



DECLARATION OF CAROLINE CHAMPNEY

I, Caroline Champney, am over twenty-one (21) years of age. I have never been convicted of a felony, and I am fully competent to make this declaration. I declare the following to be true to the best of my knowledge, information and belief:

1. I am Publisher, STM Journals for Elsevier, Ltd. (“Elsevier”).
2. Elsevier is a neutral third party in this dispute.
3. I am not being compensated for this declaration.
4. Among my responsibilities as Publisher, STM Journals I act as a custodian of certain records for Elsevier.
5. I make this declaration based on my personal knowledge and information contained in the business records of Elsevier.
6. As part of its ordinary course of business, Elsevier, publishes and makes available technical articles. These publications are made available for public download through Elsevier’s digital library, ScienceDirect.
7. It is the regular practice of Elsevier to publish articles, abstracts, and other writings and make them available to the public through Elsevier’s ScienceDirect platform. Elsevier maintains copies of publications in the ordinary course of its regularly conducted activities.
8. The article identified below has been attached as Exhibit A to this declaration:

H. Lee and H. Ko, “Competing models-based text-prompted speaker independent verification algorithm”, Speech Communication, Vol. 48, Issue 1.
9. I obtained a copy of Exhibit A through ScienceDirect, where it is maintained in the ordinary course of Elsevier’s business. Exhibit A is a true and correct copy of the Exhibit, as it existed on or about January 15, 2025.
10. The article and abstract from Science Direct shows the date of publication for the first online version. Science Direct populates this information using persistent publication metadata that is embedded in the articles metadata. Once this publication metadata has been first calculated it is not changed.



11. H. Lee and H. Ko, "Competing models-based text-prompted speaker independent verification algorithm" was published in Speech Communication, Vol. 48, Issue 1 and made available online on July 5, 2005. The article is currently available for public download on ScienceDirect.

12. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

I declare under penalty of perjury that the foregoing statements are true and correct.

Date

20 January 2025

Signature



EXHIBIT A



Speech Communication

Volume 48, Issue 1, January 2006, Pages 28-44

Competing models-based text-prompted speaker independent verification algorithm

Heungkyu Lee ^a , Hanseok Ko ^b  ^a Department of Visual Information Processing, Korea University, Seoul, South Korea^b Department of Electronics and Computer Engineering, Korea University, 5ka 1 Anamdong, Sungbuk-gu, Seoul 136-701, South Korea

Received 8 September 2004, Revised 7 February 2005, Accepted 18 May 2005, Available online 5 July 2005.

 What do these dates mean?[Show less](#)  Outline |  Share  Cite<https://doi.org/10.1016/j.specom.2005.05.014> [Get rights and content](#) 

Abstract

In this paper, we propose competing models based on text-prompted speaker independent verification algorithm for an intelligent surveillance guard robot, wherein a robot prompts a code (i.e. word or phrase) for user entrance authentication. The proposed system requires the text-prompted speaker independent verification. In addition, it does not require a speaker dependent model and extra trained model as an alternative hypothesis for log-likelihood ratio test because of memory limitation. This is due to the given application scenario that an administrator changes the voice code every day for security reasoning and the targeting domain is unlimited. To resolve these issues, we propose to exploit the sub-word based anti-models for log-likelihood normalization through reusing an acoustic model and competing with voice code model. Anti-models using automatic production rules are set up in an initial time by using the statistical distance of phonemes against a voice code. The proposed system uses a two-pass strategy using a SCHMM-based recognition and verification step. In addition, a harmonics-based spectral subtraction algorithm is applied for a noisy robustness

on an outdoor environment. The performance evaluation is done by using a common Korean database, PBW452DB, which consists of 63,280 utterances of 452 isolated words recorded in silent environment.

Reading Assistant

Beta

Powered by GenAI

AI-generated content may vary in quality. [Learn more.](#)

[Find answers across publications ↗](#)

Previous

Next



Keywords

Utterance verification; LRT; Voice code verification

1. Introduction

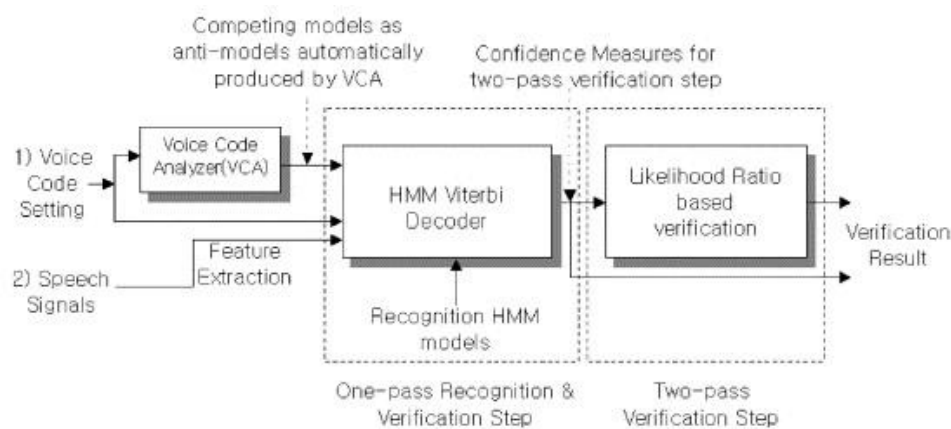
For surveillance tasks, much manpower is expended on sentry duty to guard the premises against unauthorized personnel for 24h. To lessen the time and effort expended by human guard, an intelligent surveillance guard robot is desirable. The surveillance guard robot takes the role of detecting and authorizing persons entering the perimeter of the secured area as well as passing status warnings. This system includes detection, recognition and tracking by using multiple sensors such as stereo cameras, IR cameras and array microphones. Under such an environment, a robot prompts for a code (i.e. word or phrase) for user entrance authentication. In the application scenario, an administrator changes a voice code every day for security reasons and the targeting domain is unlimited. Thus, a person authorized to enter a specific region should know the voice code previously before doing so. First, the proposed system speaks the password. Then, the authorized person responds with the corresponding voice code word. Such a proposed system is classified as text-prompted. In addition, as the target domain is unlimited and a person cannot register his identity information to the proposed system, the system cannot use the speaker dependent model. Because of this limitation, the proposed system utilizes the general acoustic model that is a speaker independent. As a result, the voice code verification system requires the text-prompted speaker independent verification (Sukkar and Lee, 1996). In addition, it does not require a speaker registration procedure to produce a speaker dependent model or an extra trained model such as a filler or garbage model for an alternative hypothesis in a log-likelihood ratio test (LRT). Thus, the proposed system is text-prompted speaker independent verification system. In this paper, the proposed system leverages on the combination of speech recognition and code verification

simultaneously. While the purpose of this paper is to authenticate the user's identity, it employs the concept of code verification as opposed to the speaker verification, which analyzes the unique quality of spectral content. Therefore, it is clearly not a speaker dependent system.

This paper is motivated by the task where the system does not need to know the speaker and has only to verify whether the uttered voice code is correct or not in a specific area. To cope with this task, we consider the utterance verification approach. Over the past several years, much research has been done in this field (Jiang and Lee, 2003, Rahim et al., 1997, Lleida and Rose, 2000, Wu et al., 2000). However, such a trial has not been undertaken yet. The utterance verification techniques are applied to several dialog systems in order to provide a user-friendly service. In addition, for reliability measurement (Jiang and Lee, 2003), this technique is applied to reject non-speech noises, out-of-vocabulary words, and potential recognition mistakes. Mostly, the confidence measure (CM) (Xin and Wang, 2001, Sankar and Wu, 2003, Rahim et al., 1995, Wessel et al., 2001) for this task is used to verify the uttered observation sequences after or during calculating the probability of a word W being recognized by an ASR system. Besides the utterance verification, a filler model or garbage model can be used for these purposes. Most algorithms require the extra model trained for a garbage model or anti-model. However, the limited memory size of our proposed embedded system prevents the algorithm from using and storing the extra alternative hypothesis model. Thus, the methods that do not require the extra trained model, and can reuse the acoustic model are investigated for the voice code utterance verification. Generally, a log-likelihood ratio test is applied to verify the utterance in this field of utterance verification where the verification step requires the alternative model for doing this task. To manage this problem, anti-models that are reusable from an acoustic model and can compete with a voice code model are considered.

For the general purpose of speech recognition, acoustic models are modeled to represent general characteristics of speech using HMM. The acoustic models are trained using speech databases of statistically significant number of people. Thus, the speech recognition system can recognize a spoken utterance without having speaker dependency. Thus, the acoustic models are speaker independent model. At this point, acoustic models can be modeled as word unit or phoneme unit. When it is modeled as a phoneme unit and independently, the acoustic models are context independent monophone. However, if we take into consideration both the left and the right neighboring phones, the acoustic models are context dependent models because a phoneme affects the utterance change. This is triphone, which is the model used in the proposed system. Such a phoneme model is modeled using HMM having states. In our system, we modeled it using three states, left-right HMM, which is then arranged in triphone models. Our proposed system uses a two-pass strategy using a SCHMM-based recognition (Huang et al., 2001) and verification step as in Fig. 1. In the first pass, recognition is performed via a conventional Viterbi beam search algorithm that segments the test utterance into the N -best strings of a phoneme hypotheses (Wessel et al., 2000). In the second pass, voice code verification is performed. It computes a confidence measure that determines whether or not to reject the recognized voice code. The voice code verification technique is formulated as a statistical hypothesis test where the task is to test the null hypothesis that is a

claimed voice code against the alternative hypothesis. Generally, the alternative hypothesis model as an anti-model is a previously trained model consisting of data that are mis-recognized or out-of-vocabulary generating false alarms, and noise tokens that also generates false alarms. However, the embedded hardware system requires limited and inexpensive memory. Thus, we reuse the general acoustic model as an anti-model to compete with the claimed voice code model. This is derived from the fact that the conventional speech recognition algorithms as a pattern classification problem use the maximum a posteriori (MAP) decision rule to find the most likely sequence of words. Respective phonemes can be modeled using HMM having states. When an uttered speech vectors are entered into decoding stage, likelihood scores are computed at all sequences of words until the end-point of speech is found. Thus, a likelihood score of phoneme models similar with uttered speech features become increased, or else a likelihood score has low value. When a person speaks a word, spoken word can be a completely different word or a similar word with a voice code word. The conventional speech recognition system obtains recognition success (false acceptance) when a person speaks a similar word because it uses the maximum a posteriori decision rule to find the most likely sequence of words. However, such a problem gives the high risk in a user authentication system. Thus, the anti-models to cope with this problem are required. Thus, in this paper, we propose the anti-models that are specific against a given voice code word model by using a regular production rules. The anti-models are automatically constructed using a grapheme to phoneme converter through the analysis of voice code text. The anti-models are based on phoneme and syllable units against the claimed voice code using the statistical distance of phonemes. In our system, the role of the anti-models is very important to verify an utterance. These anti-models compete with voice code model in the first pass, and are used as a normalization model for LRT in the second pass. Thus, we propose the voice code verification technique using the anti-models that are statistically opposed to the each phoneme of the voice code.



[Download: Download full-size image](#)

Fig. 1. Block-diagram of voice code verification.

This paper is organized as follows. In Section 2, we describe the voice code verification method using sub-word based anti-models. In Section 3, we describe the harmonics-based spectral subtraction

algorithm. Then, we conduct the representative experiments and discuss the results on performance of the proposed methods in Section 4. Finally, the concluding remarks are presented in Section 5.

2. Voice code verification

2.1. Competing models as anti-models

The a posteriori probability used for the likelihood normalization method in text-prompted speaker verification (Xiang and Berger, 2003, Tomoko and Furui, 1995) is given by

$$p(S_c, W_c/O) = \frac{p(O/S_c, W_c)p(S_c, W_c)}{\sum_i \sum_j \{p(O/S_i, W_j)p(S_i, W_j)\}} \approx \frac{p(O/S_c, W_c)}{\sum_i \sum_j p(O/S_i, W_j)} \quad (1)$$

where S_i is a speaker and S_c is the claimed speaker. W_i is a text, and W_c is the prompted text, $p(S_i, W_j)$ is the simultaneous probability for speaker i and text j . $p(O/S_c, W_c)$ is the probability of the claimed speaker's HMM corresponding to the prompted text. In the voice code verification, S_c and S_i can be ignored because this is speaker independent verification. Thus, Eq. (1) can be simplified as

$$p(W_c/O) \approx \frac{p(O/W_c)}{\sum_j p(O/W_j)} \quad (2)$$

This is the same with the likelihood normalization method for an utterance verification in a conventional ASR algorithm. In Eq. (2), W_c becomes the uttered word sequence, and $p(O/W_j)$ is approximated by the summation of the n highest likelihoods by using parallel phoneme HMM networks for all registered words. As a result, if the speaker information is ignored, the text-prompted verification technique is equal to the conventional ASR algorithms as a pattern classification problem using the maximum a posteriori (MAP) decision rule to find the most likely sequence of words as follows:

$$W_k = \arg \max_j L(O/W_j) \quad (3)$$

where $L(O/W_j)$ is the likelihood of an observation sequence O , given word W_j . In a text-prompted verification, this is the time that the number of a given word, j is equal to one. At this time, when someone speaks a false word for a text-prompted verification, we cannot verify the uttered word because we have no normalized models to test a likelihood score. Thus, we need the models to increase the likelihood score more than the one of the claimed voice code model when someone speaks the false voice code. However, we do not want the previously trained models (filler or garbage model) for likelihood normalization because the memory of our system is limited. The general filler-based methods, which try to calculate the probability of observing an utterance sequence, are previously trained. The utterance verification approach (Xiang and Berger, 2003, Tomoko and Furui, 1995, Benitez et al., 2000), mostly motivated by speaker verification (Ahn et al., 2002), also requires a trained model for likelihood normalization in LRT to verify the claimed content of a spoken utterance. In this case, methods that use just some phonemes such as cohort modeling or train the anti-models by using the speech interval that results in a recognition failure can be applied.

Otherwise, someone may be able to select an overall phoneme set as an anti-model except that phoneme by itself. However, too much time is required to calculate the log-likelihood of the overall phoneme sets. To cope with these problems, we reuse the original acoustic model for the alternative hypothesis model. Alternative hypothesis models as anti-models can be made automatically through the analysis of phoneme information with respect to the prompted text word. In this paper, we propose the construction method of anti-models using the statistical distance of phonemes against the voice code. This reused anti-model can be used to compete with the prompted voice code as follows:

$$W_k = \arg \max_j L(O/W_0, \overline{W}_1, \dots, \overline{W}_j) \quad (4)$$

where W_0 is a prompted voice code, and $\overline{W}$ is a competing model to be used for likelihood normalization. W_k is a concatenation of syllable units that can be written as

$$W_k = S_1^k S_2^k \dots S_N^k \quad (5)$$

where N is the number of syllables. In addition, a syllabic unit is a concatenation of context independent phoneme units that can be written as

$$S_N^k = P_{N,1}^k P_{N,2}^k \dots P_{N,M}^k \quad (6)$$

where M is the number of phonemes. Then, this context independent phoneme unit is changed into a context dependent phoneme unit after anti-phoneme units are constructed. Ultimately, the anti-phoneme units become the context dependent model.

