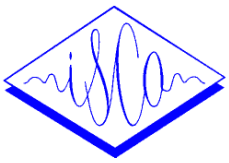


3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

AN APPLICATION OF WORD-SPOTTING IN A VOICE ACTIVATED SERVICE ENTRY SYSTEM

Akihiro IMAMURA *Mikio KITAI*

NTT Human Interface Laboratories
1-2356 Take, Yokosuka, Kanagawa, 238-03 JAPAN

ABSTRACT

This paper describes a speaker-independent word spotting algorithm and its application in a voice activated service name entry system over telephone networks. The system allows a user to specify a keyword with extraneous speech in response to system audio prompts. A task oriented multi-speaker isolated-word database is used to construct keyword HMMs. The score of keywords is computed by using the continuous Viterbi decoding algorithm with score normalization using a background HMM. The candidates for keywords are obtained after evaluating the partial matches. The dialogue design of the system improves user-friendliness and shortens the user's operation time by employing a newly proposed adaptive confirmation procedure. The procedure effectively repeats commands and masks inappropriate control functions. An evaluation over real telephone lines is presented that confirms improved user interface for menu selection tasks.

Keywords : Word spotting, Dialogue system

1. INTRODUCTION

The need for automatic telephone services has been increasing. By using DTMF signals, a handy human machine interface to realize such services can be constructed. However, it is difficult for users to accurately remember the strings of digits required for each command. An accurate speech recognizer would create a much friendlier interface. In practice, users utterances include much extraneous speech, and automatic speech detection methods will not work accurately in noisy environments. The solution for these problems is word spotting.

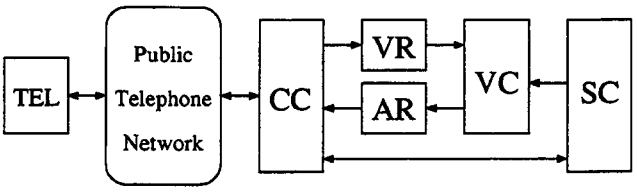
This paper focuses on the design of an HMM-based word spotter and the techniques needed to improve the user interface for the menu selection task. The design of the word spotter is an extension of our previous work [1]. Keyword candidates and their location are computed by using continuous Viterbi decoding with task oriented Fuzzy VQ type whole-word HMMs. The introduction of likelihood normalization using the score from a background HMM and a reduction method for partial matches effectively improve recognition accuracy. To construct a word spotter driven real time dialogue system, the algorithm was implemented on a Transputer [2] based VME bus board with a telephone exchange interface.

With voice activated dialogue systems, it is impossible to eliminate the confirmation phase completely. However, continuous confirmation is very boring for the user who tends to repeat the last command which may confirm incorrect word candidates. Therefore, in our system, three methods were introduced to improve user-friendliness and to shorten the user's operation time: 1) confirmation is omitted for some candidates according to the likelihood of detected keywords; 2) commands can be repeated in the confirmation phase; 3) rejected keywords in confirming phase are masked from recognition objects.

The following sections describe the outline of the dialogue system and word spotter design, discuss the improvements made to the user interface, and introduce experimental results.

2. SYSTEM STRUCTURE

The proposed system was implemented as a user interface for requesting services and services parameters. A block diagram of the system is shown in Fig. 1. The CC (Connection Controller) detects the arrival of telephone calls, handset state and DTMF signals. The CC then sets or releases the connection between the caller and the system. The VR (Voice Recognition unit) extracts keywords embedded in the caller's speech by word spotting. The AR (Audio Response unit) outputs system prompts to the caller as synthesized voice messages. The VC (Voice interaction Controller) enables the system to converse with callers by controlling VR and AR. The SC (Service Controller) makes a dialogue plan to realize each service.



