Connected digit recognition in spontaneous speech

Étienne Bauche†, Bojana Gajic†**, Yasuhiro Minami†, Tatsuo Matsuoka†, and Sadaoki Furui††*

†NTT Human Interface Laboratories
††Tokyo Institute of Technology
E-main: minami@nttspch.hil.ntt.co.jp

ABSTRACT

In this paper, we describe a new recognition system for 4-digit-strings in Japanese under fluent speech conditions. In particular, we introduce several methods to solve the problems related to the spontaneity of speech: discrimination of speech and background noise, out-of-vocabulary words, pauses between digits, etc. These methods led to an error rate reduction of 76%, compared to a simple start- and end-point detection based recognizer using non-refined models.

1. INTRODUCTION

A robust automatic recognition system for connected digit strings under spontaneous speech conditions is crucial for a number of applications; such as voice dialing of telephone numbers and automatic credit card or personal identification number entry [1-5|. One may think the recognition of digit strings is a relatively easy task, considering the limited size of the dictionary (10 words). However, the output of the recognizer must be the exact transcription of the input speech, not only its "overall meaning". In other words, the *word accuracy* needs to be high for the application to be of any use. In order to identify the major sources of error in a spontaneous speech recognition system, we started with the crudest version of a recognizer and performed experiments on a set of about 1000 4-digit-strings. We found some errors specific to spontaneous speech recognition tasks (incorrect speech detection, excessive pauses between digits, interference from out-of-vocabulary words) and other errors common to all recognition systems (digit substitutions, digit insertions and deletions, background noise). In what follows we introduce several techniques to deal with the above sources of error.

2. BASELINE SYSTEM

We performed simple speech recognition experiments for 4-digit strings to analyze the major errors in spontaneous speech .

2.1 Recognition system

•Start-point and end-point detection

The input of a realistic recognition system, being a continuous sequence of speech and background events, requires an efficient algorithm to distinguish the speech utterances from the surrounding background. In this study, we implemented a start-point and end-point detector based on HMM likelihoods. For every input speech frame we compared the probability between two HMMs: a background model and a composite *all-digit* model, whose purpose is to match any digit event. This model was trained using a large database containing samples of all digits. When the difference between the probabilities rises above an experimentally determined threshold, the start-point is set and recognition is started. The comparison is then carried on until the end-point is decided, using the same likelihood-based criterion. The thresholds were first determined so that the largest pause between two digits in a string did not exceed 320ms.

•HMMs

A feature vector consists of the 16-dimension mean normalized cepstrum vector and 16-dimension delta

---

*currently, The École Nationale Superieure des Télécommunications (E.N.S.T.)

** currently, Norwegian Institute of Technology.

cepstrum vector. A context-independent HMM was trained for each digit, with a number of states (including entry and exit states) varying from 8 to 14 according to the digit length. The number of mixtures was 4. The silence model is a single-state 4-mixture HMM. The *all-digit* model has also one state represented by 4 mixtures.

• Network

The network specifications are as follows: *sil digit digit digit digit sil,* where sil stands for silence or background.

2.2 Database

We used 1013 strings uttered by 27 male speakers, recorded in at NTT's speech laboratory in a slightly noisy room for testing. The speakers were asked to respond to questions while the microphone was kept open. In these recordings, speakers were free to utter anything they liked besides the 4-digit-string. This resulted in a database containing a number of phenomena. HMMs for the 10 digits were trained using a database consisting of 9704 noiseless 4-digit-strings clearly uttered by 70 male speakers. The same database served to build the background and *all-digit* models.

2.3 Result and analysis

Table 1 shows the base line result of our system. The 247 errors can be divided into different groups: (This classification will be used throughout this study.)

Table 1. Baseline recognition result.

Method	Error rate	Number of errors
base line	24.4%	247

•Overly long pauses between digits: 121 errors. This frequent problem is due to a silent pause between two digits in a string that is longer than the pause threshold. In this case the end-point detector cuts the string into two substrings, none of which contain the four correct digits, thus causing an error and a string insertion.

•Out-of-vocabulary-words interference: 69 errors. All these errors are due to extraneous speech adjacent to the true digits and misinterpreted as such. The consequence is either a digit substitution or insertion. More precisely, some extraneous words were found to cause more errors than others, especially sounds like "eee" or its more nasal version "mmm" (31 errors; very often misrecognized as 4 -"yon" in Japanese-) and "eeto'' (13 errors; sometimes misrecognized as a 5 -"go" in Japanese- when the "o" has a long duration). "Eeto" has literally no meaning; it is may be close to the English "well".