As seen in Eq. (4), in the first pass, the Viterbi algorithm is employed to find the most likelihood word W_k . In this step, the prompted voice code is first verified as follows:

$$PVC = \begin{cases} \text{true} & \text{if } j = 0 \\ \text{false} & \text{else} \end{cases} \quad (7)$$

If the verification result PVC is true, the second pass to test a likelihood score is followed.

In this sub-section, we describe the automatic construction method of anti-models that opposes the statistical distance according to the manner and place of articulation, and tongue advancement and aperture (Park et al., 2002, William et al., 1997). When a voice code is presented to the system, the proposed system produces the anti-models automatically against a given voice code. If an arbitrary phoneme sequence or a free-phoneme sequence as an anti-phoneme sequence is used, the difference of likelihood scores depends on the phoneme sequence according to a given voice code during the Viterbi decoding. In the case of similar likelihood scores between models, out of vocabulary rejection can occur at the stage of confidence measure. In addition, when a person speaks a similar word, the false acceptance rate can increase. To resolve this problem, accurate modeling of anti-models against a given voice code is required. In this paper, we propose anti-models that can be produced

automatically using regular rules, and can compete with a voice code model in order to reject both similar and completely different words.

At first, the prompted voice text is automatically changed into a phoneme string, produced using a grapheme to phoneme (G2P) converter through text analysis. Then, the following rules for construction of anti-models are applied.

The voice code can be composed of a concatenation of a syllable, S , that is the set of phonemes. A voice code, W_0 is expressed by

$$W_0 = \{S_1, S_2, \dots, S_N\} \quad (8)$$

where N is the total number of syllables of a given voice code. At first, when a person produces a similar word, this may result in verification success. This occurs when any person says the word as follows:

$$\begin{aligned} \overline{W}_1^1 &= \{\overline{S}_1, S_2, \dots, S_N\}, \\ \overline{W}_2^1 &= \{S_1, \overline{S}_2, \dots, S_N\}, \dots, \\ \overline{W}_N^1 &= \{S_1, S_2, \dots, \overline{S}_N\} \end{aligned} \quad (9)$$

where N is the number of anti-syllable models for the first method and the variable, $\overline{S}$ is the anti-syllable. This sometimes results in verification success. To prevent false acceptance, we can use Eq. (9) as anti-models that compete with the voice code model when a person says a similar password. The anti-syllable model can be constructed using a concatenation of an anti-phoneme against each syllable unit as

$$\overline{S}_N = \{\overline{P}_1, \overline{P}_2, \dots, \overline{P}_M\} \quad (10)$$

The criterion to select the anti-phoneme is classified into two categories. Korean phonemes can be explained using a group according to the manner and place of articulation, and tongue advancement and aperture as shown in Table 1. By using this characteristic, anti-models can be produced using the one-to-many mapping method. Otherwise, anti-models can be produced using the one-to-one mapping method as shown in Table 2. This one-to-one matching method is produced manually considering the prosody generation characteristics of Korean phonemes. The first approach is to use a grouping method using the statistical distance of phonemes as in Table 1. The second approach involves the method to classify phonemes according to the manner and place of articulation, and tongue advancement and aperture as in Table 2, which is matched in order between phoneme and anti-phoneme. Table 3 depicts the matched phoneme set between Korean and English for comprehension. In this paper, we use the 44-phoneme set for Korean voice code verification. The anti-phoneme group is the one that is statistically far from the phoneme group. In first approach, the anti-phoneme is chosen as one of phoneme groups according to Table 1. The second approach finds the one-to-one matching between phoneme and anti-phoneme. Two approaches will be evaluated in Section 4.

Table 1. Phoneme grouping for construction of anti-phoneme model using statistical distance of phonemes

	Group	Phoneme	Group number	Anti-phoneme group number
Consonant	Back tooth	ㄱ ㅋ ㆁ ㆁ ㆁ	1	4
	Palate I	ㄷ ㅌ ㄷ ㄷ ㄷ ㄷ ㄷ	2	4
	Palate II	ㄴ ㄴ	3	1
	Lips	ㅂ ㅍ ㅂ ㅂ	4	1
Vowel	ㅏ series	ㅏ ㅑ ㅓ	5	6
	ㅓ series	ㅓ ㅕ ㅗ	6	5
	ㅗ series	ㅗ ㅛ ㅛ	7	5
	ㅗ series	ㅗ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ	8	9
	ㅗ series	ㅗ ㅛ ㅛ ㅛ ㅛ	9	8

Download: [Download full-size image](#)

Table 2. Anti-phoneme model production rules using statistical distance of phonemes

Phoneme to Anti-phoneme			
Consonant	Phoneme	ㄱ ㅋ ㆁ ㆁ ㆁ ㄷ ㅌ ㄷ ㄷ ㄷ ㄷ ㄷ ㄴ ㄴ ㅂ ㅍ ㅂ ㅂ	Manner and place of articulation
	Anti-phoneme	ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ ㅂ	
Vowel	Phoneme	ㅏ ㅑ ㅓ ㅓ ㅕ ㅗ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ	Tongue advancement and aperture
	Anti-phoneme	ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ ㅛ	

Download: [Download full-size image](#)

Table 3. Matching table between Korean and English phonemes

Consonant						Vowel					
ㄱ	g	ㅋ	kh	ㆁ	ch	ㅏ	a	ㅑ	yo	ㅓ	yv
ㄴ	n	ㄷ	th	ㄹ	j	ㅕ	v	ㅠ	yu	ㅗ	ya
ㄷ	d	ㅌ	ph	ㄲ	jj	ㅛ	o	ㅜ	yae	ㅜ	we
ㄹ	r	ㅎ	h	ㅇ	ng	ㅜ	u	ㅡ	ye	ㅡ	eui
ㅁ	m	ㅈ	gg	ㅅ	ss	ㅡ	eu	ㅗ	wa	ㅗ	wv
ㅂ	b	ㅊ	dd			ㅣ	i	ㅗ	we	ㅗ	we
ㅅ	s	ㅍ	bb			ㅓ	e	ㅗ	wi	ㅓ	e

Download: [Download full-size image](#)

To produce the anti-model for each syllable, the corresponding syllable in the prompted voice code is changed into an anti-syllable using the anti-phoneme according to [Table 1](#) or [Table 2](#) after the text is changed into the phoneme list using a G2P converter, where it needs a parsing process to find each syllable that is composed of a consonant and vowel. In the Korean language, a syllable can be composed of “V”, “C+V”, “C+V+C” and “V+C” where V is the vowel and C is the consonant. A Korean syllable can be classified into nine groups as illustrated in [Table 4](#). Using this rule, a given text is classified into the syllable lists.

Table 4. Korean syllable production rules

Syllable	Word rules	Group	No	Comment
CV	CV/CV	CV/CV	1	
	CV/CVC	(PART1)		
	CV/VC	CV/V	2	
	CV/V	(PART2)		
CVC	CVC/CV	CVC/V	3	
	CVC/CVC	(PART3)		
	CVC/VC	CVC/V	1	Follows rule part 1 according to Korean utterance rules
	CVC/V	(PART4)		
VC	VC/CV	VC/C	4	

Syllable	Word rules	Group	No	Comment
	VC/CVC	(PART5)		
	VC/VC	VC/V	5	Follows rule part 7 according to Korean utterance rules
	VC/V	(PART6)		
V	V/CV	V/CV	5	
	V/CVC	(PART7)		
	V/VC	V/V	6	
	V/V	(PART8)		

The second, when any person utters a similar word that includes all parts of a prompted voice code, verification success often results. This is particularly true when a false text is uttered as follows:

$$\begin{aligned}\overline{W}_1^2 &= \{S_1, S_2, \dots, S_N, \overline{S_{N+1}}\}, \\ \overline{W}_2^2 &= \{S_1, S_2, \dots, S_N, \overline{S_{N+1}}, \overline{S_{N+2}}\}, \dots, \\ \overline{W}_M^2 &= \{S_1, S_2, \dots, S_N, \overline{S_{N+1}}, \overline{S_{N+2}}, \dots, \overline{S_{N+M}}\}\end{aligned}\quad (11)$$

where M is the number of anti-syllable models competing with a given voice code model, and anti-syllable, $\overline{S_{N+M}}$ is matched to its syllable, S_N . To prevent this case, we use Eq. (11) as anti-models. The anti-syllable model also can be constructed using Table 1 or Table 2.

The third, when any person says a similar word that is some part of the password text, this also often results in verification success. This happens when any person says a text as follows:

$$\begin{aligned}\overline{W}_1^3 &= \{S_1\}, \\ \overline{W}_2^3 &= \{S_1, S_2\}, \dots, \\ \overline{W}_{N-1}^3 &= \{S_1, S_2, \dots, S_{N-1}\}\end{aligned}\quad (12)$$

where $N-1$ is the number of anti-syllable models. To prevent this case, we use Eq. (12) as anti-models. In addition, we can use anti-models contrary to Eq. (12) as follows:

$$\begin{aligned}\overline{W}_1^4 &= \{\overline{S_1}\}, \\ \overline{W}_2^4 &= \{\overline{S_1}, \overline{S_2}\}, \dots, \\ \overline{W}_{N-1}^4 &= \{\overline{S_1}, \overline{S_2}, \dots, \overline{S_{N-1}}\}\end{aligned}\quad (13)$$

Finally, the following anti-model is applied:

$$\overline{W}_1^5 = \{\overline{S_1}, \overline{S_2}, \dots, \overline{S_N}\} \quad (14)$$

In the proposed anti-model equations for automatic production, Eqs. (9), (11), (12) represent anti-models to reject similar words with a voice code. In addition, Eqs. (12), (13) represent anti-models to reject completely different words. After these anti-models are constructed through the analysis of a given voice code, all anti-models are used to compete with a voice code model. These models would increase the likelihood score of anti-models while the likelihood score is reduced when someone speaks a false word or phrase.

2.2. Voice code verification using sub-word based anti-models

In the second pass, the voice code verification task is applied. Generally, a sub-word based utterance verification or out-of-vocabulary rejection method is based on a likelihood ratio test. The major difficulty with an LRT in utterance verification is modeling the alternative hypothesis, where the true distribution of the data is unknown and an alternative hypothesis usually represents a very complex and composite event. Given a decoded sub-word in an observed segment, we need a decision rule by which we assign the sub-word to either hypothesis H_0 or H_1 . For the binary testing problem, one of the most useful tests for decision is the Neyman–Pearson Lemma. For a given number of observations, which minimizes the error for one class while maintaining the error for the other class constant, the likelihood ratio test is as follows:

$$\text{LRT}(X) = \frac{P(X/H_0)}{P(X/H_1)} = \frac{P(O_n/\lambda_n)}{P(O_n/\bar{\lambda}_n)} \geq \eta \quad (15)$$

where H_0 means that the hypothesis is true and H_1 means that the hypothesis is false, λ is the sub-word model, $\bar{\lambda}$ is the anti-sub-word model, and X is the uttered input observation that the number of sub-words is N . Hence, Eq. (15) accepts the hypothesis when the likelihood ratio is greater than the given threshold, η . If the alignment of observations with respect to the sub-word unit is obtained, the LRT is performed using log-likelihood of a sub-word model.

The sub-word alignment and log-likelihood score are obtained on a log domain through the Viterbi segmentation. For the likelihood ratio normalization, an average frame log-likelihood ratio (LLR), R_n is defined as

$$R_n = \frac{1}{l_n} [\log P(O_n/\lambda_n) - \log(O_n/\bar{\lambda}_n)] \quad (16)$$

where $\bar{\lambda}$ is an anti-model automatically produced by the rule as in the previous sub-section and $\log(O_n/\bar{\lambda}_n)$ is the likelihood score approximated by the summation and averaging of the n_{Best} highest likelihood scores using the parallel phoneme HMM networks by competing with each other. Thus, Eq. (16) can be written as

$$R_n = \frac{1}{l_n} \left[\log P(O_n/\lambda_n) - \frac{1}{n_{Best}} \sum_{m=1}^{n_{Best}} \log(O_n/\bar{\lambda}_m) \right] \quad (17)$$

In Eq. (17), the dynamic range of a sub-word based log-likelihood ratio is high. This can affect the overall performance. One way to limit the dynamic range of the sub-word confidence measure is to

use a sigmoid function of the following form [2]:

$$U_n = \frac{1}{1 + \exp(-\alpha \times (R_n - \tau))} \quad (18)$$

where τ and α are location and weighting parameters. The log confidence score has a slope of α when the log-likelihood score is less than zero. The τ reduces the dynamic range caused by frames having a high likelihood ratio. Thus, we can control the effect of the log-likelihood score with τ and α .

2.3. Confidence measure

For effective voice code verification, we need to define a function to combine the results of sub-word tests. The confidence measure (CM) for an input utterance O can be represented as

$$\text{CM}(O) = f(\text{CM}_1, \text{CM}_2, \dots, \text{CM}_N) \quad (19)$$

where $f()$ is the function to combine the verification scores. This is defined as a function of their likelihood ratios. It can be considered as a joint statistic for overall word-level verification.

The first confidence measure is based on frame-duration normalization, which is defined as follows:

$$\text{CM}_1 = \frac{1}{L} \sum_n^N (l_n * R_n) \quad (20)$$

where N is the total number of sub-words in the utterance, and L is the total number of utterance frames $L = \sum_{n=1}^N l_n$. The second one is based on syllable segment-based normalization. It is a simple average of log-likelihood of all the syllables:

$$\text{CM}_2 = \frac{1}{N} \sum_n^N R_n \quad (21)$$

$$\text{CM}_3 = \exp\left(\frac{1}{N} \sum_{n=1}^N \log R_n\right) \quad (22)$$

$$\text{CM}_4 = \frac{1}{N} \sum_{n=1}^N U_n \quad (23)$$

$$\text{CM}_5 = \exp\left(\frac{1}{N} \sum_{n=1}^N \log U_n\right) \quad (24)$$

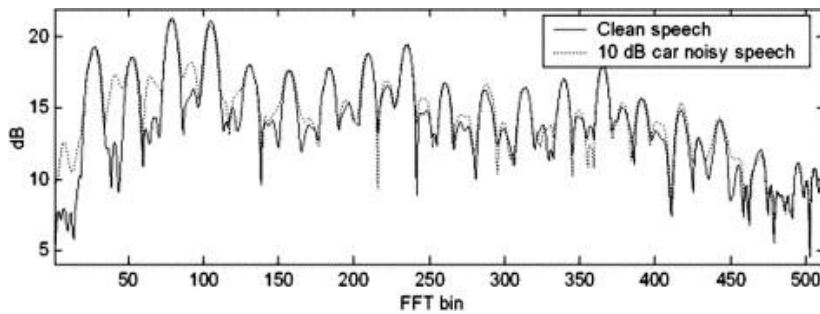
where Eqs. (21), (22) are the arithmetic and geometric means of the unweighted sub-word level confidence scores, and Eqs. (23), (24) are the arithmetic and geometric means of the sigmoid weighted sub-word score.