CC : Connection Controller VC : Voice interaction Controller
VR : Voice Recognition unit AR : Audio Response unit
SC : Service Controller TEL : User's Telephone

Figure 1. Block diagram of the system

A typical service-entry voice dialogue is as follows;

User : (Call up system)
System : (off-hook) Operator. Service name, please ?
User : Well, *formatted-reservation*, please.
System : Is that formatted-reservation ?
User : *Yes*, that's correct.
System : What date do you wish to reserve ?
User : Ah... *September 21st.*
System : What time do you wish to start it ?
User : From... Umm... *just 10 o'clock.*

3. WORD-SPOTTER DESIGN

The VR was realized as a word spotting board as shown in Fig. 1. The word spotting algorithm is based on a whole-word HMM technique and is an extension of our previous work [1]. The extended part of the design is described in the following sections.

3.1. FUNDAMENTAL ALGORITHM

The processing flow of the word spotting algorithm is shown in Fig. 2. The input speech signal is converted into a set of log-scaled delta-energy, LPC derived cepstral and delta-cepstral vectors every 12 msec. Each vector for frame t is then coded to fuzzy observation symbols using the fuzzy VQ technique [3]. Fuzzy VQ represents an input vector x_t as a weighted combination of the code-vectors k in the given codebook. Using the weighting coefficients $u_{t,k}$, a log-scaled observation symbol output probability $\omega_{i,j}(x_t)$ for state transition from i to j of each HMM is given by

$$\omega_{i,j}(x_t) = log(\sum_{k=1}^{M} u_{t,k}\ b_{i,j}(k)), \qquad (1)$$

where $b_{i,j}(k)$ is the discrete observation symbol output probability of codeword k associated with state transition from i to j and M is codebook size.

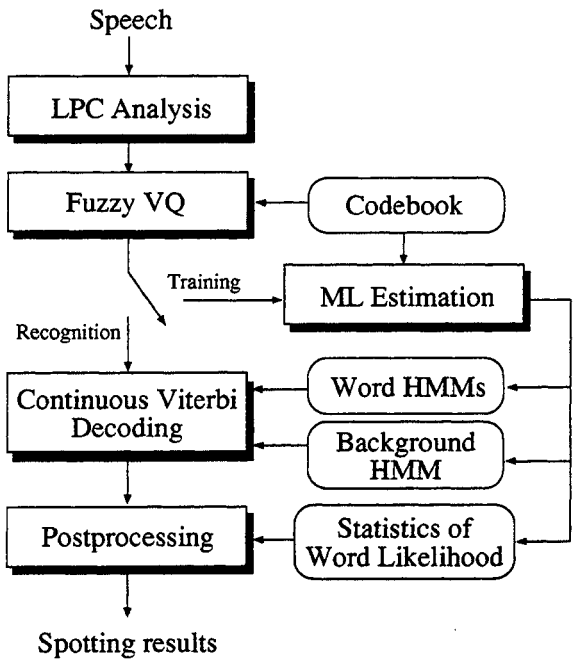


Figure 2. Processing flow of word spotting

Against whole vocabulary word, input speech is scored frame-synchronously by using continuous Viterbi decoding and likelihood normalization using the score from a background HMM. The background HMM, in our system, has two states and its parameters are estimated from non-keyword speech and noise. The details of the algorithm are almost the same as described in [1] except for the optimal path selection process and the final likelihood scoring of candidates. In this paper, the optimal state i_{best} at frame t is given as

$$i_{best} = \underset{1 \le i \le N}{\text{argmax}}\{\frac{\delta_{t-1}(i) + a_{i,j} + \omega_{i,j}(x_t)}{Q_t(B_{t-1}(i))}\}, \qquad (2)$$

where $\delta_{t-1}(i)$ denotes the best non-normalized accumulated likelihood score, $a_{i,j}$ is the state transition probability from state i to j and $Q_t(B_{t-1}(i))$ is a likelihood score obtained from the background HMM for the speech period between the start frame of candidate $B_{t-1}(i)$ and present frame t.

The final likelihood score $P(t)$ of a candidate terminating at frame t is computed as

$$P(t) = \frac{\delta_t(i_{t_{max}})}{Q_t(B_t(i_{t_{max}}))}, \qquad (3)$$

where $i_{t_{max}}$ denotes the index of the optimal final state at which the normalized accumulated score becomes maximum. This score is useful to compare candidates extracted from different speech periods.

