Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 1-12

Toward Constructing A Multilingual Speech Corpus for Taiwanese (Min-nan), Hakka, and Mandarin

Ren-yuan Lyu[*], Min-siong Liang[+], Yuang-chin Chiang[**]

Abstract

The **Formosa** speech database (ForSDat) is a multilingual speech corpus collected at Chang Gung University and sponsored by the National Science Council of Taiwan. It is expected that a multilingual speech corpus will be collected, covering the three most frequently used languages in Taiwan: Taiwanese (Min-nan), Hakka, and Mandarin. This 3-year project has the goal of collecting a phonetically abundant speech corpus of more than 1,800 speakers and hundreds of hours of speech. Recently, the first version of this corpus containing speech of 600 speakers of Taiwanese and Mandarin was finished and is ready to be released. It contains about 49 hours of speech and 247,000 utterances.

Keywords: Phonetic Alphabet, Pronunciation Lexicon, Phonetically Balanced Word, Speech Corpus

1. Introduction

To design a speaker independent speech recognition system, it is essential to collect a large-scale speech database. Taiwan (also called ***Formosa*** historically), which has become famous for its IT industry, is basically a multilingual society. People living in Taiwan usually speak at least two of the three major languages, including Taiwanese (also called Min-nan in the linguistics literature), Hakka and Mandarin, which are all members of the Chinese language family. In the past several decades, most of the researchers studying natural language processing, speech recognition and speech synthesis in Taiwan have devoted themselves to research on Mandarin speech. Several speech corpora of Mandarin speech have, thus, been collected and distributed [Wang *et al*., 2000; Godfrey, 1994]. However, little has been done

[*] Dept. of Computer Science and Information Engineering, Chang Gung University, Taoyuan, Taiwan
Email: rylyu@mail.cgu.edu.tw Tel: 886-3-2118800ext5967, 5709
[+] Dept. of Electrical Engineering, ,Chang Gung University, Taoyuan, Taiwan
[**] Inst. of Statistics, National Tsing Hua University, Hsin-chu, Taiwan

on the other two languages used in daily life. In this paper, we describe a government-sponsored project which aims to collect a large-scale multilingual speech corpus, namely, the ***Formosa Speech Database*** (ForSDat), covering these three languages used in Taiwan. The construction of ForSDat is a 3-year project, the goal of which is to collect hundreds of hours of speech from up to 1,800 speakers. So far, we have finished about one-thrid of what the project is expected to achieve.

This paper is organized as follows. Section 1 is the introduction. Section 2 describes the ***Formosa Phonetic Alphabet*** (ForPA), which is being used to transcribe all the speech data and the pronunciation lexicons. Section 3 discusses the phonetically balanced word sheets used to record speech utterances. Section 4 reports the software tools used for corpus collection. Section 5 describes the information obtained about speakers. Section 6 provides information about the database information. Section 7 discusses data validation, and section 8 is a conclusion.

2. The Phonetic Alphabet and the Pronunciation Lexicon

One of the preliminary jobs involved in constructing a speech corpus is to build up a pronunciation lexicon. We have set up several pronunciation lexicons composed of more than 60,000 words for Taiwanese, more than 70,000 words for Mandarin and more than 20,000 words for Hakka. Each item in the lexicons contains a Chinese character string and a string of phonetic symbols encoded in the ***Formosa Phonetic Alphabet*** (ForPA), which will be described in the following paragraphs.

2.1 Formosa Phonetic Alphabet (ForPA)

Many symbolic systems have been developed for labeling the sounds of languages used throughout the world. One of the most popular systems is the International Phonetic Alphabet (IPA). Since many IPA symbols are not defined in the ASCII code set and are not easy to manipulate, many ASCII-coded IPA symbolic sets have been proposed in the literature. Two popular systems are SAMPA [Wells, 2003] and WorldBet [Hieronymus, 1994]. It has claimed that one can select parts of these phone sets for a specific language. However, both ASCII-coded phonetic systems have many symbols that are difficult to read, such as “@”or “&”. In addition, since these systems are designed for all the languages used around the world, they are too complex to be applied to some local languages, like those that will be addressed here.

The most widely known phonetic symbol sets used to transcribe Mandarin Chinese are the Mandarin Phonetic Alphabet (MPA, also called Zhu-in-fu-hao) and Pinyin (Han-yu-pin-yin), which have been officially used in Taiwan and Mainland China, respectively, for many years. However, both systems are inadequate for application to the other members of the Chinese language family, like Taiwanese (Min-nan) and Hakka. Among the phonetic systems useful for Taiwanese and Hakka, there are Church Romanized Writing (CR, also call Peh-e-ji, 「白話字」) [Chiung 2001] for Taiwanese and the Taiwan Language Phonetic Alphabet (TLPA) [Ang 2002] for Taiwanese and Hakka. Because the same phonemes are represented using different symbols in Pinyin, CR and TLPA, it is confusing to learn these phonetic systems simultaneously. For example, the syllable "pa(八)" in TLPA and "pa(趴)" in CR may be confused with each other because the phoneme /p/ is pronounced differently in the two systems.

Therefore, it is necessary to design a more suitable phoneme set for multilingual speech data collection and labeling [Zu, 2002][Lyu, 2000]. The whole phone set for the three major languages used in Taiwan is listed in Table 1 for four phonetic systems: MPA, Pinyin, IPA, and the newly proposed ForPA. Table 1 also lists examples of syllables and characters which contain the target phonemes.

It is known that phonemes can be defined in many different ways, depending on the level of detail desired. The labeling philosophy adopted in ForPA is that when faced with various choices, we prefer not to divide a phoneme into distinct allophones, except in cases where the sound is clearly different to the ear or the spectrogram is clearly different to the eye. Since labeling is often performed by engineering students and researchers (as opposed to professional phoneticians), it is generally safer to keep the number of units as small as possible, assuming that the recognizer will be able to learn any finer distinctions that might exist within any context. Generally speaking, ForPA might be considered as a subset of IPA, but it is more suitable for application to the languages used in Taiwan.

Table 1. The phone set for the three languages in Taiwan, represented as different phonetic systems. The Chinese character in parentheses followed by a syllable, is an example character used in Mandarin, e.g., "ba(八)" is pronounced in Mandarin as syllable "ba", without considering the tone. For phonemes not found in Mandarin Chinese, we use Chinese character pronounced as Taiwanese (^{T}) or Hakka (^{H}) to be example characters. For example, "bha(肉^T)" meaning "肉" is pronounced "bha" in Taiwanese.

ForPA	b	p	m	f	d	t	n	l	g	k	h	zh	ch	sh
Syllable (字)	ba(八)	pa(趴)	ma(媽)	fa(發)	da(搭)	ta(他)	na(那)	la(拉)	ga(嘎)	ka(咖)	ha(哈)	zha(渣)	cha(差)	sha(殺)
Pinyin	b	p	m	f	d	t	n	l	g	k	h	zh	ch	sh
MPA	ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	ㄓ	ㄔ	ㄕ
IPA	p	p'	m	f	t	t'	n	l	k	k'	x	tʂ	tʂ'	ʂ
SAMPA	p	p_h	m	f	t	t_h	n	l	k	k_h	x	ts`	ts_h`	s`
WorldBet	p	ph	m	f	t	th	n	l	k	kh	x	tsr	tsrh	sr

ForPA	rh	z	c	s	r	bh	gh	v	ng
Syllable (字)	rhan(然)	za(匝);zi(機)	ca(擦);ci(七)	sa(撒);si(西)	ru(如^{T});ri(字^{T})	bha(肉^{T})	ghua(我^{T})	voi(會^{H})	ang(骯);nga(雅);ng(黃)
Pinyin	r	z;j	c;q	s;x					-ng
MPA	ㄖ	ㄗ;ㄐ	ㄘ;ㄑ	ㄙ;ㄒ					ㄤ
IPA	ʐ	ts;tɕ	ts';tɕ'	s;ɕ	z;ü	b	g	v	N
SAMPA	z`	ts`;ts\	ts_h;ts_h	s;s\	z;z\	b	g	v	N
WorldBet	zr	ts;cC	tsh;chC	s;C	z;zr	b	g	v	N

ForPA	a	o	er	e	err	i	u	yu	ii	-nn	-p	-t	-k	-h
Syllable (字)	a(阿)	o(喔)	er(鵝)	ie(也)	err(而	i(一)	u(吳)	yuan(原)	zii(資)	ann(餡TH)	ap(壓TH)	at(握TH)	ak(沃TH)	ah(鴨TH)
Pinyin	a	o	e	ê	er	y;i	w;u	yu;ü	i					
MPA	ㄚ	ㄛ	ㄜ	ㄝ	ㄦ	ㄧ	ㄨ	ㄩ	ㄭ					
IPA	a	o	ɤ	ɛ	ɚ	i	u	y	ï	ã	-p	-t	-k	-h
SAMPA	A	o	7	E	@`	i	u	y	ï`					
WorldBet	A	o	2	E	&r	i	u	y	4r	~				

2.2 Formosa Lexicon (ForLex): A Pronunciation lexicon composed of Taiwanese, Hakka and Mandarin

Before producing word sheets for speakers to utter, a complete pronunciation lexicon needs to be prepared. A lexicon has been collected in this project to meet the requirement. This lexicon, called the Formosa Lexicon (ForLex), was adapted from three other lexicons: the CKIP Mandarin lexicon , Gang's Taiwanese lexicon, and Syu's Hakka lexicon [CKIP 2003] [Syu 2001]. Some statistical information about the lexicon was listed in Table 2.

Table 2. The distribution of words in three lexicons: Gang's Taiwanese lexicon, Syu's Hakka lexicon, and the CKIP Mandarin lexicon.

	1-Syl	2-Syl	3-Syl	4-Syl	5-Syl	
Gang	8027	44846	12129	1823	161	
Syu	7322	9161	4948	2382	21	
CKIP	6863	39733	8277	9074	435	
	6-Syl	7-Syl	8-Syl	9-Syl	10-Syl	Total
Gang	0	0	0	0	0	66986
Syu	3	0	0	0	0	23837
CKIP	223	125	52	2	8	64792

3. The process of producing phonetically balanced word sheets

Based on the three pronunciation lexicons transcribed in ForPA, we extracted sets of distinct syllables and inter-syllabic bi-phones from the three languages. The statistics of the phonetic units considered here are listed in Table 3. In order to collect speech data related to the co-articulation effect of continuous speech, we extracted phonetically abundant word sets. Therefore, the chosen phonetic units were not only base-syllables, phones, and RCD phones, but also Initial-Finals, RCD Initial-Finals and inter-syllabic RCD phones. The process of selecting such a word set is actually a set-covering optimization problem [Shen *et al*., 1999], which is NP-hard. Here, we adopted a simple greedy heuristic approximate solution [Cormen, 2001].

First, we set the requirements of the word set as to cover the following phonetic units: Base-syllables and Inter-syllabic RCD phones. Accordingly, the selected word set could cover all the phones, Initial-Finals, RCD phones, RCD Initial-Finals, Base-syllables and Inter-syllabic RCD phones. In this way, we could obtain several sets of words for our balance-word data sheets. All the statistics of the phonetic units considered here are listed in Table 3. [Liang 2003].

Table 3. The numbers of distinct subwod units for each of the three languages and their unions, where T: Taiwanese; H: Miaulik-Hakka M: Mandarin; ∪: union.

Language	Base syllable	Phones	Within-syllabic bi-phones	Inter-syllabic bi-phones
T	832	53	410	716
H	683	53	327	696
M	429	45	208	234
T∪H	1134	70	583	1036
T∪M	1055	64	486	809
H∪M	939	71	435	797
T∪H∪M	1326	78	600	1105

3.1 Data sheets

The process of producing data sheets is depicted in Fig.2. Before we produced the data sheets, we defined the sheets' coverage rate. The coverage rate of the sheets was defined as the total number of base-syllables (or inter-syllabic phones) over the number of all possible distinct base-syllables (or inter-syllabic phones). The format of the data sheet is partially shown in Table 4.

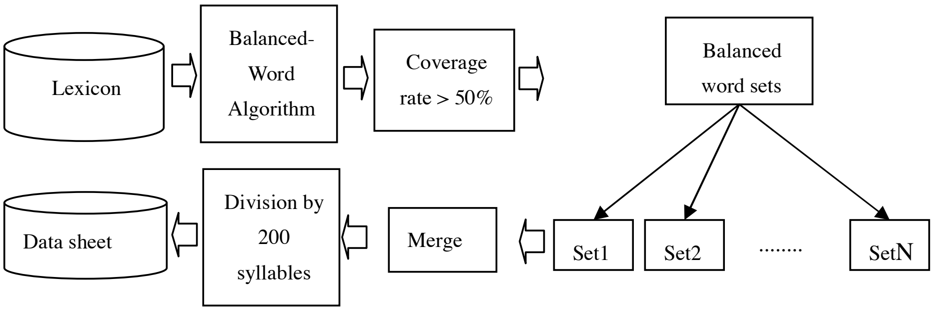


Figure 2. The process of producing data sheets.