•Digit substitutions: 45 errors. This type of error corresponds to output strings free of extraneous words in which one digit was misrecognized for another digit. For example, "ichi" (one) is sometimes misrecognized as "hachi" (eight).

•Digit insertions or deletions: 12 errors. In this task, since the number of digits to be recognized is set by the network syntax to be four, each time there is a digit insertion, there is also a digit deletion, so the two concepts are equivalent.

3. IMPROVING SYSTEM PERFORMANCE

To help solve the problems described in the preceding section, we introduce the following methods.

3.1 Garbage model

In a realistic application, again, it is very likely that users will add extraneous speech while answering the system. Thus, not only do whole segments of extraneous speech in the output of the start- and end-point detector have to be discarded, but also out-of-vocabulary words (i. e. non-digit) in the same segment that contains the digit string to be recovered should be spotted and rejected. For that purpose, we first trained an *all-speech* model, applying the same technique that served to build the *all-digit* model. Using not only the digit occurrences but also every phoneme in the database, we obtained a model that was a fair match for all Japanese words. This model was inserted in the recognizer syntax either as a separate alternative to the digit string or inside the string hypothesis, at both ends of the string, where most of the errors occur.

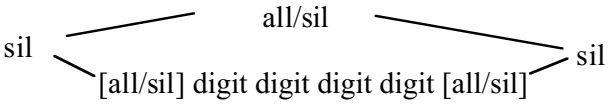

Figure 1. Network for 4 digits with garbage model.

3.2 Pause Algorithm

Another problem that can occur during recognition due to the spontaneity of speech is related to possible long pauses between the digits in a string. Indeed, any pause longer than 320ms would cause the string to be split into two sub-strings contained in two different speech segments, thus leading to an error (and a string insertion). We propose a *pause algorithm*, whose purpose is to complete a string whose digits are scattered in different speech segments. This algorithm required the implementation of a verification and rejection procedure be implemented at both the digit level and the string level. The recognition result for each digit is classified into three levels ("good", "satisfactory" and "bad"). The three following criteria are used in this verification procedure: digit duration, likelihood comparison between the HMM of recognized digits and the garbage model (all-speech), and similar likelihood involving the digit and background HMMs.

Thresholds were determined experimentally using the same database used to build the digit HMMs.

The pause algorithm is executed as indicated in Fig. 2 using these verification classes. If recognized string S contains four satisfactory digits, it is accepted. If not, but provided there is at least one good digit in S, the algorithm

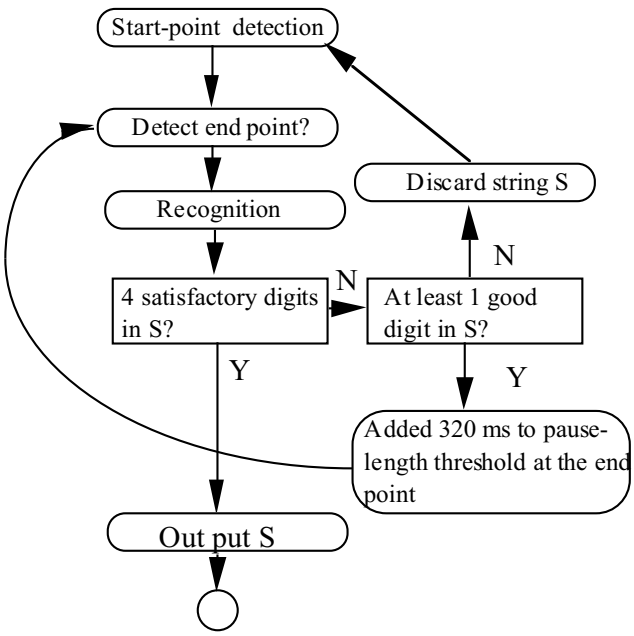

Figure 2. Pause algorithm.

supposes that there is a long pause between the digits. The pause-length threshold is then increased and a new end-point is determined. Eventually, either a string containing four satisfactory digits is found during the search, or, when no such string is found and the pause-length threshold reaches a maximum value (2560ms), the string is finally discarded unless there are at least three satisfactory digits.

3.3 Filler Models

In order to assist the garbage model in the rejection of out-of-vocabulary words, we added so-called *filler models* to match specific words or parts of easily confusable non-digit words. "Eeto", "djaa" or "ano", for example, are frequently used interjections in Japanese which, or part of which, were often misrecognized as a digit. For the training of these *filler models* we used occurrences of the confusable interjections.