Postprocessing effectively rejects candidates that have either insufficient scores or unreasonable state transition histories as described in [1]. This is performed by the threshold logic for score and the duration control for each state and word length using the statistics of training samples.

By introducing these modifications to our previous word spotter, the misrecognition rate for the recognition of a simple service menu command input task (6 choices possible) decreased by 50%.

3.2. REDUCTION OF PARTIAL MATCHES

Word spotters that have no grammar constraints (ex. word-pair, finite state grammar etc.) can output many candidates associated with any partial period of the input speech. In the vocabulary, if there is a word pair constructed from a compound word and a partial word for which pronunciation is included in that of the compound word, the word spotter will produce two candidates for the input speech including the compound word sound. These are; one for the correct compound word, and one for the corresponding partial word.

This kind of recognition (we call this *Partial Matches*) occurs frequently in the spotting of Japanese digits. For example, when the compound word *sanjuppun* (in English *thirty minutes*) is input, the word spotter would output the word candidates for both the correct word and the partial word *juppun* (in English *ten minutes*). Unfortunately, in several cases, the partial word candidate has a better likelihood score than the correct compound word.

To reduce misrecognition caused by partial matches, the following ranking of candidates is performed as a postprocess.

1. Perform next steps if the best candidate W_p is a partial word of any detected compound word candidate $\{W_{c_i}\}$, where i is the ranking of compound words.
2. Find a compound word candidate W_{c_j} which satisfies the following;

$$P(W_{c_j}) \ge P_{ave}(W_{c_j}) - K_a P_{dev}(W_{c_j}), \qquad (4)$$

$$P(W_p) - P(W_{c_j}) \le P_{diff_{ave}}(W_p, W_{c_j}) + K_b P_{diff_{dev}}(W_p, W_{c_j}), \qquad (5)$$

where $P_{ave}(W)$ and $P_{dev}(W)$ denote the average and standard deviation of likelihood score for word W, respectively.

$P_{diff_{ave}}(W_1, W_2)$ and $P_{diff_{dev}}(W_1, W_2)$ denote the average and standard deviation of likelihood score difference between the correct word W_1 and the incorrect word W_2, respectively, and K_a and K_b are constants.

3. Let W_d be a differential word between W_{c_j} and W_p. For example, the digit word *san* becomes a differential word between a compound word *sanjuppun* and a partial word *juppun*. Against the HMM of the word W_d, compute the likelihood score $P(W_d)$ for the speech period where the differential word is expected.
4. if the score $P(W_d)$ satisfies;

$$P(W_d) \geq P_{ave}(W_d) - K_a P_{dev}(W_d), \quad (6)$$

make the ranking of W_{c_j} higher than that of W_p, if not, return to the 2nd step to find next compound word.

The statistical constants used in the above decision are estimated in the training phase of the word spotter.

4. IMPROVEMENTS IN USER INTERFACE

The introduction of the word spotter into voice dialogue system may expand the application area. Any user utterance including any redundant sound can be acceptable by the system. But too much redundant sound in the input will cause misrecognition because the vocabulary and rejection ability of the word spotter is limited. From this point, it is clear that it is impossible to eliminate the confirmation phase completely even in a simple dialogue system. In the next few sections, a confirmation phase is developed that lets the system guide the user adequately without becoming boring.

4.1. ADAPTIVE CONFIRMATION PROCEDURE

Usually, word candidates are confirmed by voice each time the user says a word. Considering the lack of recognition accuracy in public telephone networks, some form of confirmation is inevitable. However, since word by word confirmation rapidly bores the user, we introduced adaptive confirmation procedures based on the candidate correctness obtained from likelihood scores for the 1st rank candidate word W_1 and 2nd one W_2 as follows;

Level I

$$P(W_1) \leq P_{ave}(W_1) - K_a P_{dev}(W_1) \quad (7)$$

Level III

$$P(W_1) \geq P_{ave}(W_1) - K_c P_{dev}(W_1), \quad (8)$$

$$P(W_1) - P(W_2) \geq P_{diff_{ave}}(W_1, W_1) - K_d P_{diff_{dev}}(W_1, W_2), \quad (9)$$

Level II : Case of not above.