Table 4. Some examples from the data sheets used to collect ForSDat.

Filename	Text	Transcription in ForPA
blwr00000	觀世音菩薩	guan1_se3_im1_po5_sat7
blwr00001	驚 ga 刺激著	giann1_ga2_ci3_gik7_diorh6
blwr00002	藥檢實驗室	iorh6_giam4_sit6_ghiam2_sik7
blwr00003	藝術工作者	ghe2_sut6_gang1_zok7_zia4

In terms of Taiwanese sheets, although we produced 364 balanced-word sets in total, we only used sets whose coverage rates exceeded 50%. Because the variation in the numbers of syllables or words in some sets was very high, we merged those sets and then re-segmented them to produce data sheets. Finally, each sheet contained about 200 syllables. The numbers of data sheets and total words were 446 and 37,275, respectively.

As for Miaulik-Hakka sheets, all the balanced-word sets were concatenated in sequence and then segmented into data sheets, each of which contained 70 words. Finally, we got 340 data sheets, which consisted of 23,837 words.

For Mandarin, one phonetically-rich set was segmented equally into ten sheets, and every sheet consisted of roughly 300 words. In addition, all the tonal-syllables were segmented into ten equal-size data sheets.

4. The software tools used for corpus collection

Two kinds of database collection systems are being used to create ForSDat. They are microphone and telephone systems, respectively.

4.1 The telephone recording system

The telephone system is set up in the Multi-media Signal Process Laboratory at Chang Gung University. The speakers dial into the laboratory using a handset telephone. Before recording, we give the speakers prompt sheets. The input signal is in format of 8K sampling rate with 8-bits μ-law compression. The speakers utter words while reading the prompt sheet, and supervised prompt speech is played to help the speakers follow the prompt speech to finish the recording. After recording, all speech data are saved in a unique directory. Figure 3 shows the recording process carried out using the telephone system.

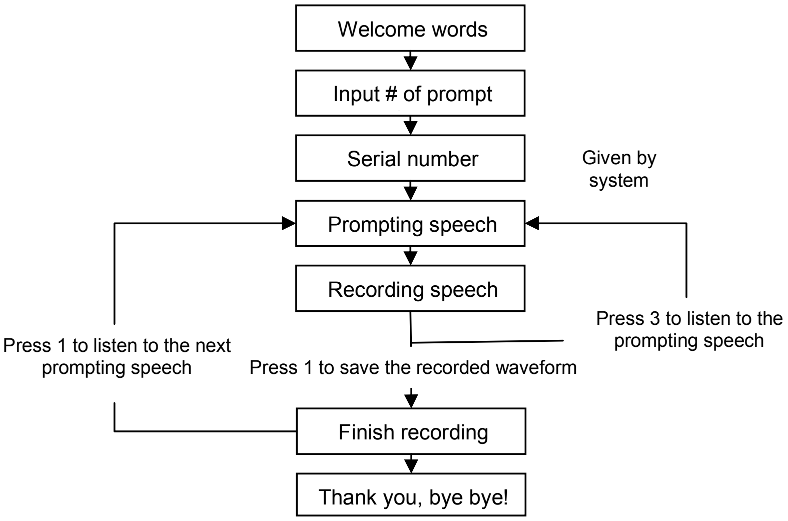


Figure 3. The telephone recording system.

4.2 The microphone recording system

When we record a waveform into a computer, it is not convenient to type the file name necessary for saving it. Therefore, we use a good tool (DQS3.1) [Chiang 2002] to record speech. If we create a script in a specific form for this software, we can record the waveform easily and get a labeled file, which contains information of transcription using ForPA. Then, we simply set up the system on a notebook computer and take it wherever we want to record speech.

5. Speaker recruiting

We employ several part-time assistants to recruit speakers around Taiwan. Each speaker is asked to record one sheet and receives a remuneration after finishing recording. Each part-time assistant receives a remuneration when they recruits a speaker.

5.1 Profiles of speakers

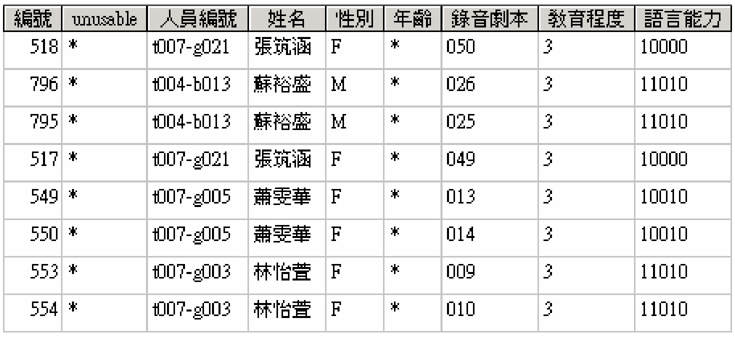
After a recording is finished, we ask the speakers to provide us with their profiles. This is useful for arranging speech data later. The user can also design experiments according to these profiles (see Fig. 4). The profile of a speaker includes the following attributes:

i. the name and gender of the speaker;

ii. the age and birthplace of the speaker;

iii. the location of the speaker and time;

iv. the number of years of education of the speaker.



編號	unusable	人員編號	姓名	性別	年齡	錄音劇本	教育程度	語言能力
518	*	t007-g021	張筑涵	F	*	050	3	10000
796	*	t004-b013	蘇裕盛	M	*	026	3	11010
795	*	t004-b013	蘇裕盛	M	*	025	3	11010
517	*	t007-g021	張筑涵	F	*	049	3	10000
549	*	t007-g005	蕭雯華	F	*	013	3	10010
550	*	t007-g005	蕭雯華	F	*	014	3	10010
553	*	t007-g003	林怡萱	F	*	009	3	11010
554	*	t007-g003	林怡萱	F	*	010	3	11010

Figure 4. A portion of a speaker's profile in the database.

5.2 Speech data format

We save the utterance in a binary file. If the speech is recorded using a microphone, we save it as a 16KHz/16bits PCM file and a corresponding label file that contains the phonetic transcription for a word. Otherwise, we save the utterances as a 8KHz/8bits μ -law file if the speech data were obtained over a telephone.

6. Database information

The database has been collected over both microphone and telephone channels, namely, ForSDat-TW01, ForSDat-MD01 and ForSDat-TW02, respectively. The tag "TW01" means that a portion of the database was collected in 2001 in Taiwanese. In the other hand, the tag

"M0" means that the recording channel used was a microphone and gender was female, and so on. Every speaker has a unique serial number and speech data, which contain a transcription of waveforms made in the early stage and are stored in a unique folder named according to the serial number. The database structure is shown in Fig.5. All the statistics of the database are listed in Table 5.

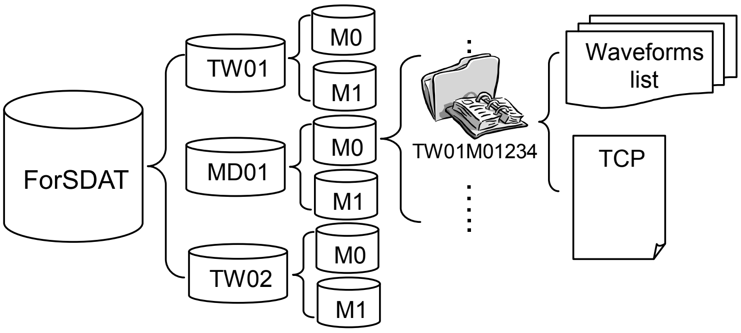


Figure 5. The structure of database for Taiwanese and Mandarin. (TW01: Taiwanese database collected in 2001; M0: the microphone channel was used and the gender was female, T1: the telephone channel was used and the gender was male; and so on. There is a transcription file for each unique speaker.)

Table 5. The statistics of utterances, speakers and data length for speech collected over microphone and telephone channels in Taiwanese and Mandarin (MIC: microphone; TEL: telephone).

	Name	Channel	Gender	Quantity	Train(hr)	Test (hr)
ForSDAT	TW01-M0	MIC	Female	50	5.92	0.29
	TW01-M1		Male	50	5.44	
	MD01-M0		Female	50	5.65	0.27
	MD01-M1		Male	50	5.42	
	TW02-M0		Female	233	10.10	0.70
	TW02-M1		Male	277	11.66	
	TW02-T0	TEL	Female	580	29.21	0.95
	TW02-T1		Male	412	19.37	

7. Database validation

After the speakers have finished recording, the speech data need to be validated. This step can guarantee that the speech data will be useful for training the acoustic models of the speech recognizer. Although the data sheets are designed to be as readable as possible and we provide prompting speech for speakers, the utterances still are not compatible with the prompt. We thus validate the speech data using a specially designed software tool, which has the user interface shown in Fig.6 and the functions described in the following subsections.

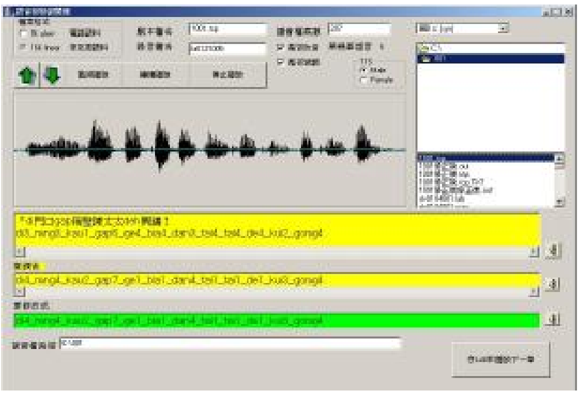


Figure 6. The software tool for validation.

7.1 Step 1: pre-processing

We browse all the waveforms using the validation tool and check whether the following problems occur:

1. the voice is cut off;i.e., the speakers pronounce too fast;

2. the voice file is empty;

3. there are other sounds mixed into the waveform, such as the voices of other people or the sounds of vehicles;

4. the speakers laughed when the waveform was being recorded.

If any one of the above problems are found, the speech file is considered unusable. If the total number of unusable files exceeds 10% of all the files in the directory, the directory is considered unusable. The speaker will then be asked to record the work sheet again.

Other problems may also occur. For example, two speakers may record speech data inturns in one work sheet, etc. These directories are also considered unusable.

7.2 Step 2: phonetic transcription by means of forced alignment

After the speech data is pre-processed, we validate it to determine whether the labels that consist of phonetic transcriptions correspond to the speech data. We use two methods to achieve this goal. First, we use HTK [Steven, 2002] to perform forced-alignment automatically on an utterance using all possible syllable combinations. We keep the highest scores for combinations to transcribe the speech. Secondly, we use the TTS (text-to-speech) technique to synthesize all the labels that were transcribed using HTK and then we transcribe the speech manually using more appropriate phonetic symbols. Finally, we can construct a relational database using ACCESS to record all the profiles of the speakers (see Fig.3) and what they recorded. Therefore, we can query the speech database using the SQL language to find the waveforms transcribed using the specific phones or syllables or even query who recorded the specific-phone waveforms. This step is on-going and will be finished soon.

8. Conclusion

Version 1.0 of this corpus containing the speech of 600 speakers of Taiwanese (Min-nan) and Mandarin Chinese has been finished and is ready to be released. We have collected the speech of 1,773 people, including 49.47 hours of speech and 247,027 utterances. As work on this project continues, more Hakka and Mandarin speech data will be collected.

References

Wang, H. C., F. Seide, C.Y. Tseng and L.S. Lee, "Mat-2000 – design, collection, and validation of a mandarin 2,000-speaker telephone speech database," *International Conference on Spoken Language Processing 2000*, Beijing, China, 2000.

Godfrey, J., "Polyphone: Second anniversary report," *International Committee for* Co-*ordination and* S*tandardisation of Speech Databases Workshop 94*, Yokohama, Japan, 1994.

Zu, Y., "A super phonetic system and multi-dialect Chinese speech corpus for speech recognition," *International Conference on Spoken Language Processing 2002*, Denver, USA, 2002.

Lyu, R. Y., "A bi-lingual Mandarin/Taiwanese (Min-nan), Large Vocabulary, Continuous speech recognition system based on the Tong-yong phonetic alphabet (TYPA)," *International Conference on Spoken Language Processing 2000*, Beijing, China, 2000.

Liang, M. S., R. Y. Lyu and Y. C. Chiang, "An efficient algorithm to select phonetically balanced scripts for constructing corpus," *IEEE International Conference on Natural Language Processing and Knowledge Engineering 2003*, Beijing, China, 2003.

Shen, J. L., H. M. Wang, R. Y. Lyu and L. S. Lee, "automatic selection of phonetically distributed sentence sets for speaker adaptation with application to large vocabulary mandarin speech recognition," *Computer speech and language*, vol. 13, no. 1,pp. 79-97, Jan. 1999.

Cormen, T. H. ect, "Chapter 37: Approximation Algorithm", *Introduction to Algorithm*, pp. 974-978, 2001.

Chiang, Y. C., and R. Y. Lyu, the speech recording system which are developed by Dr. Yuang-Chin Chiang at Nation Tsing Hua University, 2002.