For every confidence measure, a specific threshold is set up. If its value is below the threshold, the candidate is discarded from the verification task, resulting in a voice code verification failure.

3. Harmonics-based spectral subtraction

3.1. Speech-dominant region vs. noise-dominant region

In speech, voiced speech segment has peaks positioned periodically in spectrum due to the vibration of vocal cords. These points are very critical in speech sound perception. Fig. 2 illustrates this phenomenon with a sample spectrum of a speech frame capturing the pronunciation /oh/ in one utterance contained in Aurora 2 corpus. Note that at the peaks, their amplitudes are far greater than the amplitudes at the points between the adjacent small peaks, or side-lobes. Also, it is observed that the degree of corruption by the noise in the peaks is less than the degree of corruption at the points over the side-lobes. From this observation, it can be deduced that in speech spectrum, the speech-dominant regions exist over or near the peaks and the noise-dominant regions exist over or near the side-lobes. However, the illustration as displayed could be misleading. It does not show, for example, that the speech can be more corrupted in some frequency regions.



[Download: Download full-size image](#)

Fig. 2. Example spectrum of a speech frame (pronunciation /oh/ by female).

3.2. Peak point detection and segmentation

It is known that the spectral harmonics are located at the points of multiples of fundamental frequency. First, using an autocorrelation function, the indices of local maxima in autocorrelation values are found. Among those indices, we find the fundamental frequency f_0 by taking the reciprocal of the index that represents the maximum value. Second, by applying the nonlinear smoothing method, f_0 is modified to more accurately reflect the true fundamental frequency. This algorithm works well even in a high degree noisy environment. The autocorrelation function is expressed as follows:

$$\phi(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)x(n+\tau) \quad (25)$$

We then establish the index as k_0 for the frequency bin that corresponds to f_0 . Using k_0 , the peak points (harmonics) h_l are determined as $h_1=k_0, h_2=2k_0, h_3=3k_0, \dots, h_L=Lk_0$, where L is the index of the last component of harmonics. Using this procedure, we determine all harmonics in the spectrum of an input frame, which is assumed to be a speech-dominant region. We illustrate the validity of the proposed method with spectrograms and the detection of peak point. For the purpose of implementing the proposed scheme, the frequency axis is divided into several non-overlapping

bands in the following form:

$$[1, h_1], [h_1, h_1 + \frac{k_0}{2}], [h_1 + \frac{k_0}{2}, h_2], \dots, [h_L, \frac{\text{FFTorder}}{2} + 1] \quad (26)$$

3.3. Voice activity detection and noise estimation

Note that in each input frame, whether speech frame or not, the fundamental frequency is calculated. As a result, the value of autocorrelation is available at each frame. To begin, we calculate the ratio of this value over the frame energy since this ratio appears to be a more effective measure to distinguish a speech region from a non-speech region than the mundane energy measure. Then, two separate procedures are employed for robust detection of speech vs. non-speech regions. First, a smoothing procedure is carried out. If this value is greater than the threshold 1.4th_{t-1} as defined by the following expressions, the relevant frame is decided as a 'speech frame'.

$$\text{If } \frac{\phi(\tau')}{E_t} > 1.4\text{th}_{t-1} \text{ then 'speech frame' and} \quad (27)$$

$$\text{th}_t = 0.95\text{th}_{t-1} + 0.05 \frac{\phi(\tau')}{E_t}$$

$$\text{else 'non-speech frame' and} \quad (28)$$

$$\text{th}_t = \text{th}_{t-1}$$

Note that th_t indicates the threshold of the t th frame, E_t indicates the t th frame energy and $\phi(\tau')$ is the autocorrelation value corresponding to the f_0 of the relevant frame.

Secondly, for frames determined as 'speech frame', if the detected fundamental frequency is not in the 50–600Hz, then it is regarded as 'non-speech frame' once again. For the noise estimation, we assume that any starting input speech is followed by a silence or background noise segment corresponding to 10 frames (about 100ms). If the input frame is determined as a 'non-speech frame', the estimated noise spectral magnitude is updated as follows:

$$|\hat{B}_t(k)|^2 = \lambda_B |\hat{B}_{t-1}(k)|^2 + (1 - \lambda_B) |Y_t(k)|^2, \quad 0.65 \leq \lambda_B \leq 0.998 \quad (29)$$

where t is the frame index, k is the index of FFT bins and $\lambda_B=0.95$ is used.

3.4. Spectral subtraction rules

Spectral subtraction rules are applied according to whether the frame is speech or not. In speech region, the following linear function is applied:

If $k \in [1, h_1]$ or $k \in [h_{l-1} - 1, h_l - 1] + \frac{k_0}{2}$, then

$$\gamma(k) \quad (30)$$

$$= \frac{\alpha_{\min} - \alpha_{\max}}{h_l - \left(h_{l-1} + \frac{k_0}{2}\right) - k} + \alpha_{\max}$$

$$\delta(k) \quad (31)$$

$$= \frac{\beta_{\max} - \beta_{\min}}{h_l - \left(h_{l-1} + \frac{k_0}{2}\right) - k} + \beta_{\min}$$

If $k \in [h_l, h_L]$ or $k \in [h_L, \frac{\text{FFTorder}}{2} + 1] + \frac{k_0}{2}$, then

$$\gamma(k) \quad (32)$$

$$= \frac{\alpha_{\max} - \alpha_{\min}}{\left(h_l + \frac{k_0}{2}\right) - h_l - k} + \alpha_{\min}$$

$$\delta(k) \quad (33)$$

$$= \frac{\beta_{\max} - \beta_{\min}}{\left(h_l + \frac{k_0}{2}\right) - h_l - k} + \beta_{\max}$$

where $l=1,2,\dots,L$ and L is the index of the last component of harmonics.

$\gamma(k)$ applies the maximum over-subtraction factor to the midpoint at each component. Then, the over-subtraction factors of points existing in the interval between those points are interpolated

linearly. In contrast, $\delta(k)$ applies the maximum floor factor to each harmonic component and the minimum floor factor to the midpoint at each component (Beh and Ko, 2003). Then, other floor factors between these points are generated by linear interpolation.

In this procedure, we set $\alpha_{\max}=8$, $\alpha_{\min}=1$, $\beta_{\max}=0.15$ and $\beta_{\min}=0.05$. Finally, the proposed subtraction rule can be represented as follows:

$$|\widehat{X}(k)|^2 = \begin{cases} |Y(k)|^2 \left(1 - \gamma(k) \frac{|\widehat{B}(k)|^2}{|Y(k)|^2}\right) & \text{if } |Y(k)|^2 - \gamma(k) > \delta(k)|Y(k)|^2 \\ \delta(k)|Y(k)|^2 & \text{else} \end{cases} \quad (34)$$

Meanwhile, in the non-speech region, we apply the subtraction rule in the same way as conventional methods with $\alpha_{\max}$ and $\beta_{\min}$.

4. Evaluation of the proposed system

4.1. Experimental condition

For speech input to verify the uttered voice code, the sampling rate is 16bits PCM. In cases of speech recognition and text-to-speech engine, sampling rate is the same. The isolated word decoding module with dynamic vocabularies to reduce computing time and memory size is optimized, and speech signals are analyzed within 125ms frame with 10ms lapped into 26th order feature vector that has 13th order MFCCs including log energy and their 1st derivatives. The training data set consists of about 120,000 utterances of 6000 isolated words sets recorded in an office environment. In addition, we used a different speech corpus for testing the data set, which is PBW452DB, Korean Common DB, which consists of 63,280 utterances of 452 isolated words recorded in a silent environment. For this recording, 70 men and women uttered one word twice. The microphones used in the training data collection and testing data collection are different. Among the 63,280 utterances in the test data, 140 utterances of one word are used as the voice code and the 140 utterances of any other word are used for the out-of-vocabulary data. Then, the false acceptance rate (FAR) and false rejection rate (FRR) of the 452 computed voice codes are averaged. The threshold is gradually increased as much as 0.1.

4.2. Experimental result

In Korean language, the consonant is produced according to the manner and place of articulation, and the vowel is produced according to the tongue advancement and aperture whenever a person utters a word. The proposed anti-phoneme model is based on the Korean phonetic theory. In this paper, competing models are considered to increase the likelihood scores against a score of the voice code model in order to reject either a word that is completely different or a similar word when a person utters a false code. In addition, there is a need for automatic production rules of anti-models because

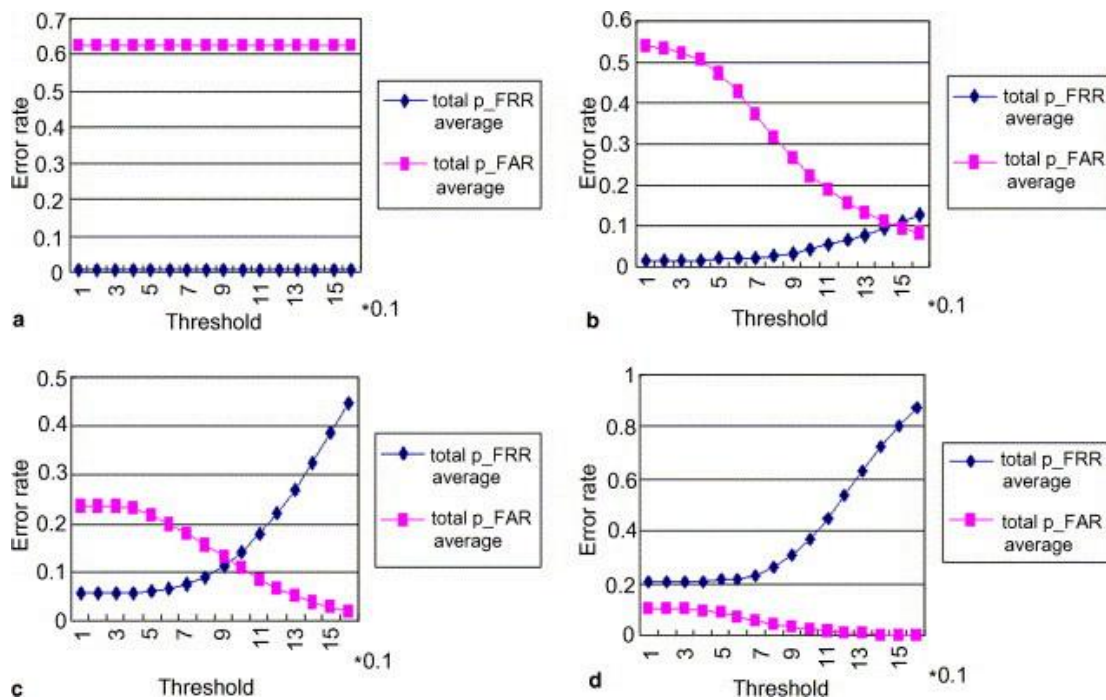
the voice code is changed every day. In this case, specific anti-model is desirable against a given voice code model since its performance may depend on the given voice code word.

In this paper, we apply an utterance verification technique using an N -best alternative hypothesis model for likelihood normalization in an LRT. In our previous work (Kim and Ko, 2003) on utterance verification, several utterance verification methods were simulated. Our method, the Bayesian fusion technique showed a performance higher than any other method. However, we apply the 5-best technique that is easy to implement and has a low computing time on a DSP board.

For a voice code verification simulation, an anti-model making routine for likelihood normalization against a log-likelihood score of voice code is implemented as in Section 2. The simulation is done using the total anti-models as in Eqs. (9) and (11), (12), (13), (14). At first, the two approaches using Table 1, Table 2 as described in Section 2.1 are compared. Four categories of methods are used for evaluation. Method 1 is the anti-model set using Eq. (14), method 2 is the anti-model set combining Eqs. (11), (14), method 3 is the anti-model set combining Eqs. (11), (12), (14), and method 4 is the anti-model set combining Eqs. (9), (11), (12), (14). The simulation result shows that the second approach is a bit higher than the first approach. The detection probability of the second approach is 2% higher than the one of the first approach, and the FAR of the second approach is 1% lower than the one of the first approach as in Table 5. Thus, we choose the second approach as in Fig. 3.

Table 5. Comparison results between the first approach using “one-to-many mapping method” as in Table 1 and the second approach using “one-to-one mapping method” as in Table 2 (threshold=0)

Method	First approach		Second approach	
	FRR	FAR	FRR	FAR
1	0.0088179	0.5827750	0.0087547	0.5759799
2	0.0171302	0.5166876	0.0168885	0.5069785
3	0.0567947	0.2383059	0.0561109	0.2262200
4	0.2288869	0.0987989	0.2053726	0.0840138

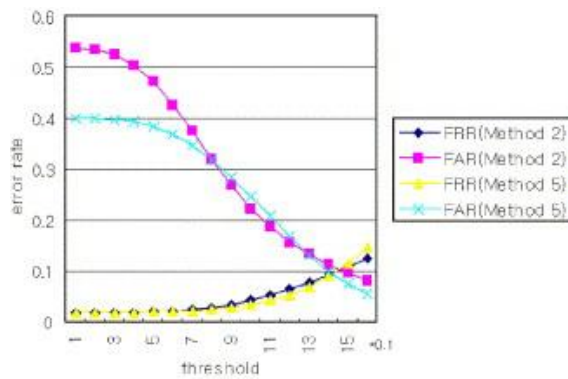


[Download: Download full-size image](#)

Fig. 3. Simulation results of the second approach using statistical distance of phonemes. Simulation results of (a) method 1, (b) method 2, (c) method 3 and (d) method 4.

As shown in Fig. 3(a), method 1 has a high FAR while FRR is low. In method 2 (Fig. 3(b)) and method 3 (Fig. 3(c)), the EER is 0.09 and 0.125 respectively. However, it cannot cope with various situations as described in Section 2. In addition, method 4 has no EER. Thus, we are not able to decide a stable threshold because methods 1 and 2 have no EER. As shown in Fig. 3(a)–(d), respective anti-models cannot cope with either completely different or similar words, and it is difficult to decide a threshold even for methods 1 and 2.