If the candidate score is Level I, the confirmation of the 1st rank candidate W_1 is omitted and the subsequent dialogue sequence proceeds. Conversely, when the score is Level III, the system prompts the user to retry without confirmation. At Level II, the systems asks the user for candidate confirmation.

4.2. COMMAND REPEAT and MASKING

Only *Yes* or *No* are recognized in the confirmation of word candidates in most voice response and speech recognition systems. However, the user has a tendency to repeat the last command even if an incorrect word candidate is presented. It has been reported that in practice, recognition faults were frequently caused by such command repetition [4]. Since such actions seem very natural for the inexperienced user, our system allows for command repetition except for the just confirmed word candidates. Allowing such command repeats to occur in the confirmation phase may cause same misrecognition. To avoid this, the just confirmed words are masked from the recognition objects in our system. In some situations, the masking of rejected words in the confirming phase also increases recognition accuracy. This works well only when it is reasonable to assume that the user will never reject a word already confirmed.

5. EVALUATION

To evaluate the performance of the fundamental design of the word spotter and the user interface, the following speaker-independent recognition tests were carried out using real telephone networks.

5.1. CONDITION OF EXPERIMENTS

All the tests were performed in a speaker-independent mode using the vocabulary of the ten Japanese digits (*ichi, ni, san, yon, go, roku, nana, hachi, kyuu and rei*), 9 Japanese time words (*juppun, nijuppun, sanjuppun, yonjuppun, gojuppun, rokujuppun, nanajuppun, hachijuppun, and kyuujuppun*) and 6 simple command words. The 10 state HMMs loaded in the word spotter were trained by a database containing 116 isolated tokens per word uttered by 116 male speakers through public telephone networks.

5.2. TESTS OF PARTIAL MATCH REDUCTION

Word spotting test results without the partial match reduction are given in Table 1. These experiments were performed using the 9 Japanese time words as the recognition vocabulary. The recognition target is only 8 compound words. A database consisting of short sentences containing 170 tokens obtained from 9 new male speakers was used. Eighty four percent of the test tokens consisted of redundant speech either prior to, or following target keywords. In the experiments, the word accuracy of the top candidate was too low, only 59%, because partial matching was observed in 36% of the compound words.

Table 1. Performance without Partial Matching Reduction

Algorithm	*Word Accuracy (%)* *Candidate Rank*			*Partial Match Occurrence (%)*
	1	2	3	
No Partial Match Reduction	59.0	89.0	95.0	36.0

The effect of the proposed reduction method of partial matching was tested using the same database as in Table 1. The results are also shown in Table 2. In the table, the simulated compound word accuracy is shown as *Without Compound Word* case, in which the recognition vocabulary used only 8 digits parts (ie. the whole digits set except *ichi, rei*) and one partial word *juppun*. In the simulation, higher priority was given to the candidate of the partial word than to the digits in making connected compound word candidates. To apply the proposed method, the constants K_a and K_b in equations (4)

and (5) were fixed *a posteriori* to values which give the best word accuracy.

Table 2. Performance with Partial Matching Reduction

Algorithm	*Word Accuracy (%)* *Candidate Rank* 1	2	3	*Partial Word Accuracy (%)*
Without Compound Word	75.0	86.0	88.0	70.0
With Partial Match Reduction	91.0	95.0	96.0	100.0

The proposed method substantially improved the word accuracy of the top and second candidates from 59% and 89% to 91% and 95%, respectively. The effectiveness of this method is very high because the improvement can be achieved without decreasing the word accuracy of the partial words. Even in the case of simulated *Without Compound Word* using 8 digits and a partial word only, the word accuracy of the top candidate was improved to 75%, but that of the second candidate, the third candidate, and the partial word decreased to 86%, 88%, and 70%, respectively. The reason for these results seems to be the poor accuracy of recognizing a single digit as a partial word. It appears, therefore, that the auxiliary use of digit recognition, as used in proposed method, is effective.