Steven, Y., "The HTK book version 3.2", Cambridge University Engineering Department, 2002.

Wells, J., SAMPA (Speech Assessment Methods Phonetic Alphabet), http://www.phon.ucl.ac.uk/home/sampa/home.htm, April, 2003.

CKIP, Chinese Knowledge Information Processing, http://rocling.iis.sinica.edu.tw/ CKIP/, 2003.

Syu, J. C., "Hakka dictionary of Taiwan", Nantian Bookstore published, 2001.

Hieronymus, J., "ASCII phonetic symbols for the world's languages: Worldbet," *AT&T Bell Laboratories, Technical Memo*, 1994.

Ang, U., Taiwan Language Phonetic Alphabet(TLPA), Taiwan Languages and Literature Society, http://www.tlls.org.tw/, 2002.

Chiung, W. V. T., "Romanization and Language Planning in Taiwan," *The Linguistic Association of Korea Journal 9(1),* pp. 15-43, 2001.

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 13-28

Multiple-Translation Spotting for Mandarin-Taiwanese Speech-to-Speech Translation

Jhing-Fa Wang[*], Shun-Chieh Lin[*], Hsueh-Wei Yang[*], and Fan-Min Li[*]

Abstract

The critical issues involved in speech-to-speech translation are obtaining proper source segments and synthesizing accurate target speech. Therefore, this article develops a novel multiple-translation spotting method to deal with these issues efficiently. Term multiple-translation spotting refers to the task of extracting target-language synthesis patterns that correspond to a given set of source-language spotted patterns in conditional multiple pairs of speech patterns known to be translation patterns. According to the extracted synthesis patterns, the target speech can be properly synthesized by using a waveform segment concatenation-based synthesis method. Experiments were conducted with the languages of Mandarin and Taiwanese. The results reveal that the proposed approach can achieve translation understanding rates of 80% and 76% on average for Mandarin/Taiwanese translation and Taiwanese/Mandarin translation, respectively.

Keywords: Multiple-Translation Spotting, Speech-to-Speech Translation

1. Introduction

Automatic speech-to-speech translation is a prospective application of speech and language technology [See JANUS III [Lavie *et al.* 1997], Verbmobil [W. Wahlster 2000], EUTRANS [Casacuberta *et al.* 2001] and ATR-MATRIX [Sugaya *et al.* 1999]]. However, the unsolved problems in speech-to-speech translation are how to obtain proper source segments and how to generate accurate target sequences while the system performance is degraded by speech input. With the rising importance of parallel texts (bitexts) in language translation, an approach called translation spotting has been applied for proposing appropriate translations, referring to the TransSearch system [Macklovitch *et al.*, 2000] and sub-sentential translation memory systems [M. Simard, 2003]. Previous works in this area have suggested that manual review or

[*] **Corresponding author:**
Prof. Jhing-Fa Wang, Department of Electrical Engineering, National Cheng Kung University, No.1, Dasyue Rd., East District, Tainan City 70101, Taiwan, R.O.C.
Email: wangjf@csie.ncku.edu.tw Tel: 886-6-2757575 ext. 62341 Fax: 886-6-2746867

crafting is required to obtain example bases of sufficient coverage and accuracy to be truly useful.

Translation spotting (TS) is a term coined by Véronis and Langlais [2000] and refers to the task of identifying word tokens in a target-language (TL) translation that correspond to some given word-patterns in a source-language (SL) text. This process takes as input a couple, i.e., a pair of SL and TL text segments known to be translation patterns, and an SL query, i.e., a subset of the patterns of the SL segment, on which the TS will focus its attention. In more formal terms:

> The input to the TS process is a pair of SL and TL text segments $\langle S,T \rangle$ and a contiguous, non-empty input sequence of word-tokens in SL, $q = s_1 \mathbf{L} s_n$.
>
> The output is a pair of sets of translation patterns $\langle r_q(S), r_q(T) \rangle$: the SL answer and TL answer, respectively.

Table 1 shows some examples of TS, where the words in italics represent the SL input, and the words in bold are the SL and TL answers. As can be seen in these examples, the patterns in the input q and answers $r_q(S)$ and $r_q(T)$ may or may not be contiguous (examples 2 and 3), and the TL answer may possibly be empty (example 4) when there is no satisfactory way of linking TL patterns to the input. By varying the identification criteria, the translation spotting method can help evaluate units over various dimensions, such as frequency ranges, parts of speech and even speech features of spoken language.

Table 1. Translation spotting examples.

Query		Sentence Pair	
		SL (Mandarin)	TL (Taiwanese)
1.	*q:待 幾 天*	你 預計 要 待 幾 天	lie phahsngx bueq doax kuie jit
		$r_q(S)$={待,幾,天}	$r_q(T)$={doax,kuie,jit}
2.	*q:我 要 訂 兩 間 單人房*	我 明天 要 訂 兩 間 有 淋浴 設備 的 單人房	minafzaix goar bueq dexng lerng kefng u sea sengqw e danjiin paang
		$r_q(S)$={我,要,訂,兩,間,單人房}	$r_q(T)$={goar,bueq,dexng,lerng,kefng,danjiin paang}
3.	*q:今晚 有 […] 雙人房 嗎*	請 問 你們 今晚 有 一 間 雙人房 嗎	chviar bun lirn ehngf u cit kefng sianglaang paang but
		$r_q(S)$={今晚,有,雙人房,嗎}	$r_q(T)$={ehngf,u,sianglaang,paang,but}
4.	*q:包括 … 在 內*	有 包括 早餐 在 內?	u zafdngx but
		$r_q(S)$={包括,在,內}	$r_q(T)$={ϕ }

However, translation spotting can only draw out the TL answer from the best translation; it can not handle an SL query whose word-tokens are distributed in different translations. Consequently, we propose conducting multiple-translation spotting of a speech input using multiple pairs of translation patterns. Figure 1 shows an example of multiple-translation spotting of a speech input. When a speaker inputs an SL speech query ”今晚會有三間單人房嗎”, the proposed system can obtain a TL speech pattern set that includes five elements, ”ehngf”, ”kvaru”, ”svaf”, ”kefng”, and ”danjiinpaang”, according to the spotted SL speech patterns ”今晚”, “會有”, “間”, “嗎”, ”三”, and ”單人房”. The rest of this article is organized as follows. Section 2 presents the framework of the proposed system. Section 3 presents system data training for Mandarin and Taiwanese. Section 4 describes the proposed translation method for speech-to-speech translation. Section 5 presents experimental results. Finally, Section 6 draws conclusions.

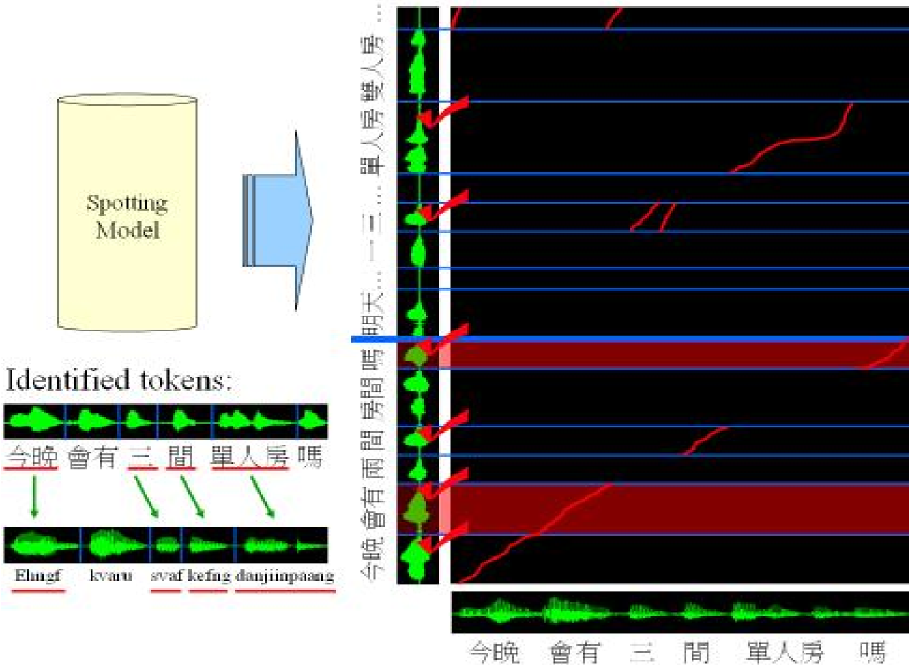

Spotting Model
Identified tokens:
今晚 會有 三 間 單人房 嗎
Ehngf kvaru svaf kefng danjiinpaang
今晚 會有 兩 間 房間 嗎 明天... 一 三 ... 單人房 雙人房 ...
今晚 會有 三 間 單人房 嗎

Figure 1. An example of multiple-translation spotting.

2. Framework of the Proposed System

The proposed speech-to-speech translation system is divided into two phases – a training phase and a translation phase. In the training phase, the developed translation examples are imported to derive multiple-translation templates and develop speech data. In the following step, the developed speech data are applied to construct multiple-translation spotting models and synthesis templates. Figure 2(a) shows a block diagram of the training phase.

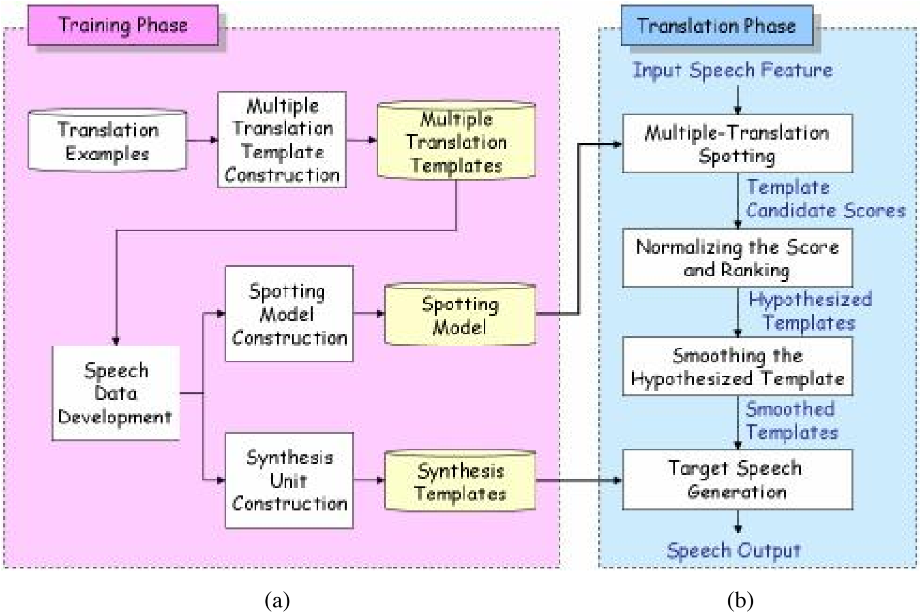

Training Phase
Translation Examples
Multiple Translation Template Construction
Multiple Translation Templates
Speech Data Development
Spotting Model Construction
Spotting Model
Synthesis Unit Construction
Synthesis Templates
Translation Phase
Input Speech Feature
Multiple-Translation Spotting
Template Candidate Scores
Normalizing the Score and Ranking
Hypothesized Templates
Smoothing the Hypothesized Template
Smoothed Templates
Target Speech Generation
Speech Output
(a)
(b)

Figure 2. Framework of the proposed system: (a) a training phase; (b) a translation phase.

Figure 2(b) shows a block diagram of the translation phase. A one-stage based spotting method is adopted to identify input spoken phrases for each spotting template, and the template candidates are assigned in the following score normalization and ranking process. However, the hypothesized word sequence generally includes noise-like segments. Accordingly, the segments are adjusted by smoothing the hypothesized word sequences. After the hypothesized word sequences of all template candidates have been smoothed, the hypothesized target sequences are generated using the translation template with the maximum number of spotting tokens of speech input. The obtained target speech segments are used to produce target speech by means of the corresponding synthesis template in the final target generation process.

3. Data Training Phase

As for the task of translating Mandarin and Taiwanese language pairs, although these languages both belong to the family of Chinese languages, their language usages still have various development by language families and their origins, Mandarin belongs to Altaic language family, and Taiwanese belongs to Sinitic language family [Sher *et al.*, 1999]. Therefore, in the following section, we will consider their language usages for three template construction.

3.1 Multiple Translation Template Construction

While translation templates can be fully constructed, one major issue in translation pattern exploitation, called "divergence," makes straightforward transfer mapping extraction impractical. Dorr (1993) describes divergence in the following way: "translation divergence arises when the natural translation of one language into another result in a very different form than that of the original." Therefore, we choose translations with no divergence to practice constructing templates. An example of a simple translation template derived from a practicable translated example is shown below.