3.4 Accurate Models

Another major source of errors is related to digit substitutions, i. e., digits misrecognized as other digits. In particular, we spotted 15 errors (out of 1000 sentences) in the original recognition experiment caused just by confusion between "ichi" (one) and "hachi" (eight). To improve the discriminating capability of HMM models, one can increase the number of parameters used in them. In this study, we increased the number of Gaussian mixtures in the digit HMMs from four to eight.

3.5 Noise Adaptation

The last technique we tried in order to improve the system performance aimed at the reduction of background noise mismatch between the training and testing databases. To adapt the digit HMMs to noisy conditions, we implemented Maximum A Posteriori (MAP) adaptation, a technique usually used in speaker adaptation [6].

4 EXPERIMENTS

4.1 Experimental conditions

The experimental conditions, except for a few extra conditions, were the same as for the baseline system. Here, we describe extra conditions. For the *all-speech* model, we used a large Japanese database containing phonetically balanced newspaper text read by 30 speakers. The *all*

speech model had one state represented by 16 mixtures. The third smaller database (300 6-digit-string uttered by 10 speakers) recorded in the same noise conditions as the training database served two purposes. It was used as a training database for the *filler models* and as a base for the noise adaptation of the digit HMMs.

4.2 Recognition results

In the initial test, in which we used the 4-mixture-digit models and a simplified network syntax, we did not allow the insertion of out-of-vocabulary speech. Table 2 shows the results (baseline). With the garbage model, the error rate reduction was almost 33% (24.4% to 19.2%). We also tried to allow the recognition of out-of-vocabulary words between the digits of a string, but this led to a great drop of performance. Introducing the pause algorithm reduced the error rate from 19.2% to 11.4%. The addition of *filler models* further improved performance. However, increasing the number of mixtures from 4 to 8 did not yield the expected improvement. We noticed that while a lot of errors were indeed recovered, a number of new errors occurred. Following the idea that this was due to a mismatch in the noise conditions between the training and the testing databases, we performed "environmental adaptation". This led to a great reduction of the error rate (over 30%). In the end, the use of the pause algorithm, garbage and *filler* models, and MAP-adapted HMMs resulted in error reduction of about 76%.

Table 2. Recogntion result

Method	Error rate	Number of errors
base line	24.4%	247
(1) + garbage model	19.2%	194
(2) + pause algorithm	11.4%	115
(3) + filler models	9.5% (8.7%*)	96 (68*)
(4) + accurate models mixtures	8.3%*	65*
(5) + noise adaptation	5.5%*	43*

* means that this result was obtained by evaluating only 21 speakers, because the other 6 speakers were also speakers of the smaller database, tha t served for the training of *filler models* and for MAP adaptation.

5. CONCLUSION

We evaluated 4-digit recognition in spontaneous speech and analyzed the errors. Investing these error phenomena, we introduced several methods into our system. Of these methods, the pause algorithm and noise adaptation using MAP improve the recognition performance significantly. Finally, introducing all of the methods, we could improve string error rates from 24.4% to 5.5%.

ACKNOWLEDGMENTS

We thank the members of the Furui Research Laboratory of the NTT Human Interface Laboratories for their useful discussions. We also thank Mr. Takagi of Tokyo Institute of Technology for collecting digit string data.

REFERENCE

[1] L. R. Rabiner et al., "High performance connected digit recognition using hidden Markov models," Proc. ICASSP-89, pp. 1211-1214 , 1989.

[2] L. Netsch et al., "Enhanced voice services in the telecommunication network using the Texas Instruments Multiserve™," Proc. Workshop on IVTTA, pp. 81-84, September 1994.

[3] D. J. Brems et al., "Dialog design for automatic speech recognition of telephone numbers and account numbers, "Proc. Workshop on IVTTA, pp. 117-120, September 1994.

[4] T. Isobe et al. "Telephone speech data corpus and performance of speaker independent recognition system using the corpus," Proc. Workshop on IVTTA, pp. 101-104, September 1994.

[5] T. Isobe et al, "Nationwide collection of telephone speech and evaluation of recognition using the data," 2-Q-26, Proc. ASJ Spring meeting, pp. 135-136, March 1993.

[6] J.-L. Gauvian and C.-H. Lee "Maximum a posteriori estimation for multivariate Gaussian mixture observation s of Markov models", IEEE Trans. Speech Audio Processing, vol. 2, pp. 291-298, 1994.