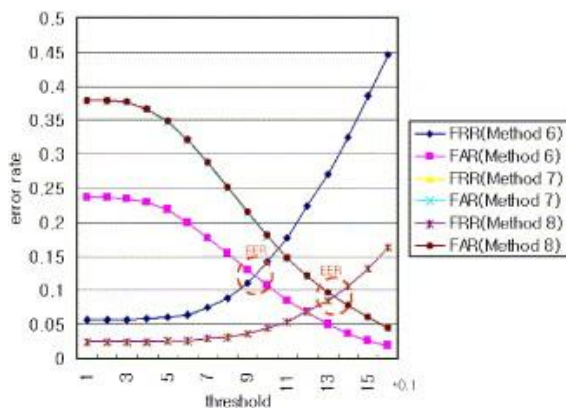
Thus, we did extra simulations on the basis of method 2. Fig. 3(b) shows the result of method 2 when we use the variable, $M=1$, of Eq. (11). When we apply the variable, $M=3$, of Eq. (11) in method 2, FRR and FAR are improved. Fig. 4 shows this comparative result. Method 5 is the anti-model set that variable M of Eq. (11) is 3 in method 2. This result shows that anti-models having long sequences of anti-phonemes reduce the false acceptance rate considerably when a person speaks a word that has long false word sequence.



[Download: Download full-size image](#)

Fig. 4. Comparison result of methods 2 and 5.

Using the result from Fig. 4, we combined the anti-model equations of method 2 with Eqs. (12), (13). In Fig. 5, method 6 is the anti-model set combining Eq. (12). Method 7 is the anti-model set combining Eq. (13). Method 8 is the anti-model set combining Eqs. (12), (13). Methods 7 and 8 showed similar results. However, method 8 is a bit improved and also can cope with various utterances of people very well.

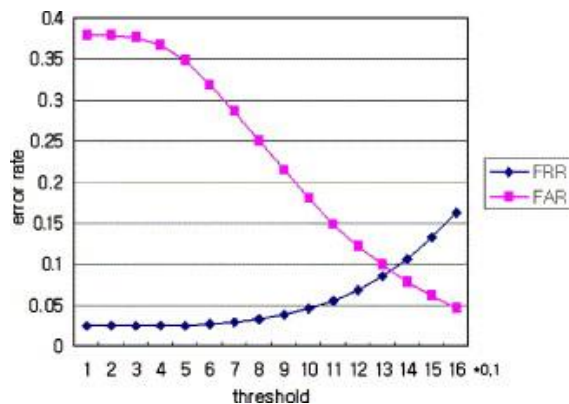


[Download: Download full-size image](#)

Fig. 5. Comparison result of the methods 6–8.

Finally, we combine the anti-models using Eq. (9) with method 8 as shown in Fig. 6. The curve shape and result of the final method is similar with method 8 of Fig. 5. However, the final result is a bit improved over method 8 of Fig. 5. In addition, the EER is 0.08. This result is improved by 16% over the utterance verification result as shown in Table 6. Aimed at applying to utterance verification, the “Sub-word LRT”, “Phonetic filler”, “5-best”, “256 mix GMM” and “Bayesian fusion” employed an arbitrary phoneme sequence or a general free-phoneme sequence for comparative evaluation. The Sub-word LRT (Likelihood Ratio Test) is done at a sub-word unit (phoneme or syllables) during Viterbi decoding and the results are combined for final decision. The phonetic filler or 256 mix GMM uses an arbitrary phoneme sequence or previously trained models such as garbage models that is trained using predictable words set. When a word network is constructed, these models are linked to

this word network. So, during Viterbi decoding, uttered word (feature vectors) is checked whether the uttered word is within the given recognition list (constructed phoneme sequence) or not. In 5-best method for LRT, all path sequences computed in Viterbi decoding procedure are not used. N -best sequences having maximum likelihood scores are computed for LRT. In the case of Bayesian fusion method applying to utterance verification problem, conventional confidence measure modeling and evaluation methods can be treated as local sensors which detect or verify acoustic events concerned. Final decision is made in global fusion center, which combines the local decisions sent by each local utterance verifier. However, Bayesian fusion method considers parallel configuration for utterance verification for utterance verification where each sensor observes common phenomenon and makes local decision regarding it. Thus, the objective of Bayesian formulation is to minimize the Bayesian risk. It is to find optimal decision rules for each local sensor as well as a global one.



[Download: Download full-size image](#)

Fig. 6. Final result using all anti-models.

Table 6. Comparison results with other utterance verification methods (threshold=1.2)

Threshold	Method					
	Sub-word LRT	Phonetic filler	5-best LRT	256 mix GMM	Bayesian fusion	Proposed method
1.2	0.1882	0.1448	0.2432	0.1862	0.1383	0.08

In this paper, we proposed automatic production rules of anti-models such that the performance would not depend on a code. In order to accomplish the code independence, anti-models are produced against the specific voice code. Eqs. (9), (11), (12) represent the anti-models to reject similar words. Eqs. (13), (14) are anti-models to reject completely different words.

As shown in Fig. 3, specific equations for anti-models have low performance when respective equations are applied alone because they are able to cope only with the specific words. However, combined anti-models result in more robust performance because they can cope with various acoustic situations. If the modeling of anti-models is not precisely done and a few anti-models are

applied, the performance should depend on a given voice code or uttered words. In contrast, if a lot of anti-models are applied and thus very precise modeling of anti-models are considered, some words can be rejected in the confidence measure stage because the likelihood score of anti-models that represent a similar word against a voice code word has increased. Therefore, an optimal number of anti-models should be applied to obtain a robust performance.

This system usually is utilized in an outdoor surveillance region, which requires a noise robust voice code verification to cope with not only environmental noise, but also other white noise. To resolve this problem, a harmonics-based spectral subtraction algorithm is applied for preprocessing the noise. First, an experiment is conducted by the Aurora 2 evaluation procedure under a continuous digits recognition tasks. Test sets are reproduced using TIDigits of which the entire speech data are down-sampled to 8kHz and various realistic noises are added artificially. The feature vector order is 39 and is composed of 13 order static MFCC (c1–c12+log energy) and its derivatives and accelerations. For comparison, a spectral subtraction algorithm and nonlinear algorithm are evaluated as in Table 7. As seen in Table 7, HSS shows that the proposed algorithm is more robust than other algorithms. A notable advantage of the proposed scheme is that it does not require an exact SNR estimate under various noise conditions. If a noise characteristic is stationary and each magnitude of harmonics is not masked by a noise, the SS technique itself may work alone. However, real-life conditions are not all that cooperative. In the case of SS, due to the fact that the non-stationary characteristic of noise a posteriori and a biased SNR are estimated inaccurately, there may be a distortion in the shape of harmonics by subtracting the noise magnitude more in the harmonics region and less in the side-lobes region. As a result, this condition will lead directly to a poor performance in the ASR system. However, HSS also has a remedial capability that corrects the described deficiency. The key to improving the performance of HSS is pinpointing the harmonics corresponding to the spectrum.

Table 7. Summary of overall word accuracies on Aurora2 mis-matched training/testing condition (%)

	Baseline	SS	HSS
Baseline	60.06	77.89	80.59
CMN	71.16	78.56	82.00

Next, HSS is applied to the voice code verification experiment, which is also conducted using PBW452DB. The simulation result is shown in Table 8. In a babble noise environment, EER did not show a rapid decrease of EER, while in white noise, EER showed a rapid decrease of EER. However, it brought about a 40% relative improvement than when there was no harmonics-based spectral subtraction algorithm.

Table 8. Equal error rate (EER) of voice code verification under babble and white noise environments

		Equal error rate (EER)			
		Clean	5dB	10dB	15dB
Clean DB	FRR	0.076738	–	–	–
	FAR	0.109624	–	–	–
Babble noise	FRR	–	0.088653	0.088069	0.089238
	FAR	–	0.096128	0.095923	0.095385
White noise	FRR	–	0.518268	0.376042	0.221871
	FAR	–	0.362295	0.320828	0.218299

4.3. Discussion

Our proposed method for a text-prompted and a speaker independent verification showed that it provides a voice code verification function without an extra trained model for the likelihood normalization in N -best based LRT. The key point is to use the competing models that are anti-models using a statistical distance of phonemes as the alternative model always follows the same state as the target model.

The conventional speech recognition algorithms as a pattern classification problem use the maximum a posteriori (MAP) decision rule to find the most likely sequence of words. Respective phonemes can be modeled using HMM having states. When uttered speech vectors enter the decoding stage, likelihood scores are computed at all sequences of words until the end-point of speech is found. Thus, a likelihood score of phoneme models similar to the uttered speech features increase, or else the likelihood score will be assigned a low value. When a person utters a word, it can be either a completely different word or a similar word with a voice code word. The conventional speech recognition system obtains recognition success (false acceptance) when a person speaks a similar word because it uses the maximum a posteriori decision rule to find the most likely sequence of words. However, such a problem presents high risk in a user authentication system, requiring anti-models to cope with this problem. Thus, in this paper, we propose anti-models that are specific to a given voice code word model by using regular production rules.

Thus, if we can perform the modeling of the alternative hypothesis very well, the voice code verification task can be solved by competing against each other without extra trained models such as filler or garbage models. In some cases, traditional LLR does not find the most representative alternative hypothesis (Leung and Fung, 1999). Thus, the modeling of an alternative model is a crucial issue for the voice code verification.

From the simulation result, we knew that the use of numerous anti-models degraded detection probability while the use of few anti-models degraded the false acceptance rate. Thus, the optimal number of anti-models that can compete with the voice code model should be used. In addition, outdoor noise is an important issue that should be considered. Under this condition, the speaker verification rate is very low while the voice code verification rate is the same. Even though we applied a harmonic-based spectral subtraction algorithm, some other algorithms should also be used for compensating the verification rate in an outdoor environment where wind or rain noise exists, and so on. Considering its application to an outdoor environment, some kinds of feature compensation algorithms such as the multi-variate Gaussian-based feature compensation or the parallel combined mixture model ([Kim et al., 2003](#)) should be considered.

This system can be utilized by combining it with other security algorithms. To prevent and compensate for the false rejection or verification failure, we can use other methods such as face recognition and fingerprint recognition using an image sensor. Thus, the proposed system can make use of verifying suspicious persons with other verification techniques. By combining image techniques into our proposed system for detection, tracking and recognition of people, more reliable detection and recognition rate can be achieved.

5. Conclusions

We proposed a voice code verification method for a text-prompted speaker independent verification, based on an utterance verification method using the sub-word based anti-models that employ the statistical distance of phonemes against the voice code. Our proposed method for a text-prompted and speaker independent verification provided a voice code verification function without an extra trained model such as filler or garbage models for the likelihood normalization through the reuse of a general acoustic model. At this point, the use of numerous anti-models degraded the detection probability while the use of few anti-models degraded the false acceptance rate. Thus, the optimal number of anti-models that can compete with the voice code model should be used.

In experiment, the performance evaluation is done by using a common Korean database, PBW452DB, which consists of 63,280 utterances of 452 isolated words recorded in a silent environment. The result was improved by 16% over the utterance verification result presented in [Table 6](#). In addition, simulation result showed that the performance is higher under noisy environment than in any other algorithms when we applied the harmonics-based spectral subtraction algorithm compared to general spectral subtraction and nonlinear spectral subtraction.

Acknowledgement

This work was supported by grant No. 10012805 from the Korea Institute of Industrial Technology Evaluation & Planning Foundation.

[Recommended articles](#)

References

- [Ahn et al., 2002](#) Ahn, S.J. et al., 2002. On effective speaker verification based on subword model. In: International Conference on Spoken Language Processing (ICSLP) 2002, Denver, USA, September, Vol. 2, pp. 1361–1364.
[Google Scholar ↗](#)
- [Beh and Ko, 2003](#) Beh, J.H., Ko, H., 2003. A novel spectral subtraction scheme for robust speech recognition: spectral subtraction using spectral harmonics of speech. ICME 2003, July, pp. III 633–III 636.
[Google Scholar ↗](#)
- [Benitez et al., 2000](#) M.C. Benitez, *et al.*
Different confidence measures for word verification in speech recognition
Speech Comm., 32 (2000), pp. 79-94
 [View PDF](#) [View article](#) [View in Scopus ↗](#) [Google Scholar ↗](#)
- [Huang et al., 2001](#) X. Huang, *et al.*
Spoken Language Processing
Prentice Hall PTR (2001)
[Google Scholar ↗](#)
- [Jiang and Lee, 2003](#) H. Jiang, C.H. Lee
A new approach to utterance verification based on neighborhood information in model space
IEEE Trans. Speech Audio Process., 11 (5) (2003)
[Google Scholar ↗](#)
- [Kim and Ko, 2003](#) Kim, T.Y., Ko, H., 2003. Utterance verification under distributed detection and fusion framework. Eurospeech 2003, September, pp. 889–892.
[Google Scholar ↗](#)
- [Kim et al., 2003](#) Kim, W.I. et al., 2003. Feature compensation scheme based on parallel combined mixture model. Eurospeech 2003, September, pp. 677–680.
[Google Scholar ↗](#)
- [Leung and Fung, 1999](#) Leung, L.K., Fung, P., 1999. A more efficient and optimal LLR for decoding and verification. Proceedings of ICASSP '99, IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, 15–19 March, 1999.
[Google Scholar ↗](#)
- [Lleida and Rose, 2000](#) E. Lleida, R.C. Rose
Utterance verification in continuous speech recognition: Decoding and training procedures

IEEE Trans. Speech Audio Process., 8 (2) (2000)

[Google Scholar](#) ↗

[Park et al., 2002](#) Park, H.S. et al, 2002. Temporal and spectral characteristics of Korean phonation types. Doctor of philosophy degree thesis, The University of Texas at Austin, August.

[Google Scholar](#) ↗

[Rahim et al., 1995](#) Rahim, M.G. et al, 1995. Robust utterance verification for connected digits recognition. In: ICASSP-95, International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, 9–12 May, 1995.

[Google Scholar](#) ↗

[Rahim et al., 1997](#) M.G Rahim, *et al.*

Discriminative utterance verification for connected digits recognition

IEEE Trans. Speech Audio Process., 5 (3) (1997)

[Google Scholar](#) ↗

[Sankar and Wu, 2003](#) Sankar, A., Wu, S.-L., 2003. Utterance verification based on statistics of phone-level confidence scores. In: Proceedings of (ICASSP '03), IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, 6–10 April, 2003.