Translated Example: SL: "我 朋友 要 訂 房間"

↔ TL: "goarn pengiuo bueq dexng pangkefng"

Intention Translation: M_{p1} 要 訂 M_{p2} ↔ T_{p1} bueq dexng T_{p2}

alignments

Variable Translation: If M_{p1}↔ T_{p1}, 我 朋友 ↔ goarn pengiuo

If M_{p2}↔ T_{p2}, 房間 ↔ pangkefng

The translation template is composed of a translated example, an intention translation, and two variable translations. The example shows how a sentence in Mandarin (SL) that contains an intention "要 訂" with two variables, M_{p1} (我 朋友) and M_{p2} (房間), can be translated into a sentence in Taiwanese (TL) with an intention "bueq dexng" and two variables, T_{p1} (goarn pengiuo) and T_{p2} (pangkefng). According to the template, the number of variable translations should be expanded to improve the capability for spotting the speech input. From the preceding example, variable translation expansion can be illustrated as follows:

Variable Translation Expansion:

If M_{p1} ↔ T_{p1},

我 ↔ goar

我 朋友↔ goarn pengiuo

If M_{p2} ↔ T_{p2}, 房間 ↔ pangkefng

票 ↔ phiaux

Therefore, we can obtain corpus-specific multiple translations in a template constructed from three translation patterns, which are “我 朋 友 要 訂 房間↔goarn pengiuo bueq dexng pangkefng”, “我↔goar”, and “票↔phiaux”.

3.2 Spotting Model Construction

Taiwanese is a typical oral language and still has no uniform system of writing. In the literature, there are two ways to represent Taiwanese words: Chinese characters and alphabetic writing. [Sher *et al.*, 1999]. Chinese characters have huge hieroglyph character sets; therefore, it is difficult to systematize developed examples. Although alphabetic writing would be an appropriate representation form, a universal phonemic transcription system is still not available.

Therefore, for the purpose of practical system construction, a collection of speech data is developed from derived text-form templates not only to obtain spotting models but also to transcribe text data as waveform-based representations. For one of the translating languages, the speech data, including intention speech and related variable speech, are used in chorus to construct spotting reference models for use in multiple-translation spotting. Such spotting reference models are embedded with latent grammars from the constructed templates. When dealing with Mandarin-Taiwanese speech feature models, we build the database by extracting LPCC features from recorded template speeches. Hence, when speech recognition is performed, the LPCC features are extracted from the recorded template speeches, and the LPCC features of speech input are used in combination to compute the degree of dissimilarity. After language pairs of both Taiwanese and Mandarin speech data are developed, the transfer mapping information for a pair of Taiwanese and Mandarin speech segments known to be similar in terms of text-form word alignment is constructed.

3.3 Synthesis Template Construction

Both Mandarin and Taiwanese are tonal languages, and it is difficult to determine whether a morpheme will take its inherent tone or the derived tone when every word in a sentence is synthesized. [Wang *et al.*, 1999; Sher *et al.*, 1999]. Therefore, we utilize the obtained intention speech and variable speech as synthesis templates that include intention synthesis units and variable synthesis units. These synthesis units can be used to generate a speech output to be processed using a waveform segment concatenation-based synthesis method [Wang *et al.*, 1999]. For each synthesis unit in the obtained speech data, the following features are stored:

- the waveform and its length,
- the code of the synthesis unit.

4. Translation Phase

4.1 Multiple-Translation Spotting Method

To deal with the problem of spotting between a speech input X_1^L and a translation pattern set $\left\{\left\langle s_j^{(v)}, t_j^{(v)}\right\rangle\right\}_{j=1}^{J}$ in the *v*-th translation template (r_v), we use the standard notation *l* to represent the frame index of $X_1^L, 1 \le l \le L$, *j* to represent the spotting pair ($\left\langle s_j^{(v)}, t_j^{(v)}\right\rangle$) index of r_v, $1 \le j \le J$, and *k* to represent the frame index of *j*-th spotting pattern $s_j^{(v)}$, $1 \le k \le K_j$. Then for each input frame, the accumulated distance $d_A(l,k,j)$ is computed by

$$d_A(l,k,j) = d(l,k,j) + \min_{k-2\le m\le k}\left(d_A(l-1,m,j)\right)\cdot \tag{1}$$

For $2 \le k \le K_j$, $1 \le j \le J$, where $d(l,k,j)$ is the local distance between the *l*-th frame of X_1^T and the *k*-th frame of source pattern $s_j^{(v)}$. The recursion of (1) is carried out of for all internal frames (i.e., $k \ge 2$) of each source pattern. At the speech pattern boundary, i.e., $k = 1$, the recursion can be calculated as follows:

$$d_A(l,1,j) = d(l,1,j) + \min\left\lfloor \min_{1\le m\le J}\left(d_A(l-1,K_m,m)\right), d_A(l-1,1,j)\right\rfloor\cdot \tag{2}$$

The final solution for the best path is

$$d_G^{(v)} = \min_{1\le j\le J}\left\lfloor d_A\left(L,K_j,j\right)\right\rfloor \tag{3}$$

The details of the multiple-translation spotting algorithm are given below:

```
/*  Parameter descriptions
      $\left\{\tau_j^{(v)}\right\}_{j=1}^{J}$ : the spotting results of $\left\{s_j^{(v)}\right\}_{j=1}^{J}$, where
      $\tau_j^{(v)} = \begin{cases} 1, & \text{if SL speech pattern } s_j^{(v)} \text{ is spotted by } X_1^L\cdot; \\ 0, & \text{otherwise.} \end{cases}$
      $w_v \leftarrow \left\{t_j^{(v)} \mid \tau_j^{(v)} = 1,\ 1 \le j \le J\right\}$: the hypothesized TL synthesis patterns;   */

/*  Initialization   */
      $l \leftarrow k \leftarrow j \leftarrow 1$;
      $\tau_j^{(v)} \leftarrow 0, 1 \le j \le J$;
      $w_v \leftarrow \{\ \phi\ \}$;

/*  v-th template spotting   */
      while ($l \le L$)
            for each spotting pattern $s_j^{(v)}$
                  while ($k \le K_j$)
                        if ($k = 1$)
                              $d_A(l,k,j) \leftarrow d(l,k,j) + \min[\min_{1\le j\le J}[d_A(l-1,K_j,j)], d_A(l-1,k,j)]$
```

```
                    $p(l,k,j) \leftarrow \arg\min[\min_{1 \le m \le J}[d_A(l-1,K_m,m)],d_A(l-1,k,j)]$
                **else if** $(k > 1)$
                    $d_A(l,k,j) \leftarrow d(l,k,j) + \min_{k-2 \le m \le k}(d_A(l-1,m,j))$
                    $p(l,k,j) \leftarrow \arg\min_{k-2 \le m \le k}(d_A(l-1,m,j))$
                **else if** $(k = K_j)$
                    $k \leftarrow 1$;
                **else**
                    $k$++;
                **end if**
            **end while**
        **end for**
        $l$++;
    **end while**
    $d_G^{(v)} \leftarrow \min_{1 \le j \le J}[d_A(L,K_j,j)]$;
    $\hat{j} \leftarrow \arg\min_{1 \le j \le J}[d_A(L,K_j,j)]$;
/*  Trace back and TL synthesis pattern extraction  */
    $\{\tau_j^{(v)}\}_{j=1}^J \leftarrow \text{trace back}[(L,K_{\hat{j}},\hat{j})]$; /*, $\tau_j^{(v)}$is assigned as 1 or 0*/
    **for** each $\tau_j^{(v)}$ ,$j$=1,2,…,$J$
        **If** $(\tau_j^{(v)} = 1)$
            $w_v \leftarrow w_v \cup \{ t_j^{(v)} \}$;
        **end if**
    **end for**
    **return** $w_v$ and $\{\tau_j^{(v)}\}_{j=1}^J$;
```

4.2 Normalizing the Score and Ranking

The length of the matching sequence can severely impact the cumulative dissimilarity measurement, so a length-conditioning weight is applied to overcome this defect. Scoring methods that involve the length measurement $\Delta\left(X_1^L,s^{(v)}\right)(s^{(v)} = \bigcup_{j=1}^{J} s_j^{(v)})$ [J.N.K. Liu and L. Zhou, 1998] can be defined in a number of similar ways:

$$\Delta(X_1^L,s^{(v)}) = \max(\|X_1^L\|,\|s^{(v)}\|)(\text{or } \min(\|X_1^L\|,\|s^{(v)}\|) , \tag{4}$$

$$\Delta(X_1^L,s^{(v)}) = \|X_1^L\| * \|s^{(v)}\| , \tag{5}$$

$$\Delta(X_1^L, s^{(v)}) = N(L, \sum_{j=1}^{J} K_j) + F(L, \sum_{j=1}^{J} K_j)/3, \tag{6}$$

where $\left\|X_1^L\right\|$ is the number of frames in speech input x; $\left\|s^{(v)}\right\|$ is the total number of search frames in $\left\{s_j^{(v)}\right\}_{j=1}^{J}$; $N(L, \sum_{j=1}^{J} K_j)$ is the number of frames compared; and $F(L, \sum_{j=1}^{J} K_j)$ is the number of frames that fail to be matched. To improve the flexibility and reliability of the dissimilarity measurement, an exponential $\Delta\left(X_1^L, s^{(v)}\right)$ is exponentially defined as follows:

$$\Delta(X_1^L, s^{(v)}) = \partial^{w_{X,s^{(v)}}}, \tag{7}$$

where $\partial^{w_{X,s^{(v)}}}$ is a weighting factor and $w_{X,s^{(v)}} = \left(\left|\left\|X_1^L\right\| - \left\|s^{(v)}\right\|\right|\right) \cdot \left\|s^{(v)}\right\|^{-1}$. The weighting factor of Eq. (7) has two features: one is length correlation normalization, and the other is exponential score normalization. For length correlation normalization, the tendency to choose a template $\left\|s^{(v)}\right\|$ with the same length difference of $\left\|X_1^L\right\|$ but smaller length multiplication is eliminated. With exponential score normalization, when the difference between the speech input and each template is larger, a higher dissimilarity score is obtained and spotting discrimination improves. Finally, the normalized measured dissimilarity is determined as follows:

$$d_G^{(v)} = d_G^{(v)} \cdot \partial^{w_{X,s^{(v)}}}. \tag{8}$$

The experimental analysis shown in Fig. 3 indicates that the interval ∂ that yields the most accurate dissimilarity measurement is $[1.2-\delta, 1.2+\delta]$. Therefore, the value of ∂ chosen here is 1.2. The weighting factor is determined using the feature models of the first speaker for inside training. The feature models are different from the test data; thus, ∂ is a test-independent weighting factor. After all the templates are ranked, the retrieval accuracy is estimated using the criterion that the intention of the source speech is located in the set of the best N retrieved translation templates.

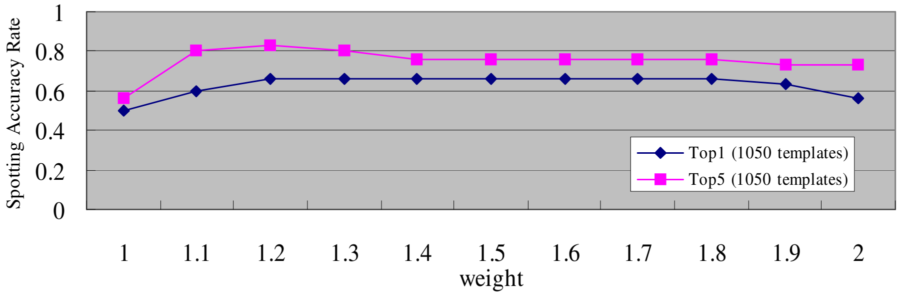


Figure 3. Time-conditioned weight convergence for dissimilarity measurement

4.3 Smoothing the Hypothesized Template

The main weakness with the one-stage algorithm for multiple-translation spotting is that it provides no mechanism for controlling the resulting sequence length, that is, for determining the optimal token sequence of arbitrary length. The algorithm finds a single best path whose sequence length is arbitrary. Therefore, the hypothesized token sequence generally includes noise-like components. The components should be in the form of duplications, and their durations should be below a threshold. Based on this assumption, hypothesized token outputs with segmented durations below the threshold are considered for further smoothing. With Mandarin and Taiwanese, the duration of a syllable is 0.3 sec on average [Sher *et al.*, 1999], and this value is set as the relevant threshold to sift out noise-like components whose durations are less than 0.3 sec. These are the preliminary speaker-dependent results of our experiments. This system is able to adjust the threshold when a speaker speaks at different rates. Additionally, this system is corpus-specific, and out of vocabulary (OOV) words are rejected based on their high dissimilarity scores. After the token sequences of all the TopN templates have been smoothed, the hypothesized target sequences is generated using the translation template with the maximum number of spotting tokens of speech input.

4.4 Target Speech Generation

Once the hypothesized target sequences have been determined, the target speech generation process is straightforward, similar to the waveform segment concatenation-based synthesis method. In this method, waveform segments are extracted beforehand from the recorded intention synthesis units and variable synthesis units of the synthesis template, and they are rearranged with adequate overlapping portions to generate speech with the desired energy and duration. The merits of the method are the small computational cost in the synthesis process and the high level of intelligibility of the synthesized speech. The generation process includes complete matching, waveform replacement, and waveform deletion; thus, it is similar to the example-base translation method [J. Liu and L. Zhou, 1998].