5.3. TESTS OF CONFIRMATION PROCEDURE

The proposed confirmation methods described in section 4. were tested using short interactive sentences including the 6 service menu selection command words. Eighty-two interactions spoken by 12 male speakers were used in this test. Fifty-six percent of the utterances were composed of redundant speech either prior to, or following target command words.

The average number of user utterances needed to ensure correct command recognition and the word accuracy in each try until the interaction was completed was used to evaluate the effectiveness of the ordinary and proposed methods: (1) ordinary confirmation (*Always Confirm*); (2) confirmation omission according to word correctness as described in 4.1. (*Only Adaptive Confirmation*); (3) allowance for command repeats and masking in confirmation as described in 4.2. (*Only Word Repeat and Mask*); (4) combination of (2) and (3).

Table 3. Comparison of Confirmation Procedure

Algorithm	*Word Acceptance (%)* *Number of Utterances* 1	2	3	*Average Number of Utterances*
(1) Always Confirm	-	95.0	98.0	2.11
(2) Only Adaptive Confirmation	85.0	96.0	98.0	1.24
(3) Only Word Repeat and Mask	-	95.0	100.0	2.05
(4) Combination of (2) and (3)	85.0	99.0	100.0	1.15

Actual test results and simulated results are shown in Table 3. The actual test results correspond to case (4) with both methods. The results for case (2) and case (3), were derived from case (4) results under the following assumptions. In case (2), the user's response to prompt confirmation was *Hai* or *Iie* (in English *Yes* or *No*), and the answer was always recognized correctly. For case (3), confirmation was always requested.

Eighty-five percent of first confirmations could be omitted, and as a result, if case (2) is used, the average number of utterances could be reduced from 2.11 (case (1) : *Always Confirm*) to 1.24 (case (2) : *Adaptive Confirmation*). The effect of adaptive confirmation could be reliably estimated because these improvements occurred without evaluation errors with a high probability of word correctness.

The inclusion of command word repeat and masking (case (3)) improved the command acceptance rate up to the third utterance from 98% (case (1)) to 100%. Although there were only four interactions with which the function was evaluated, it is concluded that case (3) will achieve faster operation given the same conditions as used in this test, such as the small number of recognition vocabulary items.

Furthermore, the command acceptance rate up to the second utterance could be improved 3% (case (2)) and 4% (case (3))by combining the adaptive confirmation method and the allowance of command repeat and masking method. This combination greatly improves the user interface.

6. CONCLUSION

The effectiveness of a modified word spotter and the 2 proposed methods proposed to improve the user interface of a word spotter based dialogue system were experimentally confirmed.

The likelihood score normalization using background HMM trained by non-keywords and noises increases system reliable more than using the ordinary frame length normalized scores. Using the proposed technique, the misrecognition rate for the recognition of a simple service menu command input task decreased by half.

A method was proposed to evaluate the occurrence of partial matches. It ranks compound words higher than the partial words in word spotting. For example, the word accuracy of 9 Japanese time commands, including one partial word and 8 compound words, could be improved from 59% to 91% using the proposed method.

The user interface enhancements are (1) adaptive confirmation to adaptively omit confirmation according to word correctness and (2) allowance for command repeats and masking in the confirmation phase. The enhancements significantly improve the user interface. Test results show that for 6 service menu command selections, the average number of utterances required to ensure command acceptance could be reduced from 2.11 to 1.15.

A high performance voice activated interaction system for public telephone networks can be constructed by using the proposed methods.

REFERENCES

[1] A. Imamura, Y. Suzuki : Speaker-Independent Word Spotting and a Transputer-based Implementation, ICSLP'90, pp.13.5.1–13.5.4.

[2] The Transputer Databook, INMOS, 1989.

[3] H. Tseng, et al. : Fuzzy Vector Quantization applied to Hidden Markov Modeling, ICASSP'87, pp.641-644.

[4] R. Touji, M. Kitai, T. Yoshida : Voice Command/Dialing System Field Test Results, SPEECH TECH'89, pp.262–265.