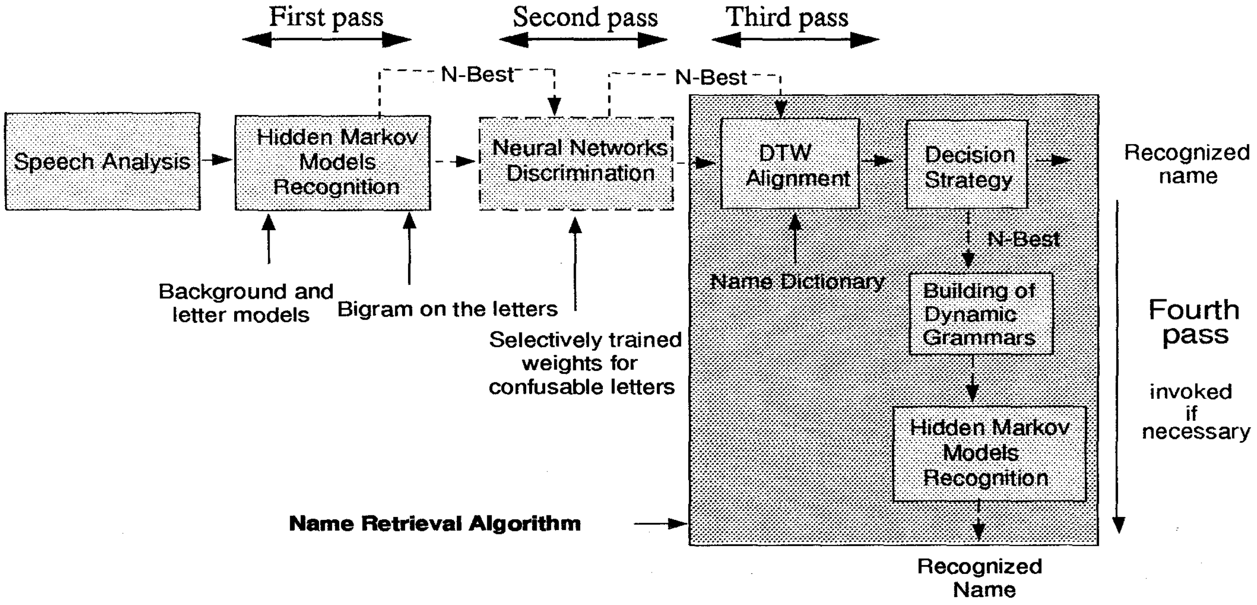

Figure 1. Block Diagram of the SmarTspelL recognition system.

independent.

IV. THE FIRST PASS

IV.1 The HMM1 recognizer

Our first pass is based on a *frame synchronous first-order continuous density hidden Markov model* recognizer with beam search. The development of this recognizer started from a modified version of the HTK toolkit Viterbi decoder [10]. A bigram letter grammar, computed on the training set labels, is used in the decoder. The output of the decoder consists of N-best hypotheses computed with a word-dependent algorithm derived from [9]. To be more efficient in the word-dependent algorithm and limit the memory space allocated, we included an adaptive path pruning threshold which decreases the number of paths processed, and a local word pruning which eliminates theories whose last word probability does not score well as compared to the best last word probability. In the case of confusable words, we use state tying to help the recognizer focus on the discriminative part of the word and to decrease the number of estimated parameters. The tied letters are (m, n), (i, r), (p, t) and (b, d). We chose 6 state HMM models for all letters but "w" (12 states) and the silence model (1 state). Letter models have different numbers of Gaussian mixtures, depending on how confusable the letters are. The letters are modeled with 3 mixture densities, except b, c, d, e, g, p, t, v and z (the "e-set") and m, n, s and f, which are modeled with 6 mixture densities.

IV.2 Front-end optimization

At the speech analysis level, we compared the 8th-order PLP-RASTA [4] cepstral coefficients with a 14th-order MFCC analysis. For PLP-RASTA, we used a 10ms frame shift and a 20ms analysis window. As shown in figure 2, we optimized the RASTA filter coefficient to decrease the number of substitution, deletion and insertion errors. The best compromise was found for a value of 0.90. In these experiments, the energy, the first derivative of the energy, and the first derivative of the static cepstral coefficients C_1 through C_8 (computed over 7 frames) were combined with the static cepstral coefficients to form the speech parametric representation (a total of 18 coefficients). For the MFCC analysis, we used 11 static cepstral coefficients (C_0 included) computed with a frame shift of 16ms and an analysis window of 32ms. Various feature sets, combining static and dynamic features, have been compared with PLP-RASTA (see Figure 3). To obtain filtered data for the test set, we applied a distorting filter as shown in [7]. Filtering the test data artificially created a mismatch between training and testing sets. In Figure 3, S, stands for static coefficients and R1, R2 for, respectively, first-order and second-order regression coefficients.