[Google Scholar](#) ↗

[Sukkar and Lee, 1996](#) R.A. Sukkar, C.H. Lee

Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition

IEEE Trans. Speech Audio Process., 4 (6) (1996)

[Google Scholar](#) ↗

[Tomoko and Furui, 1995](#) M. Tomoko, S. Furui

Likelihood normalization for speaker verification using a phoneme- and speaker-independent model

Speech Comm., 17 (1995), pp. 109-116

[Google Scholar](#) ↗

[Wessel et al., 2000](#) Wessel, F.K., et al., 2000. A comparison of word graph and N-best list based confidence measures. In: Proceedings of ICASSP-2000, pp. 1587–1590.

[Google Scholar](#) ↗

[Wessel et al., 2001](#) F. Wessel, *et al.*

Confidence measures for large vocabulary continuous speech recognition

IEEE Trans. Speech Audio Process., 9 (March) (2001)

[Google Scholar](#) ↗

[William et al., 1997](#) J. William, *et al.*

The Handbook of Phonetic Sciences

Blackwell publishers Ltd (1997)

[Google Scholar ↗](#)

[Wu et al., 2000](#) C.-H. Wu, *et al.*

Integration of phonetic and prosodic information for robust utterance verification

IEE Proc. Vision Image Signal Process., 147 (1) (2000)

[Google Scholar ↗](#)

[Xiang and Berger, 2003](#) B. Xiang, T. Berger

Efficient text-independent speaker verification with structural Gaussian mixture models and neural network

IEEE Trans. Speech Audio Process., 11 (5) (2003), pp. 447-456

[View in Scopus ↗](#) [Google Scholar ↗](#)

[Xin and Wang, 2001](#) Xin, L., Wang, B.X., 2001. Utterance verification for spontaneous Mandarin speech keyword spotting. In: Proceedings of ICII 2001—Beijing. 2001 International Conferences on Info-tech and Info-net, Vol. 3, 29 October–1 November, 2001.

[Google Scholar ↗](#)

Cited by (3)

[Command-based voice teleoperation of a mobile robot via a human-robot interface ↗](#)

2015, Robotica

[Controlling a prosthetic arm with a throat microphone ↗](#)

2007, Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings

[Speaker identification and verification for intelligent service robots ↗](#)

2007, Proceedings of the 2007 International Conference on Artificial Intelligence, ICAI 2007

[View Abstract](#)

Copyright © 2005 Elsevier B.V. All rights reserved.



All content on this site: Copyright © 2025 Elsevier B.V., its licensors, and contributors. All rights are reserved, including those for text and data mining, AI training, and similar technologies. For all open access content, the Creative Commons licensing terms apply.



Competing models-based text-prompted speaker independent verification algorithm

Heungkyu Lee ^a, Hanseok Ko ^{b,*}

^a Department of Visual Information Processing, Korea University, Seoul, South Korea

^b Department of Electronics and Computer Engineering, Korea University, 5ka 1 Anamdong, Sungbuk-gu, Seoul 136-701, South Korea

Received 8 September 2004; received in revised form 7 February 2005; accepted 18 May 2005

Abstract

In this paper, we propose competing models based on text-prompted speaker independent verification algorithm for an intelligent surveillance guard robot, wherein a robot prompts a code (i.e. word or phrase) for user entrance authentication. The proposed system requires the text-prompted speaker independent verification. In addition, it does not require a speaker dependent model and extra trained model as an alternative hypothesis for log-likelihood ratio test because of memory limitation. This is due to the given application scenario that an administrator changes the voice code every day for security reasoning and the targeting domain is unlimited. To resolve these issues, we propose to exploit the sub-word based anti-models for log-likelihood normalization through reusing an acoustic model and competing with voice code model. Anti-models using automatic production rules are set up in an initial time by using the statistical distance of phonemes against a voice code. The proposed system uses a two-pass strategy using a SCHMM-based recognition and verification step. In addition, a harmonics-based spectral subtraction algorithm is applied for a noisy robustness on an outdoor environment. The performance evaluation is done by using a common Korean database, PBW452DB, which consists of 63,280 utterances of 452 isolated words recorded in silent environment.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Utterance verification; LRT; Voice code verification

1. Introduction

For surveillance tasks, much manpower is expended on sentry duty to guard the premises against unauthorized personnel for 24 h. To lessen the time and effort expended by human guard, an intelligent surveillance guard robot is desirable. The surveillance guard robot takes the role of

* Corresponding author. Tel.: +82 2 3290 3239; fax: +82 2 3291 2450.

E-mail addresses: hklee@ispl.korea.ac.kr (H. Lee), hsko@korea.ac.kr (H. Ko).

detecting and authorizing persons entering the perimeter of the secured area as well as passing status warnings. This system includes detection, recognition and tracking by using multiple sensors such as stereo cameras, IR cameras and array microphones. Under such an environment, a robot prompts for a code (i.e. word or phrase) for user entrance authentication. In the application scenario, an administrator changes a voice code every day for security reasons and the targeting domain is unlimited. Thus, a person authorized to enter a specific region should know the voice code previously before doing so. First, the proposed system speaks the password. Then, the authorized person responds with the corresponding voice code word. Such a proposed system is classified as text-prompted. In addition, as the target domain is unlimited and a person cannot register his identity information to the proposed system, the system cannot use the speaker dependent model. Because of this limitation, the proposed system utilizes the general acoustic model that is a speaker independent. As a result, the voice code verification system requires the text-prompted speaker independent verification (Sukkar and Lee, 1996). In addition, it does not require a speaker registration procedure to produce a speaker dependent model or an extra trained model such as a filler or garbage model for an alternative hypothesis in a log-likelihood ratio test (LRT). Thus, the proposed system is text-prompted speaker independent verification system. In this paper, the proposed system leverages on the combination of speech recognition and code verification simultaneously. While the purpose of this paper is to authenticate the user's identity, it employs the concept of code verification as opposed to the speaker verification, which analyzes the unique quality of spectral content. Therefore, it is clearly not a speaker dependent system.

This paper is motivated by the task where the system does not need to know the speaker and has only to verify whether the uttered voice code is correct or not in a specific area. To cope with this task, we consider the utterance verification approach. Over the past several years, much research has been done in this field (Jiang and Lee, 2003; Rahim et al., 1997; Lleida and Rose, 2000; Wu et al., 2000). However, such a trial has not been

undertaken yet. The utterance verification techniques are applied to several dialog systems in order to provide a user-friendly service. In addition, for reliability measurement (Jiang and Lee, 2003), this technique is applied to reject non-speech noises, out-of-vocabulary words, and potential recognition mistakes. Mostly, the confidence measure (CM) (Xin and Wang, 2001; Sankar and Wu, 2003; Rahim et al., 1995; Wessel et al., 2001) for this task is used to verify the uttered observation sequences after or during calculating the probability of a word W being recognized by an ASR system. Besides the utterance verification, a filler model or garbage model can be used for these purposes. Most algorithms require the extra model trained for a garbage model or anti-model. However, the limited memory size of our proposed embedded system prevents the algorithm from using and storing the extra alternative hypothesis model. Thus, the methods that do not require the extra trained model, and can reuse the acoustic model are investigated for the voice code utterance verification. Generally, a log-likelihood ratio test is applied to verify the utterance in this field of utterance verification where the verification step requires the alternative model for doing this task. To manage this problem, anti-models that are reusable from an acoustic model and can compete with a voice code model are considered.

For the general purpose of speech recognition, acoustic models are modeled to represent general characteristics of speech using HMM. The acoustic models are trained using speech databases of statistically significant number of people. Thus, the speech recognition system can recognize a spoken utterance without having speaker dependency. Thus, the acoustic models are speaker independent model. At this point, acoustic models can be modeled as word unit or phoneme unit. When it is modeled as a phoneme unit and independently, the acoustic models are context independent monophone. However, if we take into consideration both the left and the right neighboring phones, the acoustic models are context dependent models because a phoneme affects the utterance change. This is triphone, which is the model used in the proposed system. Such a phoneme model is modeled using HMM having states. In our system,

we modeled it using three states, left–right HMM, which is then arranged in triphone models. Our proposed system uses a two-pass strategy using a SCHMM-based recognition (Huang et al., 2001) and verification step as in Fig. 1. In the first pass, recognition is performed via a conventional Viterbi beam search algorithm that segments the test utterance into the N -best strings of a phoneme hypotheses (Wessel et al., 2000). In the second pass, voice code verification is performed. It computes a confidence measure that determines whether or not to reject the recognized voice code. The voice code verification technique is formulated as a statistical hypothesis test where the task is to test the null hypothesis that is a claimed voice code against the alternative hypothesis. Generally, the alternative hypothesis model as an anti-model is a previously trained model consisting of data that are mis-recognized or out-of-vocabulary generating false alarms, and noise tokens that also generates false alarms. However, the embedded hardware system requires limited and inexpensive memory. Thus, we reuse the general acoustic model as an anti-model to compete with the claimed voice code model. This is derived from the fact that the conventional speech recognition algorithms as a pattern classification problem use the maximum a posteriori (MAP) decision rule to find the most likely sequence of words. Respective phonemes can be modeled using HMM having states. When an uttered speech vectors are entered

into decoding stage, likelihood scores are computed at all sequences of words until the end-point of speech is found. Thus, a likelihood score of phoneme models similar with uttered speech features become increased, or else a likelihood score has low value. When a person speaks a word, spoken word can be a completely different word or a similar word with a voice code word. The conventional speech recognition system obtains recognition success (false acceptance) when a person speaks a similar word because it uses the maximum a posteriori decision rule to find the most likely sequence of words. However, such a problem gives the high risk in a user authentication system. Thus, the anti-models to cope with this problem are required. Thus, in this paper, we propose the anti-models that are specific against a given voice code word model by using a regular production rules. The anti-models are automatically constructed using a grapheme to phoneme converter through the analysis of voice code text. The anti-models are based on phoneme and syllable units against the claimed voice code using the statistical distance of phonemes. In our system, the role of the anti-models is very important to verify an utterance. These anti-models compete with voice code model in the first pass, and are used as a normalization model for LRT in the second pass. Thus, we propose the voice code verification technique using the anti-models that are statistically opposed to the each phoneme of the voice code.

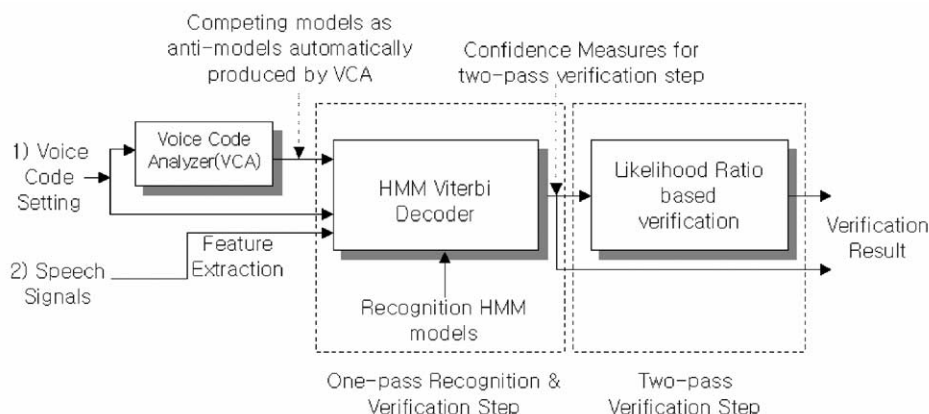


Fig. 1. Block-diagram of voice code verification.

This paper is organized as follows. In Section 2, we describe the voice code verification method using sub-word based anti-models. In Section 3, we describe the harmonics-based spectral subtraction algorithm. Then, we conduct the representative experiments and discuss the results on performance of the proposed methods in Section 4. Finally, the concluding remarks are presented in Section 5.

2. Voice code verification

2.1. Competing models as anti-models

The a posteriori probability used for the likelihood normalization method in text-prompted speaker verification (Xiang and Berger, 2003; Tomoko and Furui, 1995) is given by

$$p(S_c, W_c/O) = \frac{p(O/S_c, W_c)p(S_c, W_c)}{\sum_i \sum_j \{p(O/S_i, W_j)p(S_i, W_j)\}} \approx \frac{p(O/S_c, W_c)}{\sum_i \sum_j p(O/S_i, W_j)} \quad (1)$$

where S_i is a speaker and S_c is the claimed speaker. W_i is a text, and W_c is the prompted text, $p(S_i, W_j)$ is the simultaneous probability for speaker i and text j . $p(O/S_c, W_c)$ is the probability of the claimed speaker's HMM corresponding to the prompted text. In the voice code verification, S_c and S_i can be ignored because this is speaker independent verification. Thus, Eq. (1) can be simplified as

$$p(W_c/O) \approx \frac{p(O/W_c)}{\sum_j p(O/W_j)} \quad (2)$$

This is the same with the likelihood normalization method for an utterance verification in a conventional ASR algorithm. In Eq. (2), W_c becomes the uttered word sequence, and $p(O/W_j)$ is approximated by the summation of the n highest likelihoods by using parallel phoneme HMM networks for all registered words. As a result, if the speaker information is ignored, the text-prompted verification technique is equal to the conventional ASR algorithms as a pattern classification problem using the maximum a posteriori (MAP) decision rule to find the most likely sequence of words as follows:

$$W_k = \arg \max_j L(O/W_j) \quad (3)$$

where $L(O/W_j)$ is the likelihood of an observation sequence O , given word W_j . In a text-prompted verification, this is the time that the number of a given word, j is equal to one. At this time, when someone speaks a false word for a text-prompted verification, we cannot verify the uttered word because we have no normalized models to test a likelihood score. Thus, we need the models to increase the likelihood score more than the one of the claimed voice code model when someone speaks the false voice code. However, we do not want the previously trained models (filler or garbage model) for likelihood normalization because the memory of our system is limited. The general filler-based methods, which try to calculate the probability of observing an utterance sequence, are previously trained. The utterance verification approach (Xiang and Berger, 2003; Tomoko and Furui, 1995; Benitez et al., 2000), mostly motivated by speaker verification (Ahn et al., 2002), also requires a trained model for likelihood normalization in LRT to verify the claimed content of a spoken utterance. In this case, methods that use just some phonemes such as cohort modeling or train the anti-models by using the speech interval that results in a recognition failure can be applied. Otherwise, someone may be able to select an overall phoneme set as an anti-model except that phoneme by itself. However, too much time is required to calculate the log-likelihood of the overall phoneme sets. To cope with these problems, we reuse the original acoustic model for the alternative hypothesis model. Alternative hypothesis models as anti-models can be made automatically through the analysis of phoneme information with respect to the prompted text word. In this paper, we propose the construction method of anti-models using the statistical distance of phonemes against the voice code. This reused anti-model can be used to compete with the prompted voice code as follows:

$$W_k = \arg \max_j L(O/W_0, \overline{W}_1, \dots, \overline{W}_j) \quad (4)$$

where W_0 is a prompted voice code, and $\overline{W}$ is a competing model to be used for likelihood

normalization. W_k is a concatenation of syllable units that can be written as

$$W_k = S_1^k S_2^k \cdots S_N^k \quad (5)$$

where N is the number of syllables. In addition, a syllabic unit is a concatenation of context independent phoneme units that can be written as

$$S_N^k = P_{N,1}^k P_{N,2}^k \cdots P_{N,M}^k \quad (6)$$

where M is the number of phonemes. Then, this context independent phoneme unit is changed into a context dependent phoneme unit after anti-phoneme units are constructed. Ultimately, the anti-phoneme units become the context dependent model.