5. Experimental Results

5.1 The Task and the Corpus

We built a collection of Mandarin sentences and their Taiwanese translations that usually appear in phrasebooks for foreign tourists. Because the translations were made sentence by sentence, the corpus was sentence-aligned at birth. *Table 2* shows the basic characteristics of the collected corpus.

Table 2. Basic characteristics of the collected translated examples.

	Mandarin	Taiwanese
Number of sentences	2,084	2,084
Total number of words	14,219	14,317
Number of word entries	6,278	6,291
Average number of words per sentence	6.82	6.87

In this work, the content of the high divergent example sentence pairs needed to be collated or sieved out to improve the accuracy and effectiveness of alignment exploration between word sequences and the derivation of multiple translation templates. *Table 3* shows the basic characteristics of the derived multiple translation templates. The derived templates were used to develop the speech corpus, which was used to construct spotting models and synthesis templates.

Table 3. Basic characteristics of the derived translation templates.

Number of templates	1,050
Number of intentions	1,050
Total number of translation patterns	5,542
Number of translation entries	1,260
Average number of translations per template	5.28

In order to evaluate the system performance, a collection of 1,050 utterances were speaker-dependent trained, and 30 additional utterances of each language were collected by using one male speaker (Sp1) for inside testing and by using two bilingual male speakers (Sp2 and Sp3) for outside testing. All the utterances were sampled at an 8 kHz sampling rate with 16-bit precision on a Pentium® IV 1.8GHz, 1GB RAM, Windows® XP PC.

5.2 Translation Evaluations

For the speech translation system, we found that the recognition performance of 39-dimension MFCCs and 10-dimension LPCCs was close. Therefore, we adopted 10-dimension LPCCs due to their advantages of faster operation and simpler hardware design. Speech feature analysis of recognition was performed using 10 linear prediction coefficient cepstrums (LPCCs) on a 32ms frame that overlapped every 8ms.

For estimating the computational load of the proposed MTS algorithm, a complexity analysis is shown in *Table 4*. Parts of the overall computation of the local frame distance depend on the feature dimension, so we used O(LPCC_add) and O(LPCC_mul) to represent the complexity of additions and multiplications, respectively. We applied Itakura type in each internal dynamic programming path selection employed 3 additions to decide the last node and 1 addition to accumulate the node distance, and 3 multiplications for slope weighting. In *Table 4*, the second row, *Distance computation*, presents the computational complexity of computing the local distance, and the third row, *Path selection*, presents the computational complexity of selecting the best path, that is, the computational overload of MTS for each template.

Table 4. Complexity analysis of the MTS algorithm.

	Computational Load	
	Addition	Multiplication
Distance computation	$L\cdot\sum_{j=1}^{J}K_j^{(v)}\cdot O(LPCC\_add)$	$L\cdot\sum_{j=1}^{J}K_j^{(v)}\cdot O(LPCC\_mul)$
Path selection	$5\cdot L\cdot\sum_{j=1}^{J}K_j^{(v)}$	$3\cdot L\cdot\sum_{j=1}^{J}K_j^{(v)}$
Total for each template	$L\cdot\sum_{j=1}^{J}K_j^{(v)}\cdot(5+O(LPCC\_add))$	$L\cdot\sum_{j=1}^{J}K_j^{(v)}\cdot(3+O(LPCC\_mul))$
Total for all templates	$\sum_{v}\left(L\cdot\sum_{j=1}^{J}K_j^{(v)}\cdot(5+O(LPCC\_add))\right)$	$\sum_{v}\left(L\cdot\sum_{j=1}^{J}K_j^{(v)}\cdot(3+O(LPCC\_mul))\right)$

When input speech is being spotted, a major sub-problem in speech processing is determining the presence or absence of a voice component in a given signal, especially the beginnings and endings of voice segments. Therefore, the energy-based approach, which is a classic one and works well under high SNR conditions, was applied to eliminate unvoiced components in this research. The measurement results were divided into four parts: the dissimilarity measurement of linear prediction coefficient cepstrum (LPCC)-based (baseline), the baseline with unvoiced elimination (unVE), the baseline with the time-conditioned weight (TcW), and the combination of unVE and TcW considerations with the baseline. A given translation template is called a *match* when it contained the same intention as the speech input. The reason for adopting this strategy was that variables could be confirmed again while a dialogue was being processed, while wrong intentions could cause endless iterations of dialogue. The experimental results for proper template spotting are shown in *Table 5 and Table 6*.

Based on the constructed translation templates, when the template or vocabulary size increases, more templates would possibly lead to more feature models and more similarities in speech recognition, thus causing false recognition results and lower spotting accuracy. Additionally, multiple speaker dependent results were obtained using three speakers. The first speaker's feature models (spotting models) were used to perform tests on the other two speakers, and the results are shown in Table 7. The experimental results show that although the feature models were trained by Sp1, the spotting accuracy of Sp2 and Sp3 was only reduced by 10 to 15 percent.

A bilingual evaluator was used to classify the target generation results into three categories [Yamabana *et al*., 2003]: *Good*, *Understandable*, and *Bad*. A *Good* generation needed to have no syntactic errors, and its meaning had to be correctly understood. *Understandable* generations could have some variable translation errors, but the main intention of the source speech had to be conveyed without misunderstanding. Otherwise, the translations were classified as *Bad*. With this subjective measure, the percentage of *Good* or *Understandable* generations for the Top 5 was 80% for Mandarin to Taiwanese (M/T) translation and 76% for Taiwanese to Mandarin (T/M) translation. The percentage of *Good* generations for the Top 1 was 63% for M/T translation, and it was 60% for T/M translation. We examined the translation templates in a specific domain and found that 100% translation accuracy could be achieved. In other words, translation errors occurred only as a result of speech recognition errors, such as word recognition errors and segmentation errors. The results show that T/M had poorer performance than M/T. This is perhaps because spoken Taiwanese has more tones than Mandarin; thus, it is harder for T/M translation spotting to find an appropriate translation template.

Table 5. Average accuracy of baseline spotting and the improvement in Mandarin-to-Taiwanese Translation.

	1		2		3		4	
Template Size	Baseline		Baseline + unVE		Baseline + TcW		Baseline + unVE +TcW	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
150	0.5	0.63	0.6	0.83	0.63	0.83	0.76	1
250	0.5	0.63	0.6	0.83	0.63	0.83	0.76	1
350	0.46	0.6	0.56	0.8	0.6	0.8	0.73	0.96
450	0.46	0.6	0.56	0.8	0.6	0.8	0.73	0.96
550	0.43	0.6	0.56	0.76	0.6	0.76	0.7	0.93
650	0.43	0.56	0.53	0.73	0.56	0.76	0.7	0.93
750	0.43	0.5	0.53	0.73	0.56	0.73	0.7	0.9
850	0.4	0.5	0.5	0.7	0.53	0.73	0.66	0.86
950	0.4	0.46	0.5	0.7	0.5	0.66	0.66	0.83
1050	0.4	0.43	0.46	0.66	0.46	0.66	0.63	0.8

Table 6. Average accuracy of baseline spotting and the improvement in Taiwanese-to-Mandarin Translation.

Template Size	1		2		3		4	
	Baseline		Baseline + unVE		Baseline + TcW		Baseline + unVE +TcW	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
150	0.46	0.6	0.6	0.83	0.6	0.76	0.76	1
250	0.46	0.6	0.6	0.83	0.6	0.7	0.73	0.96
350	0.46	0.56	0.56	0.8	0.56	0.7	0.7	0.96
450	0.43	0.56	0.56	0.76	0.56	0.66	0.7	0.93
550	0.43	0.53	0.53	0.76	0.56	0.66	0.66	0.86
650	0.43	0.53	0.53	0.73	0.53	0.6	0.66	0.86
750	0.4	0.5	0.5	0.7	0.5	0.6	0.63	0.83
850	0.4	0.5	0.5	0.7	0.5	0.56	0.6	0.8
950	0.4	0.46	0.46	0.66	0.46	0.56	0.6	0.76
1050	0.36	0.43	0.43	0.66	0.46	0.56	0.6	0.76

Table 7. Average accuracy of spotting in multiple speaker testing.

			Template Size (Sp1 model)									
		(Top5)	150	250	350	450	550	650	750	850	950	1050
Baseline +unVE +TcW	Sp1	M2T	1	1	0.96	0.96	0.93	0.93	0.9	0.86	0.83	0.8
		T2M	1	0.96	0.96	0.93	0.86	0.86	0.83	0.8	0.76	0.76
	Sp2	M2T	0.9	0.86	0.83	0.8	0.76	0.73	0.73	0.7	0.66	0.66
		T2M	0.86	0.83	0.8	0.76	0.76	0.73	0.7	0.7	0.7	0.66
	Sp3	M2T	0.86	0.83	0.8	0.76	0.73	0.73	0.7	0.66	0.66	0.63
		T2M	0.83	0.8	0.76	0.76	0.73	0.7	0.7	0.66	0.63	0.63

6. Conclusion

In this work, we have proposed an approach that retrieves identified target speech segments by carrying out multiple-translation spotting on a source input. According to the retrieved speech segments, the target speech can be further generated by using the waveform segment concatenation-based synthesis method. Experiments using Mandarin and Taiwanese were performed on Pentium® PCs. The experimental results reveal that our system can achieve an average translation understanding rate of about 78%.

Acknowledgements

This article is a partial result of Project NSC 90-2215-E-006-009, sponsored by the National Science Council, Taiwan, R.O.C.

References

Lavie, A., A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavalda, T. Zeppenfeld and P. Zahn, "JANUS III: Speech-to-Speech Translation in Multiple Languages," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 22(I) 1997, pp. 99–102.

Wahlster, W., "Verbmobil: Foundations of Speech-to-Speech Translation," New York: Springer-Verlag Press, 2000.

Casacuberta, F., D. Llorens, C. Martinez, S. Molau, F. Nevado, H. Ney, M. Pastor, D. Pico, A. Sanchis, E. Vidal and J. M. Vilar, "Speech-to-Speech Translation Based on Finite-State Transducers," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 26(I) 2001, pp. 613–616.

Sugaya, F., T. Takezawa, A. Yokoo and S. Yamamoto, "End-to-End Evaluation in ATR-MATRIX: Speech Translation System between English and Japanese," *Proceedings of European Conference on Speech Communication and Technology*, 6(I) 1999, pp. 2431–2434.

Macklovitch, E., M. Simard and P. Langlais, "TransSearch: A Free Translation Memory on the World Wide Web," *Proceedings of International Conference on Language Resources & Evaluation*, 3(I) 2000, pp. 1201–1208.

Michel, S., "Translation Spotting for Translation Memories," *Proceedings of HLT-NAACL Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, 2003, pp. 65–72.

Véronis, J. and P. Langlais, "Evaluation of Parallel Text Alignment Systems – The ARCADE Project," in J. Véronis (ed.): *Parallel Text Processing*. Dordrecht: Kluwer Academic, 2000, pp. 369–388.

Dorr, B. J., "Machine Translation: A View from the Lexicon," The MIT press, 1993.

Wang, J. F., B. Z. Houg and S. C. Lin, "A Study for Mandarin Text to Taiwanese Speech System," *Proceedings of the 12th Research on Computational Linguistics Conference*, 1999, pp. 37–53.

Sher, Y. J., K. C. Chung and C. H. Wu, "Establish Taiwanese 7-Tones Syllable–based Synthesis Units Database for the Prototype Development of Text-to-Speech System," *Proceedings of the 12th Research on Computational Linguistics Conference*, 1999, pp. 15–35.

Liu, J. and L. Zhou, "A Hybrid Model for Chinese-English Machine Translation," *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 2(I) 1998, pp.1201–1206.

Yamabana, K., K. Hanazawa, R. Isotani, S. Osada, A. Okumura and T. Watanabe, "A Speech Translation System with Mobile Wireless Clients," *Proceedings of the Student Research Workshop at the 41st Annual Meeting of the Association for Computational Linguistics*, 41(II) 2003, pp. 119–122.

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 29-44

Latent Semantic Language Modeling and Smoothing

Jen-Tzung Chien[*], Meng-Sung Wu[*] and Hua-Jui Peng[*]

Abstract

Language modeling plays a critical role for automatic speech recognition. Typically, the *n*-gram language models suffer from the lack of a good representation of historical words and an inability to estimate unseen parameters due to insufficient training data. In this study, we explore the application of latent semantic information (LSI) to language modeling and parameter smoothing. Our approach adopts latent semantic analysis to transform all words and documents into a common semantic space. The word-to-word, word-to-document and document-to-document relations are, accordingly, exploited for language modeling and smoothing. For language modeling, we present a new representation of historical words based on retrieval of the most relevant document. We also develop a novel parameter smoothing method, where the language models of seen and unseen words are estimated by interpolating the *k* nearest seen words in the training corpus. The interpolation coefficients are determined according to the closeness of words in the semantic space. As shown by experiments, the proposed modeling and smoothing methods can significantly reduce the perplexity of language models with moderate computational cost.