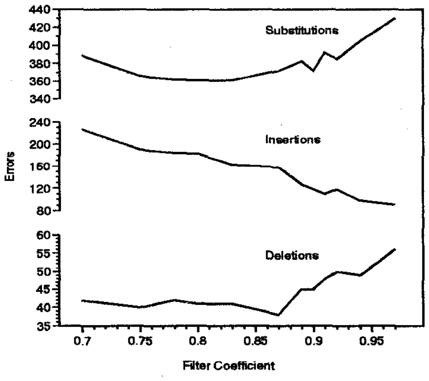

Figure 2. Optimization of PLP-RASTA.

These results show that:

- including a second derivative slightly improves recognition accuracy for unfiltered speech;
- both PLP-RASTA and the combination of MFCC first and second derivatives (R1+R2) successfully handle the mismatch between training and testing; however, R1+R2 alone decreases the recognition accuracy for the unfiltered data;
- static coefficients by themselves are not robust against a mismatch between training and testing conditions;
- long regression windows for the first and second derivatives decrease recognition accuracy (for our database the average letter duration is 386ms). Additional experiments for other window sizes confirmed this observation (e.g. R1(112ms) and R2(208ms)). This result is in agreement with Nadeu and Juang [8], who mentioned that long regression windows may not be desirable for continuous speech recognition systems.

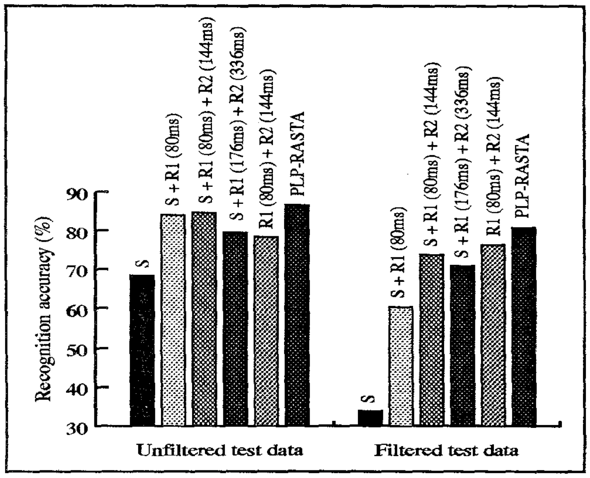

Figure 3. Recognition accuracy obtained with different feature sets. Recognition accuracy was computed by taking the percentage of the number of correctly recognized letters minus the number of insertions over the total number of letters.

PLP-RASTA gave the best recognition accuracy. However, good performance was also obtained with the MFCC analysis. The band-pass filtering included in PLP-RASTA explicitely compensates for channel distortion. However, we believe that some kind of multi-style training (more exactly multi-environment training) is happening because of the diversity and the size of our database. In the case of the MFCC analysis, multi-style training also compensates for the channel distortions. In this study, one of our concerns was to develop a system suitable for real-time implementation. Consequently, we did not investigate the combination of long-term cepstral subtraction with MFCC analysis which could have improved the accuracy. A short-term cepstral substraction may constitute an alternative. The low dimensionality of PLP-based feature vectors led us to choose the PLP-RASTA analysis for our multi-pass recognizer in the remaining part of the study.

IV.3 Comparison between first and second-order HMM

We compared our system to a second-order hidden Markov model recognizer [6, 5] where the underlying state sequence is a second-order Markov chain in which the transition probability between two states at time t depends on the states in which the process was at time t-1 and t-2. This HMM2 system has been shown to give good results on the same task [5]. Compared to the system presented in [5], we tried to optimize the whole system by:

- varying the feature sets on the basis of the MFCC and PLP-RASTA parametrization;
- varying the number of mixtures

The best results are obtained for 6 mixture densities, when the feature vector is represented by 11 Mel static cepstrum coefficients (without the energy), plus the first derivative of the energy, the first derivative of the static coefficients and the second derivative of the energy (a total of 24 coefficients). Comparative results between HMM1 and HMM2 with the different optimization schemes is presented in Table 1. Compared to the results reported in [5], while the insertion rate decreased, the recognition rate also decreased and the deletion rate increased. In these HMM2 experiments, PLP-RASTA and MFCC-based analyses gave similar performance. From Table 1 (obtained on 3145 letters from the 491 names of the test set), we can see that very comparable results are obtained with HMM1- and HMM2-based systems. (the differences are not significant). However, a number of differences exist between the two systems. HMM2 models duration information, while the system based on HMM1 uses tied state modeling and a bigram grammar. The results given in the next sections were obtained with the HMM1 recognizer, which is more suitable for a real-time implementation.

	HMM1	HMM2
Correct	86.8%	86.1%
Substituted	11.8%	11.4%
Deleted	1.4%	2.4%
Inserted	3.7%	1.1%

Table1 : Performance comparison between HMM1 and HMM2.