As seen in Eq. (4), in the first pass, the Viterbi algorithm is employed to find the most likelihood word W_k . In this step, the prompted voice code is first verified as follows:

$$PVC = \begin{cases} \text{true} & \text{if } j = 0 \\ \text{false} & \text{else} \end{cases} \quad (7)$$

If the verification result PVC is true, the second pass to test a likelihood score is followed.

In this sub-section, we describe the automatic construction method of anti-models that opposes the statistical distance according to the manner and place of articulation, and tongue advancement and aperture (Park et al., 2002; William et al., 1997). When a voice code is presented to the system, the proposed system produces the anti-models automatically against a given voice code. If an arbitrary phoneme sequence or a free-phoneme sequence as an anti-phoneme sequence is used, the difference of likelihood scores depends on the phoneme sequence according to a given voice code during the Viterbi decoding. In the case of similar likelihood scores between models, out of vocabulary rejection can occur at the stage of confidence measure. In addition, when a person speaks a similar word, the false acceptance rate can increase. To resolve this problem, accurate modeling of anti-models against a given voice code is required. In this paper, we propose anti-models that can be produced automatically using regular rules, and can compete with a voice code model in order to reject both similar and completely different words.

At first, the prompted voice text is automatically changed into a phoneme string, produced

using a grapheme to phoneme (G2P) converter through text analysis. Then, the following rules for construction of anti-models are applied.

The voice code can be composed of a concatenation of a syllable, S , that is the set of phonemes. A voice code, W_0 is expressed by

$$W_0 = \{S_1, S_2, \dots, S_N\} \quad (8)$$

where N is the total number of syllables of a given voice code. At first, when a person produces a similar word, this may result in verification success. This occurs when any person says the word as follows:

$$\begin{aligned} \overline{W}_1^1 &= \{\overline{S}_1, S_2, \dots, S_N\}, \\ \overline{W}_2^1 &= \{S_1, \overline{S}_2, \dots, S_N\}, \dots, \\ \overline{W}_N^1 &= \{S_1, S_2, \dots, \overline{S}_N\} \end{aligned} \quad (9)$$

where N is the number of anti-syllable models for the first method and the variable, $\overline{S}$ is the anti-syllable. This sometimes results in verification success. To prevent false acceptance, we can use Eq. (9) as anti-models that compete with the voice code model when a person says a similar password. The anti-syllable model can be constructed using a concatenation of an anti-phoneme against each syllable unit as

$$\overline{S}_N = \{\overline{P}_1, \overline{P}_2, \dots, \overline{P}_M\} \quad (10)$$

The criterion to select the anti-phoneme is classified into two categories. Korean phonemes can be explained using a group according to the manner and place of articulation, and tongue advancement and aperture as shown in Table 1. By using this characteristic, anti-models can be produced using the one-to-many mapping method. Otherwise, anti-models can be produced using the one-to-one mapping method as shown in Table 2. This one-to-one matching method is produced manually considering the prosody generation characteristics of Korean phonemes. The first approach is to use a grouping method using the statistical distance of phonemes as in Table 1. The second approach involves the method to classify phonemes according to the manner and place of articulation, and tongue advancement and aperture as in Table 2, which is matched in order between phoneme and anti-phoneme. Table 3 depicts the matched phoneme set between Korean and English for

Phoneme grouping for construction of anti-phoneme model using statistical distance of phonemes

	Group	Phoneme	Group number	Anti-phoneme group number
Consonant	Back tooth	ㄱ ㅋ ㆁ ㆍ ㆏	1	4
	Palate I	ㄷ ㅌ ㄴ ㄹ ㄺ ㄻ ㄼ	2	4
	Palate II	ㄴ ㄹ	3	1
	Lips	ㅍ ㅂ ㅃ ㅍ	4	1
Vowel	ㅏ series	ㅏ ㅑ ㅓ	5	6
	ㅓ series	ㅓ ㅕ ㅗ	6	5
	ㅗ series	ㅗ ㅛ ㅜ	7	5
	ㅗㅓ series	ㅗㅓ ㅛㅓ ㅜㅓ ㅗㅓ ㅛㅓ ㅜㅓ	8	9
	ㅗㅓ series	ㅗㅓ ㅛㅓ ㅜㅓ ㅗㅓ ㅛㅓ ㅜㅓ	9	8

Anti-phoneme model production rules using statistical distance of phonemes

Phoneme to Anti-phoneme			
Consonant	Phoneme	ㄱ ㅋ ㆁ ㄷ ㄸ ㄴ ㄹ ㄺ ㄻ ㄼ ㄽ ㄾ ㄿ ㅁ ㅂ ㅃ ㅅ ㅆ ㅈ ㅊ ㅊ ㅌ ㅍ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅟ	Manner and place of articulation
	Anti-phoneme	ㅅ ㅆ ㅆ ㅈ ㅊ ㅊ ㅌ ㅍ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅟ	
Vowel	Phoneme	ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅟ ㅐ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅟ	Tongue advancement and aperture
	Anti-phoneme	ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅟ ㅐ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅟ	

To produce the anti-model for each syllable, the corresponding syllable in the prompted voice code is changed into an anti-syllable using the anti-phoneme according to [Table 1](#) or [Table 2](#) after the text is changed into the phoneme list using a G2P converter, where it needs a parsing process to find each syllable that is composed of a consonant and vowel. In the Korean language, a syllable can be composed of “V”, “C + V”, “C + V + C”

Table 3
Matching table between Korean and English phonemes

Consonant						Vowel					
ㄱ	g	ㅋ	kh	ㆁ	ch	ㅏ	a	ㅑ	yo	ㅓ	yv
ㄴ	n	ㄷ	th	ㅓ	j	ㅕ	v	ㅠ	yu	ㅗ	ya
ㄷ	d	ㅌ	ph	ㅕ	jj	ㅛ	o	ㅜ	yae	ㅜ	we
ㄹ	r	ㅎ	h	ㅇ	ng	ㅜ	u	ㅡ	ye	ㅡ	eui
ㅁ	m	ㅈ	gg	ㅈ	ss	ㅡ	eu	ㅗ	wa	ㅗ	wv
ㅂ	b	ㅊ	dd			ㅣ	i	ㅗ	we	ㅗ	we
ㅅ	s	ㅍ	bb			ㅗ	e	ㅗ	wi	ㅗ	e

and “V + C” where V is the vowel and C is the consonant. A Korean syllable can be classified into nine groups as illustrated in Table 4. Using this rule, a given text is classified into the syllable lists.

The second, when any person utters a similar word that includes all parts of a prompted voice code, verification success often results. This is particularly true when a false text is uttered as follows:

$$\begin{aligned}
 \overline{W}_1^2 &= \{S_1, S_2, \dots, S_N, \overline{S_{N+1}}\}, \\
 \overline{W}_2^2 &= \{S_1, S_2, \dots, S_N, \overline{S_{N+1}}, \overline{S_{N+2}}\}, \dots, \\
 \overline{W}_M^2 &= \{S_1, S_2, \dots, S_N, \overline{S_{N+1}}, \overline{S_{N+2}}, \dots, \overline{S_{N+M}}\}
 \end{aligned} \quad (11)$$

where M is the number of anti-syllable models competing with a given voice code model, and anti-syllable, $\overline{S_{N+M}}$ is matched to its syllable, S_N . To prevent this case, we use Eq. (11) as anti-models. The anti-syllable model also can be constructed using Table 1 or Table 2.

The third, when any person says a similar word that is some part of the password text, this also often results in verification success. This happens when any person says a text as follows:

$$\begin{aligned}
 \overline{W}_1^3 &= \{S_1\}, \\
 \overline{W}_2^3 &= \{S_1, S_2\}, \dots, \\
 \overline{W}_{N-1}^3 &= \{S_1, S_2, \dots, S_{N-1}\}
 \end{aligned} \quad (12)$$

where $N - 1$ is the number of anti-syllable models. To prevent this case, we use Eq. (12) as anti-models. In addition, we can use anti-models contrary to Eq. (12) as follows:

$$\begin{aligned}
 \overline{W}_1^4 &= \{\overline{S_1}\}, \\
 \overline{W}_2^4 &= \{\overline{S_1}, \overline{S_2}\}, \dots, \\
 \overline{W}_{N-1}^4 &= \{\overline{S_1}, \overline{S_2}, \dots, \overline{S_{N-1}}\}
 \end{aligned} \quad (13)$$

Finally, the following anti-model is applied:

$$\overline{W}_1^5 = \{\overline{S_1}, \overline{S_2}, \dots, \overline{S_N}\} \quad (14)$$

In the proposed anti-model equations for automatic production, Eqs. (9), (11) and (12) represent anti-models to reject similar words with a voice code. In addition, Eqs. (12) and (13) represent anti-models to reject completely different words. After these anti-models are constructed through the analysis of a given voice code, all anti-models are used to compete with a voice code model. These models would increase the likelihood score of anti-models while the likelihood score is reduced when someone speaks a false word or phrase.

2.2. Voice code verification using sub-word based anti-models

In the second pass, the voice code verification task is applied. Generally, a sub-word based

Table 4
Korean syllable production rules

Syllable	Word rules	Group	No	Comment
CV	CV/CV	CV/CV	1	
	CV/CVC	(PART1)		
	CV/VC	CV/V	2	
	CV/V	(PART2)		
CVC	CVC/CV	CVC/V	3	
	CVC/CVC	(PART3)		
	CVC/VC	CVC/V	1	Follows rule part 1 according to Korean utterance rules
	CVC/V	(PART4)		
VC	VC/CV	VC/C	4	
	VC/CVC	(PART5)		
	VC/VC	VC/V	5	Follows rule part 7 according to Korean utterance rules
	VC/V	(PART6)		
V	V/CV	V/CV	5	
	V/CVC	(PART7)		
	V/VC	V/V	6	
	V/V	(PART8)		

utterance verification or out-of-vocabulary rejection method is based on a likelihood ratio test. The major difficulty with an LRT in utterance verification is modeling the alternative hypothesis, where the true distribution of the data is unknown and an alternative hypothesis usually represents a very complex and composite event. Given a decoded sub-word in an observed segment, we need a decision rule by which we assign the sub-word to either hypothesis H_0 or H_1 . For the binary testing problem, one of the most useful tests for decision is the Neyman–Pearson Lemma. For a given number of observations, which minimizes the error for one class while maintaining the error for the other class constant, the likelihood ratio test is as follows:

$$\text{LRT}(X) = \frac{P(X/H_0)}{P(X/H_1)} = \frac{P(O_n/\lambda_n)}{P(O_n/\bar{\lambda}_n)} \geq \eta \quad (15)$$

where H_0 means that the hypothesis is true and H_1 means that the hypothesis is false, λ is the sub-word model, $\bar{\lambda}$ is the anti-sub-word model, and X is the uttered input observation that the number of sub-words is N . Hence, Eq. (15) accepts the hypothesis when the likelihood ratio is greater than the given threshold, η . If the alignment of observations with respect to the sub-word unit is obtained, the LRT is performed using log-likelihood of a sub-word model.

The sub-word alignment and log-likelihood score are obtained on a log domain through the Viterbi segmentation. For the likelihood ratio normalization, an average frame log-likelihood ratio (LLR), R_n is defined as

$$R_n = \frac{1}{l_n} [\log P(O_n/\lambda_n) - \log(O_n/\bar{\lambda}_n)] \quad (16)$$

where $\bar{\lambda}$ is an anti-model automatically produced by the rule as in the previous sub-section and $\log(O_n/\bar{\lambda}_n)$ is the likelihood score approximated by the summation and averaging of the $nBest$ highest likelihood scores using the parallel phoneme HMM networks by competing with each other. Thus, Eq. (16) can be written as

$$R_n = \frac{1}{l_n} \left[\log P(O_n/\lambda_n) - \frac{1}{nBest} \sum_{m=1}^{nBest} \log(O_n/\bar{\lambda}_m) \right] \quad (17)$$

In Eq. (17), the dynamic range of a sub-word based log-likelihood ratio is high. This can affect the overall performance. One way to limit the dynamic range of the sub-word confidence measure is to use a sigmoid function of the following form [2]:

$$U_n = \frac{1}{1 + \exp(-\alpha \times (R_n - \tau))} \quad (18)$$

where τ and α are location and weighting parameters. The log confidence score has a slope of α when the log-likelihood score is less than zero. The τ reduces the dynamic range caused by frames having a high likelihood ratio. Thus, we can control the effect of the log-likelihood score with τ and α .

2.3. Confidence measure

For effective voice code verification, we need to define a function to combine the results of sub-word tests. The confidence measure (CM) for an input utterance O can be represented as

$$CM(O) = f(CM_1, CM_2, \dots, CM_N) \quad (19)$$

where $f()$ is the function to combine the verification scores. This is defined as a function of their likelihood ratios. It can be considered as a joint statistic for overall word-level verification.

The first confidence measure is based on frame-duration normalization, which is defined as follows:

$$CM_1 = \frac{1}{L} \sum_n^N (l_n * R_n) \quad (20)$$

where N is the total number of sub-words in the utterance, and L is the total number of utterance frames $L = \sum_{n=1}^N l_n$. The second one is based on syllable segment-based normalization. It is a simple average of log-likelihood of all the syllables:

$$CM_2 = \frac{1}{N} \sum_n^N R_n \quad (21)$$

$$CM_3 = \exp \left(\frac{1}{N} \sum_{n=1}^N \log R_n \right) \quad (22)$$

$$CM_4 = \frac{1}{N} \sum_{n=1}^N U_n \quad (23)$$

$$CM_5 = \exp \left(\frac{1}{N} \sum_{n=1}^N \log U_n \right) \quad (24)$$

where Eqs. (21) and (22) are the arithmetic and geometric means of the unweighted sub-word level confidence scores, and Eqs. (23) and (24) are the arithmetic and geometric means of the sigmoid weighted sub-word score.