Keywords: language modeling, parameter smoothing, speech recognition, and latent semantic analysis.

1. Introduction

Language models have been successfully developed for speech recognition, optical character recognition, machine translation, information retrieval, etc. Many studies in the field of speech recognition have focused on this topic [Jelinek 1990, Jelinek 1991]. As shown in Figure 1, a speech recognition system is composed of syllable-level and word-level matching processes, in which the acoustic model λ and language model τ are applied, respectively. In theory, the speech recognition procedure combines the acoustic model and language model according to the Bayes rule. Let O denote the acoustic data, and let $W = \{w_1, \mathbf{L}, w_l\} = w_1^l$ denote a string of l

[*] Department of Computer Science and Information Engineering , National Cheng Kung University, Tainan, Taiwan, ROC
E-mail: jtchien@mail.ncku.edu.tw

words. The speech recognition task aims to find the most likely word string $\hat{W}$ by maximizing the *a posteriori* probability given the observed acoustic data *O*:

$$\hat{W} = \arg\max_{W} P(W|O) = \arg\max_{W} P_{\lambda}(O|W)P_{\tau}(W), \tag{1}$$

where $P_{\tau}(W)$ is the *a priori* probability of the occurring word string *W*, and $P_{\lambda}(O|W)$ is the probability of observing data *O* given the word string *W*. The parameters τ and λ are the language model and speech hidden Markov models (HMM's), respectively. Hereafter, we will neglect the notation τ in $P_{\tau}(W)$. The language model $\Pr(W)$ aims to measure the probability of word occurrence. This model is employed to predict the word occurrence given the history words. In an *n*-gram model, we assume that the probability of a word depends only on the preceding *n*-1 words. The *N*-gram model $\Pr(W)$ is written as

$$\Pr(W) = \Pr(w_1,..., w_l) = \prod_{q=1}^{l} \Pr(w_q|w_1, w_2,..., w_{q-1}) \cong \prod_{q=1}^{l} \Pr(w_q|w_{q-n+1}^{q-1}). \tag{2}$$

The sequence $H_q = \{w_1, \mathbf{L}, w_{q-1}\}$ is referred to as the *history* H_q for word w_q. To estimate $\Pr(w_q|w_{q-n+1}^{q-1})$, we can count the number of words w_q following the history words w_{q-n+1}^{q-1} and divide it by the total number of occurring history words w_{q-n+1}^{q-1}, i.e.,

$$\Pr(w_q|w_{q-n+1}^{q-1}) = \frac{c(w_{q-n+1}^{q})}{\sum_{w_i} c(w_{q-n+1}^{q})}. \tag{3}$$

This probability estimation is called the *maximum likelihood estimation* (MLE). The bigram model $\Pr(W) \cong \prod_{q=1}^{l} \Pr(w_q|w_{q-1})$ and trigram model $\Pr(W) \cong \prod_{q=1}^{l} \Pr(w_q|w_{q-2}, w_{q-1})$ are employed in most speech recognition systems. However, when a word sequence (w_{q-2}, w_{q-1}, w_q) is not occurs in the training data, the trigram model $\Pr(w_q|w_{q-2}, w_{q-1})$ could not be estimated. We may apply parameter smoothing to find the unseen trigram model. In the literature, several smoothing methods have been proposed to deal with the data sparseness problem [Katz 1987, Kawabata and Tamoto 1996, Lau *et al.* 1993, Zhai and Lafferty 2001]. Also, *maximum a posteriori* adaptation of the language model has been presented to resolve the problem of domain mismatch between training and test corpora [Bellegarda 2000a, Federico 1996, Masataki *et al.* 1997]. Besides the problems of data sparseness and domain mismatch, the *n*-gram model is inferior in terms of characterizing long-distance word relationships. For example, the trigram model is unable to characterize word dependence beyond the span of three successive words. In [Lau *et al.* 1993, Zhou and Lua 1999], the trigram model was improved by extracting word relationships from the document history. This approach was exploited to search the trigger pair, $w_A \rightarrow w_B$, where the appearance of w_A in the document history significantly affects the probability of occurring w_B. The trigger pairs provide long distance information because the triggering and triggered words might be separated by several words. However, trigger pair selection neglects the possibility of low-frequency word triggers, which might contain useful semantic information. The LSA method was developed to resolve this problem.

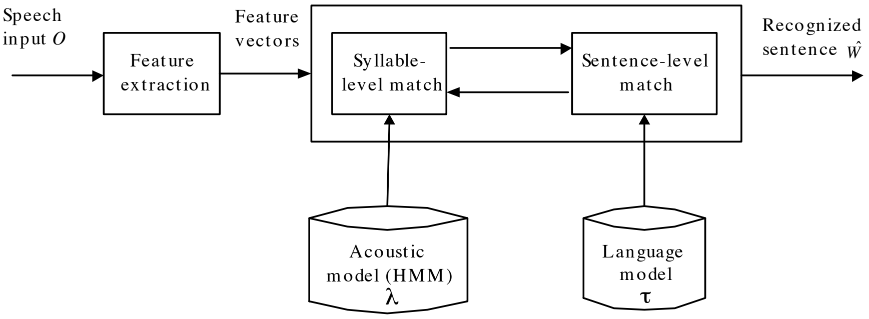


Figure 1. A schematic diagram of a speech recognition system.

In this paper, a new language modeling and smoothing method is proposed based on the framework of latent semantic analysis (LSA). The traditional n-gram model is weak in terms of characterizing the information in historical words. This weakness is compensated for herein by using the LSA framework, where word-to-word, word-to-document and document-to-document similarities are found in the semantic space. With the use of LSA, all the words are mapped to a common semantic space, which is constructed via the *singular value decomposition* (SVD) of a word-by-document matrix. Bellegarda [1998, 2000a, 2000b] applied the LSA framework to the n-gram model such that the resulting word error rate and perplexity were substantially reduced. The LSA representation of the history suffers from a drawback in that the representation of the history carries insufficient information at the beginning of a text document. To overcome this problem, we propose a relevance retrieval framework to represent the history. For language model smoothing, we estimate unseen language models by using the seen models corresponding to the k nearest neighbor words. Because this smoothing method extracts synonym and semantic information, it can be also referred to as "semantic smoothing." In the following section, we briefly introduce the framework of LSA. Section 3 addresses the proposed language modeling and smoothing approaches. The LSA framework is applied to relevance feedback language modeling and k nearest neighbor language smoothing. Section 4 describes the experimental setup and reports the results for the perplexity and computational cost. Finally, we draw conclusions in Section 5.

2. Latent semantic analysis

In the literature [Berry *et al.* 1995, Deerwester *et al.* 1990, Ricardo and Berthier 2000], latent semantic analysis (LSA) has been widely applied to vector space based information retrieval. During the past few years, LSA has also been applied to language model adaptation [Bellegarda 1998, Bellegarda 2000a, Novak and Mammone 2001]. Latent semantic analysis is a dimension reduction technique that projects the query and document into a common semantic space [Deerwester *et al.* 1990, Ding 1999]. This projection reduces the document vector from a high dimensional space to a low dimensional space, which is referred as the latent semantic space. The goal is to represent similar documents as close points in the latent semantic space, based on an appropriate metric. This metric can capture the significant associations between words and documents. Given an $M \times N$ matrix **A**, with M terms and N documents, $M \geq N$ and rank (**A**) = R. The weighted count $a_{i,j}$ of matrix **A** is the number of occurrences of each word w_i in a document d_j, calculated as follows:

$$a_{i,j} = (1 - \varepsilon_i)\frac{c_{i,j}}{n_j} \cdot \tag{4}$$

Here, $c_{i,j}$ is the number of terms w_i occurring in document d_j, n_j is the total number of words in d_j, and ε_i is the normalized entropy of w_i in the collection of data consisting of N documents, i.e.,

$$\varepsilon_i = -\frac{1}{\log N}\sum_{j=1}^{N}\frac{c_{i,j}}{t_i}\log\frac{c_{i,j}}{t_i}, \tag{5}$$

where $t_i = \sum_j c_{i,j}$ is the total number of times w_i occurs in the collection of data. A value of ε_i that is close to one occurs in case of $c_{i,j} = t_i/N$. This means that the word w_i is distributed across many documents throughout the corpus. A value of ε_i that is close to zero, i.e., the case in which $c_{i,j} = t_i$, indicates that the word w_i is present in only a few documents. Hence, in (4), $1-\varepsilon_i$ represents a global indexing weight for the word w_i, and $c_{i,j}/n_j$ indicates that the word w_i occurs in frequently in document d_j.

Latent semantic analysis is a conceptual-indexing method, which uses singular value decomposition (SVD) [Berry *et al.* 1995, Golub and Van Loan 1989] to find the latent semantic structure of word to document association. SVD decomposes the matrix **A** into three sub-matrices:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{6}$$

where **U** and **V** are orthogonal matrices, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_R$, and Σ is a diagonal matrix. As shown in Figure 2, the first R columns of **U** and **V,** and the first R diagonal elements of Σ can be used to approach **A** with $\text{rank}(\mathbf{A}) = R$ by means of $\mathbf{A}_R = \mathbf{U}_R\Sigma_R\mathbf{V}_R^T$, where $\mathbf{A}_R$ is a representative matrix **A**. The result of SVD is a set of vectors representing the location of each term and document in the reduced R-dimensional LSA space [Berry 1992]. For a given training corpus, $\mathbf{AA}^T$ characterizes all the co-occurrences between words, and $\mathbf{A}^T\mathbf{A}$ characterizes all the co-occurrences between documents. That is, a similar pattern of occurring words w_i and w_j can be inferred from the (i,j) cell of $\mathbf{AA}^T$, and a similar pattern of words contained in documents d_i and d_j can be inferred from the (i,j) cell of $\mathbf{A}^T\mathbf{A}$ [Bellegarda 1998, Bellegarda 1997, Bellegarda 2000a, Chen and Goodman 1999]. This LSA approach performs well when a major portion of the meaningful semantic structure [Deerwester *et al.* 1990] is captured.

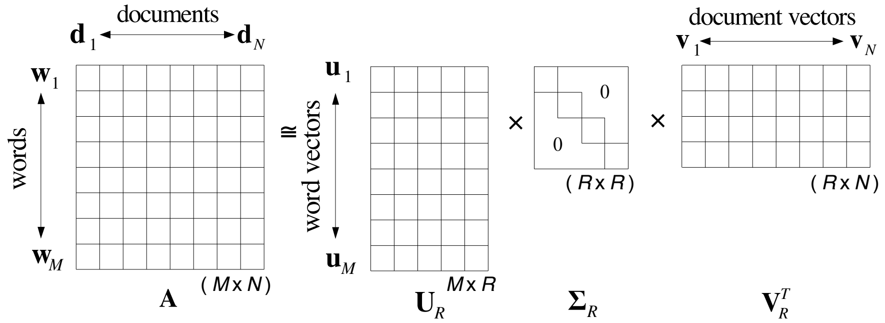


Figure 2. A diagram of the truncated SVD.

3. New language modeling and smoothing techniques

3.1 LSA Parameter Modeling

N-gram language models are useful for modeling the local dependencies of word occurrences but not for capturing global word dependencies. The modeling process leads to the estimation of the conditional probability $\Pr(w_q | w_{q-n+1}^{q-1})$, which characterizes the linguistic regularity in a span of *n* words. When the window size *n* is limited, the *n*-gram is weak in terms of capturing long distance dependencies. Long distance correlation between words is commonly found in language and is caused by closeness in meaning; e.g., the words "stock" and "fund" are both likely to occur in financial news. To deal with long distance modeling, the LSA approach can be applied to extract large span semantic knowledge. Our motivation lies in the fact that there exists some latent structure in the occurrence patterns of words across documents. Hence, the *n*-gram language model can be improved by employing LSA to perform large span prediction of word occurrence.

Let the word w_q denote the predicted word, let H_{q-1} denote the history for w_q, and let $\Pr(w_q | H_{q-1})$ be the associated language model probability. Using the *n*-gram language model, we find that $H_{q-1} = \{w_{q-1}, w_{q-2}, \mathbf{L}, w_{q-n+1}\}$ is the relevant history composed of the preceding *n*-1 words. The LSA language model is expressed by

$$\Pr(w_q|H_{q-1}) = \Pr(w_q|H_{q-1}, S) = \Pr(w_q|\mathbf{d}_{q-1}), \tag{7}$$

where the conditioning on S reflects the fact that the probability depends on the particular vector space arising from the SVD representation, and where $\Pr(w_q|\mathbf{d}_{q-1})$ is computed directly based on the closeness of w_q and $\mathbf{d}_{q-1}$ in the semantic space S. The vector $\mathbf{d}_{q-1}$ can be viewed as an additional *pseudodocument* vector for matrix **A** [Bellegarda 1998, Bellegarda 2000a, Bellegarda 2000b]. The representation $\mathbf{v}_{q-1}$ for the pseudodocument vector $\mathbf{d}_{q-1}$ in the space S is given by

$$\mathbf{v}_{q-1} = \mathbf{d}_{q-1}^T \mathbf{U}\Sigma^{-1}. \tag{8}$$

By referring to (4), we can obtain the pseudodocument vector $\mathbf{d}_q$ recursively in the LSA space via [Bellegarda 2000a, Bellegarda 2000b]

$$\mathbf{d}_q = \frac{n_q - 1}{n_q}\mathbf{d}_{q-1} + [0 \ldots 0 \frac{1-\varepsilon_q}{n_q} 0 \ldots 0]^T. \tag{9}$$

To clarify (8) and (9), we provide their derivations in the Appendix.