V. DISCRIMINATION AND ALIGNMENT

As expected, most of the confusions occurring in the first pass are obtained for confusable subsets (e.g. {M,N}, {P, T}, {B, D, V}). By applying, after the first pass, discriminative neural networks to the confusable letters of the subsets {B, D, V} and {P, T}, we could increase the recognition rate on these letters by more than 3%. The discriminative method is based on a search for the frames which bear the most distinction between the confusable words. Then, a parametrization is done on these frames and the resulting vectors are given to a neural network which provides the final decision. At this level too, different feature sets have been investigated but an MFCC-based feature set gave the best results.

The third pass of our method consists of a DTW alignment with a name dictionary. Three dictionary sizes have been tested: 491, 3,388 and 21,877 names. The alignment provides N-best candidates (20 in our implementation) to a decision strategy module which uses their scores to determine if the fourth pass should be invoked or not.

VI. DYNAMIC GRAMMARS AND NAME RETRIEVAL

Finally, in the fourth pass, dynamic grammars are built with the N-best candidates provided by the alignment module and the hidden Markov model recognizer is invoked with this constrained grammar. As the number of names in the grammar is limited (< =N-best candidates) this fourth pass is not time consuming. The results obtained after the first pass, the alignment (third pass) and the fourth pass, for the different dictionaries, are given in Table 2.

The average confusability, indicated between the parenthesis in the first column of Table 2, is a measure of confusability of the dictionary [2]. In the third dictionary (21,877) there are 39,302 pairs of names which differ by one letter substitution. This corresponds to an average of 1.8 confusions per name.

In these scores, the improvement yielded by the neural network discrimination has not yet been taken into account. These results compare favorably with previous reported work [3]. However, Cole et al. used a larger dictionary and considered names spelled with pauses between letters. A real-time version of our recognizer, working over the telephone network, has been implemented on a workstation.

Size of the dictionary and average confusability	Third pass name recognition rates	Fourth pass name recognition rates
491 (0.07)	97%	98.4%
3,388 (0.5)	90.6%	95.3%
21,877 (1.8)	87%	90.4%

Table 2 : Name retrieval accuracy after the third pass (alignment) and the fourth pass (use of dynamic grammars).

VII. DISCUSSION AND PERSPECTIVES

We compared the performance of our approach with that of a more conventional spelled name recognition task, where all the names are present in a recognizer grammar. In this case, we used a Viterbi decoder providing only the best candidate. This conventional approach yielded a 93.1% recognition rate on the 491 name dictionary. This recognition rate is much lower than that obtained with our new approach and, moreover, the decoding process is several times slower.

For the real-time multi-pass version of our system, we used various pruning mechanisms which, on the average, led to a decrease in name recognition rate between 3% and 5%.

The discrimination provided by neural networks (second pass) is most useful when the size of the name dictionary increases. We observed that in the case of the 21, 877 dictionary, more than 95% of the time the correct name can be found in the first two candidates. Furthermore, when there is a misrecognition, often the first and second candidate differ by only one letter. Consequently, it may be useful to re-apply our STNN discrimination method as a fifth processing pass.

Several improvements of SmarTspelL are investigated. Looking at the errors found after the first pass, a number of recognition errors may be avoided using duration information. Consequently, we are currently implementing a duration prediction mechanism to prune hypotheses which are outside a predicted duration length. Such a mechanism will eventually decrease the amount of space, increase the speed and possibly the performance of the system. Another potential source of improvement is the use of trigrams instead of bigrams as a language model.

REFERENCES

[1] Y. Anglade, D. Fohr, and J-C. Junqua. Speech discrimination in adverse conditions using acoustic knowledge and selectively trained neural networks. In *ICASSP-93*, pages 279–282, 1993.

[2] R. Cole, M. Fanty, M. Gopalakrishnan, and R.D.T. Janssen. Speaker-independent name retrieval from spellings using a database of 50,000 names. In *ICASSP-91*, pages 325–328, 1991.

[3] R. Cole, K. Roginski, and M. Fanty. English alphabet recognition with telephone speech. In *EUROSPEECH-91*, pages 479–482, 1991.

[4] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP). In *EUROSPEECH-91*, pages 1367-1370, 1991.

[5] J.-F. Mari, D. Fohr, Y. Anglade, and J.-C. Junqua. Hidden Markov models and selectively trained neural networks for connected confusable word recognition. In *ICSLP-94*, pages 1519–1522, 1994.

[6] J.-F. Mari and J.-P. Haton. Automatic word recognition based on second-order hidden Markov models. In *ICSLP-94*, pages 247–250, 1994.

[7] H. Murveit, J. Butzberger, and M. Weintraub. Reduced channel dependence for speech recognition. In *DARPA Workshop Speech and Natural Language*, pages 280–284, February 1992.

[8] C. Nadeu and B-H. Juang. Filtering of spectral parameters for speech recognition. In *ICSLP-94*, pages 1927–1930, 1994.

[9] R. Schwartz and S. Austin. Efficient, high performance algorithms for N-best search. In *DARPA Workshop on Speech Recognition*, pages 6–11, 1990.

[10] S. Young. HTK: Hidden Markov model toolkit V1.4. Technical report, Cambridge University, Engineering Department, Speech Group, 1992.