For every confidence measure, a specific threshold is set up. If its value is below the threshold, the candidate is discarded from the verification task, resulting in a voice code verification failure.

3. Harmonics-based spectral subtraction

3.1. Speech-dominant region vs. noise-dominant region

In speech, voiced speech segment has peaks positioned periodically in spectrum due to the vibration of vocal cords. These points are very critical in speech sound perception. Fig. 2 illustrates this phenomenon with a sample spectrum of a speech frame capturing the pronunciation /oh/ in one utterance contained in Aurora 2 corpus. Note that at the peaks, their amplitudes are far greater than the amplitudes at the points between the adjacent small peaks, or side-lobes. Also, it is observed that the degree of corruption by the noise in the peaks is less than the degree of corruption at the points over the side-lobes. From this observation,

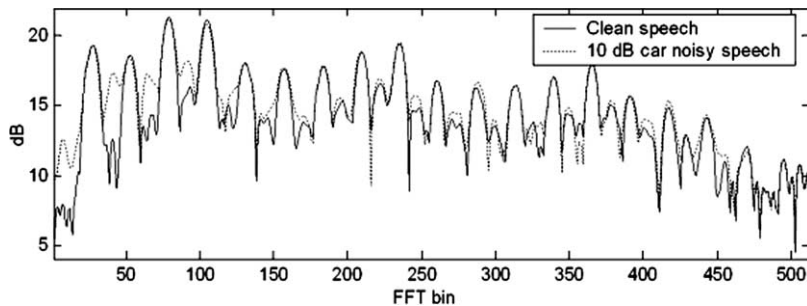


Fig. 2. Example spectrum of a speech frame (pronunciation /oh/ by female).

it can be deduced that in speech spectrum, the speech-dominant regions exist over or near the peaks and the noise-dominant regions exist over or near the side-lobes. However, the illustration as displayed could be misleading. It does not show, for example, that the speech can be more corrupted in some frequency regions.

3.2. Peak point detection and segmentation

It is known that the spectral harmonics are located at the points of multiples of fundamental frequency. First, using an autocorrelation function, the indices of local maxima in autocorrelation values are found. Among those indices, we find the fundamental frequency f_0 by taking the reciprocal of the index that represents the maximum value. Second, by applying the nonlinear smoothing method, f_0 is modified to more accurately reflect the true fundamental frequency. This algorithm works well even in a high degree noisy environment. The autocorrelation function is expressed as follows:

$$\phi(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)x(n+\tau) \quad (25)$$

We then establish the index as k_0 for the frequency bin that corresponds to f_0 . Using k_0 , the peak points (harmonics) h_l are determined as $h_1 = k_0$, $h_2 = 2k_0$, $h_3 = 3k_0$, ..., $h_L = Lk_0$, where L is the index of the last component of harmonics. Using this procedure, we determine all harmonics in the spectrum of an input frame, which is assumed to be a speech-dominant region. We illustrate the validity of the proposed method with spectrograms and the detection of peak point. For the purpose of implementing the proposed scheme, the frequency axis is divided into several non-overlapping bands in the following form:

$$\begin{aligned} &[1, h_1], \left[h_1, h_1 + \frac{k_0}{2} \right], \left[h_1 + \frac{k_0}{2}, h_2 \right], \\ &\dots, \left[h_L, \frac{\text{FFTorder}}{2} + 1 \right] \end{aligned} \quad (26)$$

3.3. Voice activity detection and noise estimation

Note that in each input frame, whether speech frame or not, the fundamental frequency is calculated. As a result, the value of autocorrelation is available at each frame. To begin, we calculate the ratio of this value over the frame energy since this ratio appears to be a more effective measure to distinguish a speech region from a non-speech region than the mundane energy measure. Then, two separate procedures are employed for robust detection of speech vs. non-speech regions. First, a smoothing procedure is carried out. If this value is greater than the threshold 1.4th_{t-1} as defined by the following expressions, the relevant frame is decided as a ‘speech frame’.

If $\frac{\phi(\tau')}{E_t} > 1.4\text{th}_{t-1}$ then ‘speech frame’ and

$$\text{th}_t = 0.95\text{th}_{t-1} + 0.05 \frac{\phi(\tau')}{E_t} \quad (27)$$

else ‘non-speech frame’ and

$$\text{th}_t = \text{th}_{t-1} \quad (28)$$

Note that th_t indicates the threshold of the t th frame, E_t indicates the t th frame energy and $\phi(\tau')$ is the autocorrelation value corresponding to the f_0 of the relevant frame.

Secondly, for frames determined as ‘speech frame’, if the detected fundamental frequency is not in the 50–600 Hz, then it is regarded as ‘non-speech frame’ once again. For the noise estimation, we assume that any starting input speech is followed by a silence or background noise segment corresponding to 10 frames (about 100 ms). If the input frame is determined as a ‘non-speech frame’, the estimated noise spectral magnitude is updated as follows:

$$\begin{aligned} |\hat{B}_t(k)|^2 &= \lambda_B |\hat{B}_{t-1}(k)|^2 + (1 - \lambda_B) |Y_t(k)|^2, \\ 0.65 &\leq \lambda_B \leq 0.998 \end{aligned} \quad (29)$$

where t is the frame index, k is the index of FFT bins and $\lambda_B = 0.95$ is used.

3.4. Spectral subtraction rules

Spectral subtraction rules are applied according to whether the frame is speech or not. In speech region, the following linear function is applied:

If $k \in [1, h_l - 1]$ or $k \in [h_{l-1} + \frac{k_0}{2}, h_l - 1]$, then

$$\gamma(k) = \frac{\alpha_{\min} - \alpha_{\max}}{h_l - (h_{l-1} + \frac{k_0}{2})} (h_l - k) + \alpha_{\max} \quad (30)$$

$$\delta(k) = \frac{\beta_{\max} - \beta_{\min}}{h_l - (h_{l-1} + \frac{k_0}{2})} (h_l - k) + \beta_{\min} \quad (31)$$

If $k \in [h_l, h_l + \frac{k_0}{2}]$ or $k \in [h_L, \frac{\text{FFTOorder}}{2} + 1]$, then

$$\gamma(k) = \frac{\alpha_{\max} - \alpha_{\min}}{(h_l + \frac{k_0}{2}) - h_l} (h_l - k) + \alpha_{\min} \quad (32)$$

$$\delta(k) = \frac{\beta_{\max} - \beta_{\min}}{(h_l + \frac{k_0}{2}) - h_l} (h_l - k) + \beta_{\max} \quad (33)$$

where $l = 1, 2, \dots, L$ and L is the index of the last component of harmonics.

$\gamma(k)$ applies the maximum over-subtraction factor to the midpoint at each component. Then, the over-subtraction factors of points existing in the interval between those points are interpolated linearly. In contrast, $\delta(k)$ applies the maximum floor factor to each harmonic component and the minimum floor factor to the midpoint at each component (Beh and Ko, 2003). Then, other floor factors between these points are generated by linear interpolation.

In this procedure, we set $\alpha_{\max} = 8$, $\alpha_{\min} = 1$, $\beta_{\max} = 0.15$ and $\beta_{\min} = 0.05$. Finally, the proposed subtraction rule can be represented as follows:

$$|\hat{X}(k)|^2 = \begin{cases} |Y(k)|^2 \left(1 - \gamma(k) \frac{|\hat{B}(k)|^2}{|Y(k)|^2} \right) & \text{if } |Y(k)|^2 - \gamma(k) > \delta(k) |Y(k)|^2 \\ \delta(k) |Y(k)|^2 & \text{else} \end{cases} \quad (34)$$

Meanwhile, in the non-speech region, we apply the subtraction rule in the same way as conventional methods with $\alpha_{\max}$ and $\beta_{\min}$.

4. Evaluation of the proposed system

4.1. Experimental condition

For speech input to verify the uttered voice code, the sampling rate is 16 bits PCM. In cases

of speech recognition and text-to-speech engine, sampling rate is the same. The isolated word decoding module with dynamic vocabularies to reduce computing time and memory size is optimized, and speech signals are analyzed within 125 ms frame with 10 ms lapped into 26th order feature vector that has 13th order MFCCs including log energy and their 1st derivatives. The training data set consists of about 120,000 utterances of 6000 isolated words sets recorded in an office environment. In addition, we used a different speech corpus for testing the data set, which is PBW452DB, Korean Common DB, which consists of 63,280 utterances of 452 isolated words recorded in a silent environment. For this recording, 70 men and women uttered one word twice. The microphones used in the training data collection and testing data collection are different. Among the 63,280 utterances in the test data, 140 utterances of one word are used as the voice code and the 140 utterances of any other word are used for the out-of-vocabulary data. Then, the false acceptance rate (FAR) and false rejection rate (FRR) of the 452 computed voice codes are averaged. The threshold is gradually increased as much as 0.1.

4.2. Experimental result

In Korean language, the consonant is produced according to the manner and place of articulation, and the vowel is produced according to the tongue advancement and aperture whenever a person utters a word. The proposed anti-phoneme model is based on the Korean phonetic theory. In this paper, competing models are considered to increase the likelihood scores against a score of the voice code model in order to reject either a word that is completely different or a similar word when a person utters a false code. In addition, there is a need for automatic production rules of anti-models because the voice code is changed every day. In this case, specific anti-model is desirable against a given voice code model since its performance may depend on the given voice code word.

In this paper, we apply an utterance verification technique using an N -best alternative hypothesis model for likelihood normalization in an LRT.

In our previous work (Kim and Ko, 2003) on utterance verification, several utterance verification methods were simulated. Our method, the Bayesian fusion technique showed a performance higher than any other method. However, we apply the 5-best technique that is easy to implement and has a low computing time on a DSP board.

For a voice code verification simulation, an anti-model making routine for likelihood normalization against a log-likelihood score of voice code is implemented as in Section 2. The simulation is done using the total anti-models as in Eqs. (9) and (11)–(14). At first, the two approaches using Tables 1 and 2 as described in Section 2.1 are compared. Four categories of methods are used for evaluation. Method 1 is the anti-model set using Eq. (14), method 2 is the anti-model set combining Eqs. (11) and (14), method 3 is the anti-model set combining Eqs. (11), (12) and (14), and method 4 is the anti-model set combining Eqs. (9), (11), (12) and (14). The simulation result shows that the second approach is a bit higher than the first approach. The detection probability of the second approach is 2% higher than the one of the first approach, and the FAR of the second approach is 1% lower than the one of the first approach as in Table 5. Thus, we choose the second approach as in Fig. 3.

As shown in Fig. 3(a), method 1 has a high FAR while FRR is low. In method 2 (Fig. 3(b)) and method 3 (Fig. 3(c)), the EER is 0.09 and 0.125 respectively. However, it cannot cope with various situations as described in Section 2. In addition, method 4 has no EER. Thus, we are not able to decide a stable threshold because methods 1 and 2 have no EER. As shown in Fig. 3(a)–

(d), respective anti-models cannot cope with either completely different or similar words, and it is difficult to decide a threshold even for methods 1 and 2.

Thus, we did extra simulations on the basis of method 2. Fig. 3(b) shows the result of method 2 when we use the variable, $M = 1$, of Eq. (11). When we apply the variable, $M = 3$, of Eq. (11) in method 2, FRR and FAR are improved. Fig. 4 shows this comparative result. Method 5 is the anti-model set that variable M of Eq. (11) is 3 in method 2. This result shows that anti-models having long sequences of anti-phonemes reduce the false acceptance rate considerably when a person speaks a word that has long false word sequence.

Using the result from Fig. 4, we combined the anti-model equations of method 2 with Eqs. (12) and (13). In Fig. 5, method 6 is the anti-model set combining Eq. (12). Method 7 is the anti-model set combining Eq. (13). Method 8 is the anti-model set combining Eqs. (12) and (13). Methods 7 and 8 showed similar results. However, method 8 is a bit improved and also can cope with various utterances of people very well.

Finally, we combine the anti-models using Eq. (9) with method 8 as shown in Fig. 6. The curve shape and result of the final method is similar with method 8 of Fig. 5. However, the final result is a bit improved over method 8 of Fig. 5. In addition, the EER is 0.08. This result is improved by 16% over the utterance verification result as shown in Table 6. Aimed at applying to utterance verification, the “Sub-word LRT”, “Phonetic filler”, “5-best”, “256 mix GMM” and “Bayesian fusion” employed an arbitrary phoneme sequence or a general free-phoneme sequence for comparative

Table 5

Comparison results between the first approach using “one-to-many mapping method” as in Table 1 and the second approach using “one-to-one mapping method” as in Table 2 (threshold = 0)

Method	First approach		Second approach	
	FRR	FAR	FRR	FAR
1	0.0088179	0.5827750	0.0087547	0.5759799
2	0.0171302	0.5166876	0.0168885	0.5069785
3	0.0567947	0.2383059	0.0561109	0.2262200
4	0.2288869	0.0987989	0.2053726	0.0840138

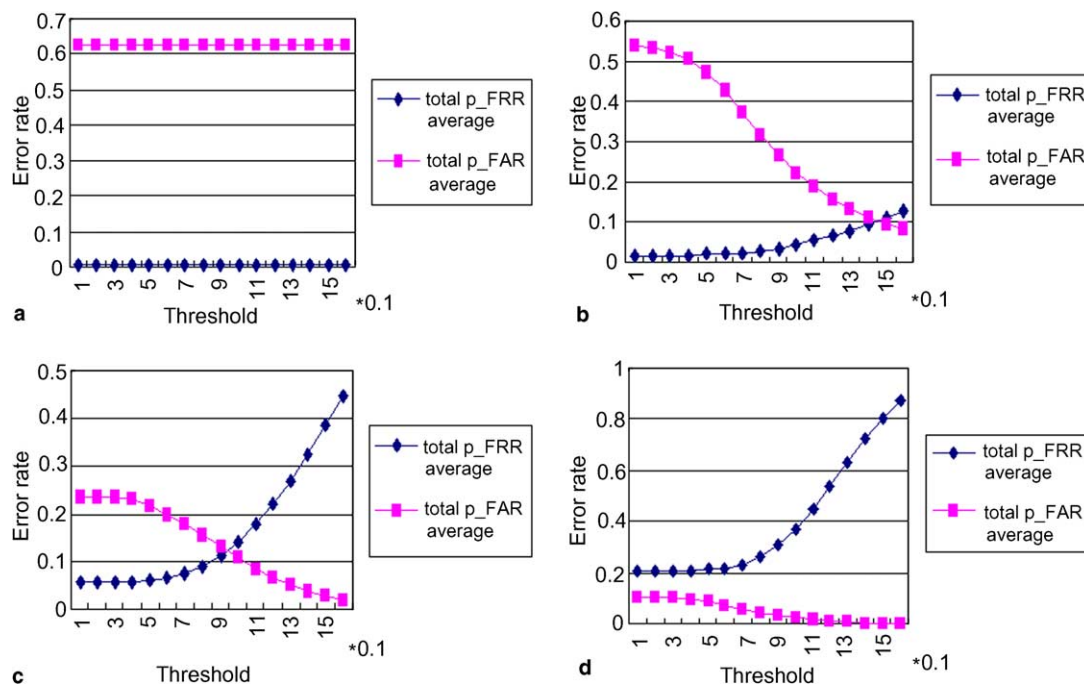


Fig. 3. Simulation results of the second approach using statistical distance of phonemes. Simulation results of (a) method 1, (b) method 2, (c) method 3 and (d) method 4.