However, at the beginning of a text document, it is difficult to capture long distance word dependencies for calculating $\Pr(w_q|\mathbf{d}_{q-1})$ due to the shortness of the history H_{q-1}. To overcome this weakness, we present here a new method for estimating the pseudodocument vector $\mathbf{d}_{q-1}$. *Our method aims to retrieve the most likely relevance document* $\hat{\mathbf{d}}_{q-1}$ *from the training documents* $\mathbf{d}_1, \ldots, \mathbf{d}_N$ *so as to represent the pseudodocument vector* $\mathbf{d}_{q-1}$. The LSA probability $\Pr(w_q|\mathbf{d}_{q-1})$ is replaced by $\Pr(w_q|\hat{\mathbf{d}}_{q-1})$. Accordingly, the pseudodocument $\hat{\mathbf{d}}_{q-1}$ is estimated by

$$\hat{\mathbf{d}}_{q-1} = \arg\max_{\mathbf{d}_i} \Pr(\mathbf{d}_i|\mathbf{d}_{q-1}), \qquad i = 1, \ldots, N. \tag{10}$$

Here, $\mathbf{d}_{q-1}$ is obtained recursively from (9). The probability $\Pr(\mathbf{d}_i|\mathbf{d}_{q-1})$ is determined by finding the cosine of the angle between the vectors $\mathbf{d}_i$ and $\mathbf{d}_{q-1}$ in the latent semantic space; i.e., by using the vectors $\mathbf{v}_i\Sigma$ and $\mathbf{v}_{q-1}\Sigma$ in

$$\Pr(\mathbf{d}_i|\mathbf{d}_{q-1}) = \cos(\mathbf{v}_i\Sigma, \mathbf{v}_{q-1}\Sigma) = \frac{\mathbf{v}_i\Sigma^2\mathbf{v}_{q-1}^T}{\|\mathbf{v}_i\Sigma\|\,\|\mathbf{v}_{q-1}\Sigma\|}. \tag{11}$$

When q is increased, the most likely document vector $\hat{\mathbf{d}}_{q-1}$ moves around in the LSA space. Assuming that $\hat{\mathbf{d}}_{q-1}$ is semantically homogeneous, we can expect the resulting trajectory to eventually settle down in the vicinity of the document cluster corresponding to the closest semantic content.

In this study, the LSA language model is exploited by integrating the effects of histories obtained from the conventional n-gram component $H_{q-1}^{(n)} = \{w_{q-1}, w_{q-2}, \ldots, w_{q-n+1}\}$ and the LSA component $H_{q-1}^{(l)} = \hat{\mathbf{d}}_{q-1}$ [Bellegarda 1998, Bellegarda 2000a]. The new language model is written as

$$
\begin{aligned}
\Pr(w_q|H_{q-1}) &= \Pr(w_q|H^{(n)}_{q-1}, H^{(l)}_{q-1}) = \frac{\Pr(w_q, H^{(l)}_{q-1}|H^{(n)}_{q-1})}{\sum_{w_i} \Pr(w_i, H^{(l)}_{q-1}|H^{(n)}_{q-1})} \\
&= \frac{\Pr(w_q|H^{(n)}_{q-1}) \Pr(H^{(l)}_{q-1}|w_q, H^{(n)}_{q-1})}{\sum_{w_i} \Pr(w_i|H^{(n)}_{q-1}) \Pr(H^{(l)}_{q-1}|w_i, H^{(n)}_{q-1})} \\
&= \frac{\Pr(w_q|w_{q-1}, w_{q-1}, \ldots, w_{q-n+1}) \Pr(\hat{\mathbf{d}}_{q-1}|w_q)}{\sum_{w_i} \Pr(w_i|w_{q-1}, w_{q-1}, \ldots, w_{q-n+1}) \Pr(\hat{\mathbf{d}}_{q-1}|w_i)} \\
&= \frac{\Pr(w_q|w_{q-1}, w_{q-1}, \ldots, w_{q-n+1}) \dfrac{\Pr(w_q|\hat{\mathbf{d}}_{q-1})}{\Pr(w_q)}}{\sum_{w_i} \Pr(w_i|w_{q-1}, w_{q-1}, \ldots, w_{q-n+1}) \dfrac{\Pr(w_i|\hat{\mathbf{d}}_{q-1})}{\Pr(w_i)}} .
\end{aligned}
\tag{12}
$$

In (12), we assume that $\Pr(H^{(l)}_{q-1}|w_q, H^{(n)}_{q-1}) = \Pr(\hat{\mathbf{d}}_{q-1}|w_q)$. The probability $\Pr(w_q|\hat{\mathbf{d}}_{q-1})$ is computed based on the representations of w_q and $\hat{\mathbf{d}}_{q-1}$ in the semantic space S, which are provided by $\mathbf{u}_q\Sigma^{1/2}$ and $\hat{\mathbf{v}}_{q-1}\Sigma^{1/2}$, respectively. The LSA probability is calculated as follows:

$$\Pr(w_q|\hat{\mathbf{d}}_{q-1}) = \cos(\mathbf{u}_q\Sigma^{1/2}, \hat{\mathbf{v}}_{q-1}\Sigma^{1/2}) = \frac{\mathbf{u}_q\Sigma\hat{\mathbf{v}}^T_{q-1}}{\left\|\mathbf{u}_q\Sigma^{1/2}\right\| \left\|\hat{\mathbf{v}}_{q-1}\Sigma^{1/2}\right\|} . \tag{13}$$

3.2 LSA Parameter Smoothing

In the real world, a training corpus is not sufficient to estimate the n-gram model for all word occurrences $\{w_{q-n+1}, \ldots, w_{q-1}, w_q\}$. To overcome the problem of insufficient data, the parameter smoothing method can be used to estimate the joint probabilities of unseen word occurrences and, simultaneously, smooth those of seen word occurrences in the training corpus. It is common to interpolate the n-gram and (n-1)-gram for the purpose of language model smoothing. Jelinek-Mercer smoothing [Jelinek and Mercer 1980] is represented as follows:

$$\Pr_{JM}(w_q|w^{q-1}_{q-n+1}) = \lambda_q \Pr(w_q|w^{q-1}_{q-n+1}) + (1-\lambda_q)\Pr_{JM}(w_q|w^{q-1}_{q-n+2}) . \tag{14}$$

The smoothed n-gram model $\Pr_{JM}(w_q|w^{q-1}_{q-n+1})$ is defined recursively as a linear interpolation between the maximum likelihood n-gram model $\Pr(w_q|w^{q-1}_{q-n+1})$ and the smoothed (n-1)-gram model $\Pr_{JM}(w_q|w^{q-1}_{q-n+2})$. This smoothing process is intended to flatten the probability distribution. Let N_q denote the number of occurrences for word w_q preceding w^{q-1}_{q-n+1}:

$$N_q = \left|\{w_q : c(w^{q-1}_{q-n+1} w_q) > 0\}\right| . \tag{15}$$

The well-known Witten-Bell smoothing approach [Written and Bell 1991] incorporates the interpolation coefficient

$$1-\lambda_q = \frac{N_q}{N_q + \sum_{w_q} c(w_{q-n+1}^q)}, \tag{16}$$

into (14) of Jelinek-Mercer smoothing to generate

$$\Pr_{WB}(w_q|w_{q-n+1}^{q-1}) = \frac{\sum_{w_q} c(w_{q-n+1}^q) + N_q \Pr_{WB}(w_q|w_{q-n+2}^{q-1})}{N_q + \sum_{w_q} c(w_{q-n+1}^q)}. \tag{17}$$

In this paper, we will present a novel smoothing method in which the language models of seen and unseen word occurrences are estimated by interpolating the LSA language model of a word occurrence and of the k nearest word occurrences. Let us consider the words "car," "automobile," "driver," and "elephant". "Car" and "automobile" are synonyms. "Driver" is related and "elephant" is unrelated to "car" and "automobile." If the words "car" and "automobile" do not appear in the given documents, we may collect many documents containing related words, e.g., the motor, vehicle, engine, etc. The statistics of these nearest seen words can be used to estimate the language model of the unseen words. When the bigram model is used, the smoothed model $\tilde{\Pr}(w_q|w_{q-1})$ is estimated by interpolating the LSA bigram $\Pr(w_q|w_{q-1})$ of the word pair occurrence (w_q, w_{q-1}) and those of the other k occurrences $(w_q, \hat{w}_j^q)$, $1 \le j \le k$, where the k nearest words $\hat{w}_j^q$ to word w_q are determined according to the LSA probabilities:

$$\Pr(w_q|w_j) = \cos(\mathbf{u}_q\Sigma, \mathbf{u}_j\Sigma) = \frac{\mathbf{u}_q\Sigma^2\mathbf{u}_j^T}{\|\mathbf{u}_q\Sigma\|\|\mathbf{u}_j\Sigma\|}, \quad 1 \le j \le M. \tag{18}$$

The interpolation is performed as follows:

$$\tilde{\Pr}(w_q|w_{q-1}) = \alpha_q \Pr(w_q|w_{q-1}) + (1-\alpha_q)\sum_{j=1}^{k} \beta_j^q \Pr(w_q|\hat{w}_j^q), \tag{19}$$

where the weighting coefficients $\{\beta_j^q, 1 \le j \le k\}$ and the interpolation coefficient α_q are estimated by

$$\beta_j^q = \frac{\Pr(w_q|\hat{w}_j^q)}{\sum_{j=1}^{k} \Pr(w_q|\hat{w}_j^q)} \tag{20}$$

and

$$\alpha_q = \frac{\Pr(w_q|w_{q-1})}{\Pr(w_q|w_{q-1}) + \sum_{j=1}^{k} \beta_j^q \Pr(w_q|\hat{w}_j^q)}, \tag{21}$$

respectively. As seen in (20), the weighting coefficient β_j^q is proportional to the LSA probability of the word pair $(w_q, \hat{w}_j^q)$ and has the property $\sum_{j=1}^{k} \beta_j^q = 1$. That is, the closer the word $\hat{w}_j^q$ is to the current word w_q, the higher is the weighting coefficient that β_j^q produces. Also, it is reasonable to adopt the interpolation coefficient α_q in (21), which is proportional to the closeness between w_q and w_{q-1} in the semantic space. The smoothing method proposed in (19) should be performed when the current word w_q is trained using LSA. Different from the Jelinek-Mercer and Witten-Bell smoothing methods that adopt the maximum likelihood language model, the proposed smoothing technique is combined with the LSA framework, and the probabilities $\Pr(w_q|w_{q-1})$ and $\Pr(w_q|\hat{w}_j^q)$ are computed via the LSA procedure.

4. Experiments

We evaluated the performance of the proposed language model through experiments. Two databases were employed. The first database was the *CKIP* balanced corpus of Modern Chinese (http://godel.iis.sinica.edu.tw), which was collected by Academia Sinica in Taiwan, ROC. Totally, this database has twenty-five million Chinese characters and a vocabulary size of 80,000 words. In addition, we collected 9,372 news documents during 2001 and 2002 from the news websites of CNA (http://www.cna.com.tw), ChinaTimes (http://news.chinatimes.com) and UDNnews (http://www.udnnews.com.tw). We randomly sampled 9,148 documents for training and the remaining 224 documents for testing. The news documents were divided into eight categories, including technology, society, international, leisure, politics, finance, entertainment, and sports news. The numbers of training and testing documents in the eight news categories are listed in Table 1. We chose the most frequent 32,941 words to construct our dictionary. Using the LSA procedure, we built a 32,941*9,148 word by document matrix $\mathbf{A}$ using training data. The SVD algorithm was applied with different numbers of singular values. In this study, we used MATLAB for the SVD operation and compared the performance of LSA language modeling, with the number of singular values R set at 25, 50, 75, and 100.

The measure of perplexity was adopted to evaluate the different language models. The computational costs were reported for comparison. Here, the computation time was measured in minutes by testing 224 documents using a personal computer with a Pentium IV-1.6GHz processor and 256 MB RAM. The bigram model was employed in the experiments.

Table 1. Numbers of training and testing documents for the eight news categories.