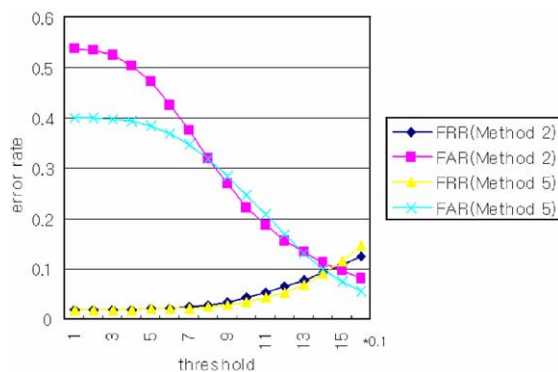


Fig. 4. Comparison result of methods 2 and 5.

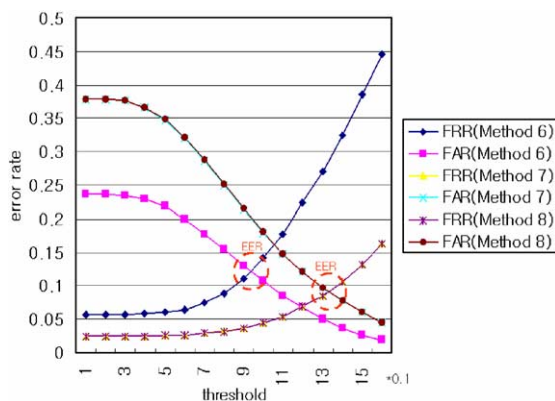


Fig. 5. Comparison result of the methods 6–8.

evaluation. The Sub-word LRT (Likelihood Ratio Test) is done at a sub-word unit (phoneme or syllables) during Viterbi decoding and the results are combined for final decision. The phonetic filler or 256 mix GMM uses an arbitrary phoneme sequence or previously trained models such as gar-

bage models that is trained using predictable words set. When a word network is constructed, these models are linked to this word network. So, during Viterbi decoding, uttered word (feature vectors) is checked whether the uttered word is within the given recognition list (constructed

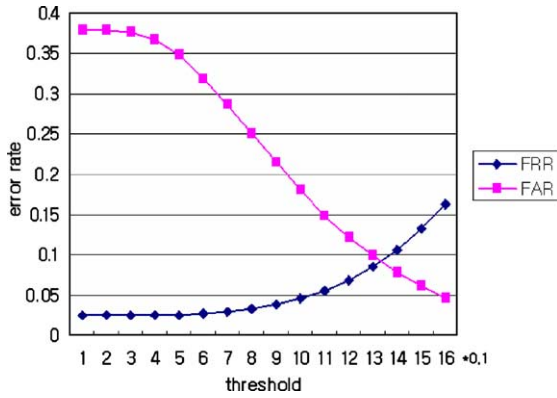


Fig. 6. Final result using all anti-models.

phoneme sequence) or not. In 5-best method for LRT, all path sequences computed in Viterbi decoding procedure are not used. N -best sequences having maximum likelihood scores are computed for LRT. In the case of Bayesian fusion method applying to utterance verification problem, conventional confidence measure modeling and evaluation methods can be treated as local sensors which detect or verify acoustic events concerned. Final decision is made in global fusion center, which combines the local decisions sent by each local utterance verifier. However, Bayesian fusion method considers parallel configuration for utterance verification for utterance verification where each sensor observes common phenomenon and makes local decision regarding it. Thus, the objective of Bayesian formulation is to minimize the Bayesian risk. It is to find optimal decision rules for each local sensor as well as a global one.

In this paper, we proposed automatic production rules of anti-models such that the performance would not depend on a code. In order to accomplish the code independence, anti-models are produced against the specific voice code. Eqs.

(9), (11) and (12) represent the anti-models to reject similar words. Eqs. (13) and (14) are anti-models to reject completely different words.

As shown in Fig. 3, specific equations for anti-models have low performance when respective equations are applied alone because they are able to cope only with the specific words. However, combined anti-models result in more robust performance because they can cope with various acoustic situations. If the modeling of anti-models is not precisely done and a few anti-models are applied, the performance should depend on a given voice code or uttered words. In contrast, if a lot of anti-models are applied and thus very precise modeling of anti-models are considered, some words can be rejected in the confidence measure stage because the likelihood score of anti-models that represent a similar word against a voice code word has increased. Therefore, an optimal number of anti-models should be applied to obtain a robust performance.

This system usually is utilized in an outdoor surveillance region, which requires a noise robust voice code verification to cope with not only environmental noise, but also other white noise. To resolve this problem, a harmonics-based spectral subtraction algorithm is applied for preprocessing the noise. First, an experiment is conducted by the Aurora 2 evaluation procedure under a continuous digits recognition tasks. Test sets are reproduced using TIDigits of which the entire speech data are down-sampled to 8 kHz and various realistic noises are added artificially. The feature vector order is 39 and is composed of 13 order static MFCC (c1–c12 + log energy) and its derivatives and accelerations. For comparison, a spectral subtraction algorithm and nonlinear algorithm are evaluated as in Table 7. As seen in Table 7, HSS shows that the proposed algorithm is more robust than other algorithms. A notable advantage of the

Table 6
Comparison results with other utterance verification methods (threshold = 1.2)

Threshold	Method					
	Sub-word LRT	Phonetic filler	5-best LRT	256 mix GMM	Bayesian fusion	Proposed method
1.2	0.1882	0.1448	0.2432	0.1862	0.1383	0.08

Table 7

Summary of overall word accuracies on Aurora2 mis-matched training/testing condition (%)

	Baseline	SS	HSS
Baseline	60.06	77.89	80.59
CMN	71.16	78.56	82.00

proposed scheme is that it does not require an exact SNR estimate under various noise conditions. If a noise characteristic is stationary and each magnitude of harmonics is not masked by a noise, the SS technique itself may work alone. However, real-life conditions are not all that cooperative. In the case of SS, due to the fact that the non-stationary characteristic of noise a posteriori and a biased SNR are estimated inaccurately, there may be a distortion in the shape of harmonics by subtracting the noise magnitude more in the harmonics region and less in the side-lobes region. As a result, this condition will lead directly to a poor performance in the ASR system. However, HSS also has a remedial capability that corrects the described deficiency. The key to improving the performance of HSS is pinpointing the harmonics corresponding to the spectrum.

Next, HSS is applied to the voice code verification experiment, which is also conducted using PBW452DB. The simulation result is shown in Table 8. In a babble noise environment, EER did not show a rapid decrease of EER, while in white noise, EER showed a rapid decrease of EER. However, it brought about a 40% relative improvement than when there was no harmonics-based spectral subtraction algorithm.

4.3. Discussion

Our proposed method for a text-prompted and a speaker independent verification showed that it provides a voice code verification function without an extra trained model for the likelihood normalization in N -best based LRT. The key point is to use the competing models that are anti-models using a statistical distance of phonemes as the alternative model always follows the same state as the target model.

The conventional speech recognition algorithms as a pattern classification problem use the maximum a posteriori (MAP) decision rule to find the most likely sequence of words. Respective phonemes can be modeled using HMM having states. When uttered speech vectors enter the decoding stage, likelihood scores are computed at all sequences of words until the end-point of speech is found. Thus, a likelihood score of phoneme models similar to the uttered speech features increase, or else the likelihood score will be assigned a low value. When a person utters a word, it can be either a completely different word or a similar word with a voice code word. The conventional speech recognition system obtains recognition success (false acceptance) when a person speaks a similar word because it uses the maximum a posteriori decision rule to find the most likely sequence of words. However, such a problem presents high risk in a user authentication system, requiring anti-models to cope with this problem. Thus, in this paper, we propose anti-models that are specific to a given voice code word model by using regular production rules.

Table 8

Equal error rate (EER) of voice code verification under babble and white noise environments

		Equal error rate (EER)			
		Clean	5 dB	10 dB	15 dB
Clean DB	FRR	0.076738	–	–	–
	FAR	0.109624	–	–	–
Babble noise	FRR	–	0.088653	0.088069	0.089238
	FAR	–	0.096128	0.095923	0.095385
White noise	FRR	–	0.518268	0.376042	0.221871
	FAR	–	0.362295	0.320828	0.218299

Thus, if we can perform the modeling of the alternative hypothesis very well, the voice code verification task can be solved by competing against each other without extra trained models such as filler or garbage models. In some cases, traditional LLR does not find the most representative alternative hypothesis (Leung and Fung, 1999). Thus, the modeling of an alternative model is a crucial issue for the voice code verification.

From the simulation result, we knew that the use of numerous anti-models degraded detection probability while the use of few anti-models degraded the false acceptance rate. Thus, the optimal number of anti-models that can compete with the voice code model should be used. In addition, outdoor noise is an important issue that should be considered. Under this condition, the speaker verification rate is very low while the voice code verification rate is the same. Even though we applied a harmonic-based spectral subtraction algorithm, some other algorithms should also be used for compensating the verification rate in an outdoor environment where wind or rain noise exists, and so on. Considering its application to an outdoor environment, some kinds of feature compensation algorithms such as the multi-variate Gaussian-based feature compensation or the parallel combined mixture model (Kim et al., 2003) should be considered.

This system can be utilized by combining it with other security algorithms. To prevent and compensate for the false rejection or verification failure, we can use other methods such as face recognition and fingerprint recognition using an image sensor. Thus, the proposed system can make use of verifying suspicious persons with other verification techniques. By combining image techniques into our proposed system for detection, tracking and recognition of people, more reliable detection and recognition rate can be achieved.

5. Conclusions

We proposed a voice code verification method for a text-prompted speaker independent verification, based on an utterance verification method using the sub-word based anti-models that employ

the statistical distance of phonemes against the voice code. Our proposed method for a text-prompted and speaker independent verification provided a voice code verification function without an extra trained model such as filler or garbage models for the likelihood normalization through the reuse of a general acoustic model. At this point, the use of numerous anti-models degraded the detection probability while the use of few anti-models degraded the false acceptance rate. Thus, the optimal number of anti-models that can compete with the voice code model should be used.

In experiment, the performance evaluation is done by using a common Korean database, PBW452DB, which consists of 63,280 utterances of 452 isolated words recorded in a silent environment. The result was improved by 16% over the utterance verification result presented in Table 6. In addition, simulation result showed that the performance is higher under noisy environment than in any other algorithms when we applied the harmonics-based spectral subtraction algorithm compared to general spectral subtraction and nonlinear spectral subtraction.

Acknowledgement

This work was supported by grant No. 10012805 from the Korea Institute of Industrial Technology Evaluation & Planning Foundation.

References

- Ahn, S.J. et al., 2002. On effective speaker verification based on subword model. In: *International Conference on Spoken Language Processing (ICSLP) 2002*, Denver, USA, September, Vol. 2, pp. 1361–1364.
- Beh, J.H., Ko, H., 2003. A novel spectral subtraction scheme for robust speech recognition: spectral subtraction using spectral harmonics of speech. *ICME 2003*, July, pp. III 633–III 636.
- Benitez, M.C. et al., 2000. Different confidence measures for word verification in speech recognition. *Speech Comm.* 32, 79–94.
- Huang, X. et al., 2001. *Spoken Language Processing*. Prentice Hall PTR.
- Jiang, H., Lee, C.H., 2003. A new approach to utterance verification based on neighborhood information in model space. *IEEE Trans. Speech Audio Process.* 11 (5).

- Kim, T.Y., Ko, H., 2003. Utterance verification under distributed detection and fusion framework. *Eurospeech 2003*, September, pp. 889–892.
- Kim, W.I. et al., 2003. Feature compensation scheme based on parallel combined mixture model. *Eurospeech 2003*, September, pp. 677–680.
- Leung, L.K., Fung, P., 1999. A more efficient and optimal LLR for decoding and verification. *Proceedings of ICASSP '99*, IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, 15–19 March, 1999.
- Lleida, E., Rose, R.C., 2000. Utterance verification in continuous speech recognition: Decoding and training procedures. *IEEE Trans. Speech Audio Process.* 8 (2).
- Park, H.S. et al., 2002. Temporal and spectral characteristics of Korean phonation types. Doctor of philosophy degree thesis, The University of Texas at Austin, August.
- Rahim, M.G. et al., 1995. Robust utterance verification for connected digits recognition. In: *ICASSP-95*, International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, 9–12 May, 1995.
- Rahim, M.G. et al., 1997. Discriminative utterance verification for connected digits recognition. *IEEE Trans. Speech Audio Process.* 5 (3).
- Sankar, A., Wu, S.-L., 2003. Utterance verification based on statistics of phone-level confidence scores. In: *Proceedings of (ICASSP '03)*, IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, 6–10 April, 2003.
- Sukkar, R.A., Lee, C.H., 1996. Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition. *IEEE Trans. Speech Audio Process.* 4 (6).
- Tomoko, M., Furui, S., 1995. Likelihood normalization for speaker verification using a phoneme- and speaker-independent model. *Speech Comm.* 17, 109–116.
- Wessel, F.K., et al., 2000. A comparison of word graph and N-best list based confidence measures. In: *Proceedings of ICASSP-2000*, pp. 1587–1590.
- Wessel, F. et al., 2001. Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. Speech Audio Process.* 9 (March).
- William, J. et al., 1997. *The Handbook of Phonetic Sciences*. Blackwell publishers Ltd.
- Wu, C.-H. et al., 2000. Integration of phonetic and prosodic information for robust utterance verification. *IEE Proc. Vision Image Signal Process.* 147 (1).
- Xiang, B., Berger, T., 2003. Efficient text-independent speaker verification with structural Gaussian mixture models and neural network. *IEEE Trans. Speech Audio Process.* 11 (5), 447–456.
- Xin, L., Wang, B.X., 2001. Utterance verification for spontaneous Mandarin speech keyword spotting. In: *Proceedings of ICII 2001—Beijing*, 2001 International Conferences on Info-tech and Info-net, Vol. 3, 29 October–1 November, 2001.