	Technology	Social	International	Leisure	Politics	Financial	Entertain	Sports
Training data	289	2,658	330	1,106	1,299	2,605	430	431
Testing data	30	24	23	24	24	49	23	27

4.1 Perplexity

Perplexity is an important parameter used to evaluate the performance of language models. Consider an information source containing of word sequence, $w_1, w_2, \mathbf{K}, w_l$, each of which is chosen from a vocabulary V. The entropy of a source emitting the words $w_1, w_2, \mathbf{K}, w_l$ is defined as

$$E = -\lim_{l \to \infty} \frac{1}{l} \sum_{w_1 w_2 \mathbf{K} w_l} \Pr(w_1, w_2, \mathbf{K}, w_l) \cdot \log \Pr(w_1, w_2, \mathbf{K}, w_l) \cdot \tag{22}$$

If the source is ergodic, the entropy in (22) is equivalent to

$$E = -\lim_{l \to \infty} \frac{1}{l} \log \Pr(w_1, w_2, \mathbf{K}, w_l) \cdot \tag{23}$$

Since the *n*-gram language model is used, *E* can be estimated as follows:

$$\tilde{E} = -\frac{1}{l} \sum_{q=1}^{l} \log \Pr(w_q | w_{q-n+1}^{q-1}) \cdot \tag{24}$$

Given testing documents with *l* words, the perplexity is calculated as follows:

$$Perplexity = 2^{\tilde{E}} . \tag{25}$$

In general, the entropy $\tilde{E}$ is the average difficulty or uncertainty of each word using the language model. The lower measured the perplexity, the better the speech recognition accuracy that can be achieved.

4.2 Evaluation of Different Language Modeling and Smoothing Methods

In the experiments, we evaluated different language modeling and smoothing methods in terms of perplexity and computation time. First of all, we investigated the effect of the SVD dimension in the proposed LSA bigram model. No parameter smoothing was performed. In Figures 3 and 4, we compare the perplexity and computation time for different SVD dimensions. Here, the computation time was a measure of the SVD operation of a 32,941*9,148 word by document matrix $\mathbf{A}$. We found that an SVD dimension of 25 was appropriate for constructing the semantic space. In the subsequent evaluation, the SVD dimension was fixed at 25 for the proposed LSA bigram and LSA smoothing. Next, we examined the effect of the parameter k in the proposed LSA smoothing method. LSA smoothing of seen and unseen bigrams was performed by combining the bigrams corresponding to the k nearest words. In Figure 5, we show the results for perplexity versus the *k* nearest neighbor words when LSA smoothing was applied to the standard bigram and proposed LSA bigram. The values $k = 5$, 10, 30 and 50 were examined. When proposed LSA modeling and smoothing was used, the lowest perplexity of 81 was achieved by using $k = 5$. The perplexity of the standard bigram with LSA smoothing was calculated as 102. We then fixed $k = 5$ in the subsequent comparison experiment.

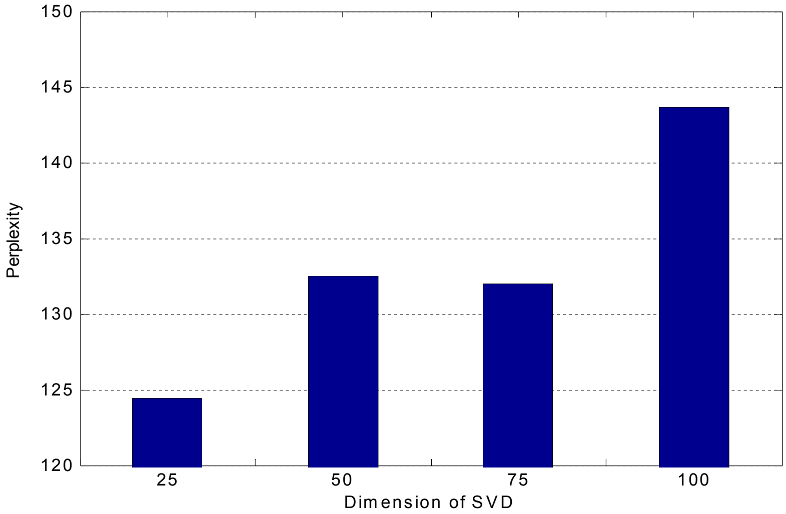


Figure 3. Comparison of perplexity results for different SVD dimensions.

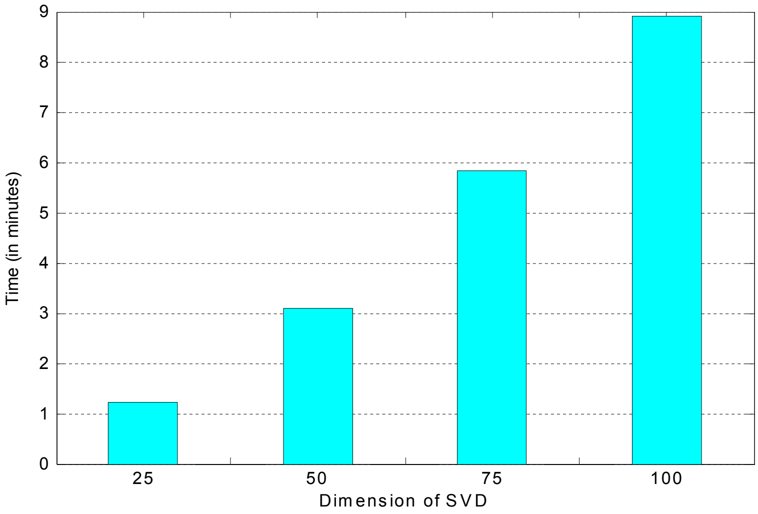


Figure 4. Comparison of computation times for different SVD dimensions.

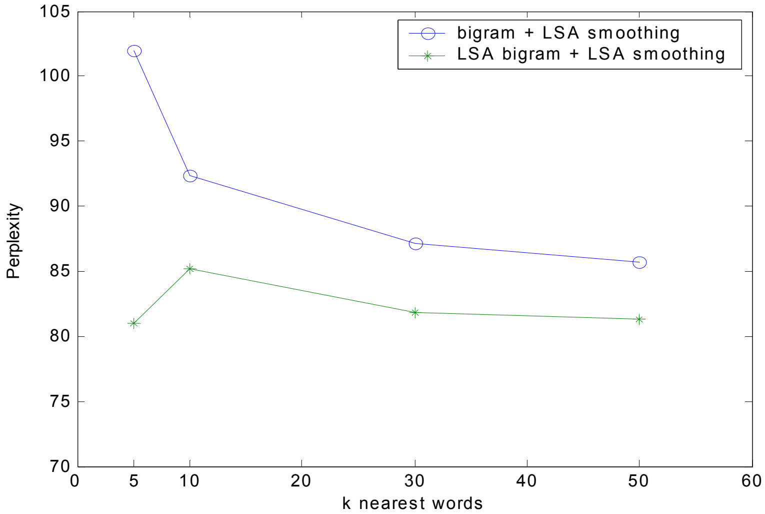


Figure 5. Perplexity versus the k nearest neighbor words when LSA smoothing was applied to the standard bigram and proposed LSA bigram.

Furthermore, different language modeling and smoothing methods were compared, and the results are shown in Table 2. Besides the standard bigram, we implemented Bellegarda's LSA bigram [Bellegarda 1998] and the proposed LSA bigram to evaluate the effect of language modeling. The main difference is that proposed LSA bigram aims to retrieve the most likely relevance document vector in order to represent the historical words. In addition, the language models with and without parameter smoothing were examined. The algorithms of Witten-Bell smoothing and the proposed LSA smoothing were also used for the purpose of comparison. The Witten-Bell smoothed bigram is estimated by interpolating with the corresponding unigram. The proposed LSA smoothing combines the bigrams corresponding to the k nearest seen words in the training corpus. We can see that the baseline bigram model has a perplexity of 158.3. The perplexity was reduced to 128.7 and 124.4 by applying Bellegarda's LSA bigram and proposed LSA bigram, respectively. However, when Witten-Bell smoothing was incorporated, the perplexity is greatly reduced from 158.3 without smoothing to 122.6 with smoothing. When the proposed LSA bigram with Witten-Bell smoothing were used, the perplexity could be improved to 108.7. This indicates the importance of adopting a smoothing algorithm in the language model. Furthermore, when the proposed LSA smoothing was used, the perplexity was reduced to 102, which is better than the perplexity of 122.6 obtained using Witten-Bell smoothing. This is because the Witten-Bell smoothing method estimates the n-gram model by using the (n-1)-gram, while the proposed LSA smoothing approach always adopts nearest n-gram models without using the (n-1)-gram. Among the different combinations, the lowest perplexity of 81 was achieved by applying the proposed LSA bigram with LSA smoothing. Compared to baseline system, the perplexity could be improved by up to 48.8%. The computation times of the different methods were also compared. The results show that the computation overhead of using a smoothing algorithm is slight. The computation load of the LSA bigram is much higher than that of the standard bigram. This result indicates that the smoothing algorithm can lead to greater improvement in perplexity with a lower computation cost than can be achieved by modifying the language model.

Table 2. Comparison of perplexity and computation time for different language modeling and smoothing methods.

Language Model		Perplexity	Reduction Rate (%)	Computation Time (minutes)
Modeling Method	*Smoothing Method*			
Bigram	N/A	158.3	N/A	48.3
Bigram	Witten-Bell Smoothing	122.6	22.6	51.3
Bellegarda’s LSA Bigram	N/A	128.7	18.7	176.7
Proposed LSA Bigram	N/A	124.4	21.4	161.2
Proposed LSA Bigram	Witten-Bell Smoothing	108.7	31.3	163.3
Bigram	LSA Smoothing	102	35.6	52.2
Proposed LSA Bigram	LSA Smoothing	81	48.8	163.4

5. Conclusion

Statistical n-gram modeling is limited in terms of its ability to represent the historical words and estimate the unseen parameters of an inadequate training corpus. In this paper, we have presented new language modeling and smoothing methods that are based on the framework of latent semantic analysis. The concept of relevance retrieval has been adopted in order to exploit a new language modeling approach, where the most likely pseudodocument is retrieved to represent the historical words. The language model is estimated according to the closeness of the current word vector and the historical pseudodocument vector in the common LSA space. To overcome the problem of insufficient training data, we perform LSA smoothing, where the bigram of the current word is computed by interpolating with the bigrams corresponding to the k nearest words. The weighting coefficients of the k nearest words are proportional to the closeness to the current word in the LSA space. From the results of experiments in which Chinese news documents were evaluated, we found that the language modeling performance could be greatly improved by applying the proposed LSA parameter modeling and smoothing algorithms. The proposed methods outperformed Bellegarda's LSA bigram and Witten-Bell smoothing. Compared to the baseline bigram model, the perplexity was reduced by up to 48.8%. Also, the perplexity improvement and computation efficiency that could be achieved through parameter smoothing were better than that which could be achieved through parameter modeling. This approach can be easily extended to the trigram model and other languages. In the future, we will explore theoretical rules for determining the SVD dimension for LSA. We will also investigate the effect of the amount of training data on the LSA framework. We are currently applying the proposed language model to information retrieval and large vocabulary continuous speech recognition.

Appendix

Derivations of Equations (8) and (9)

In (8), the pseudodocument vector $\mathbf{d}_{q-1}$ is the (q-1)th column vector of matrix $\mathbf{A}$. From SVD, we know $\mathbf{d}_{q-1} = \mathbf{U}\Sigma\mathbf{v}_{q-1}^T$. Because $\mathbf{U}$ is orthogonal and Σ is diagonal, the representation $\mathbf{v}_{q-1}$ in semantic space S is obtained by

$$\mathbf{d}_{q-1} = \mathbf{U}\Sigma\mathbf{v}_{q-1}^T \Rightarrow \mathbf{v}_{q-1}^T = \Sigma^{-1}\mathbf{U}^T\mathbf{d}_{q-1} \Rightarrow \mathbf{v}_{q-1} = (\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{d}_{q-1})^T = \mathbf{d}_{q-1}^T\mathbf{U}\Sigma^{-1}\,. \tag{26}$$

Also, from (4), we can derive the recursive formula for $a_{i,q}$ corresponding to word w_i and document d_q

$$\begin{aligned} a_{i,q} &= (1-\varepsilon_i)\frac{c_{i,q}}{n_q} = (1-\varepsilon_i)\frac{c_{i,q-1}+1}{n_q} = (1-\varepsilon_i)\frac{c_{i,q-1}}{n_q} + \frac{1-\varepsilon_i}{n_q} \\ &= (1-\varepsilon_i)\frac{c_{i,q-1}}{n_{q-1}}\cdot\frac{n_{q-1}}{n_q} + \frac{1-\varepsilon_i}{n_q} = a_{i,q-1}\cdot\frac{n_{q-1}}{n_q} + \frac{1-\varepsilon_i}{n_q} \end{aligned}\,. \tag{27}$$

By extending this formula using vector representation, we obtain (9) by

$$\mathbf{d}_q = \frac{n_q-1}{n_q}\mathbf{d}_{q-1} + \frac{1-\varepsilon_q}{n_q}\cdot[0\ldots010\ldots0]^T = \frac{n_q-1}{n_q}\mathbf{d}_{q-1} + [0\ldots0\frac{1-\varepsilon_q}{n_q}0\ldots0]^T\,, \tag{28}$$

where the "1" appears at coordinate i in the above vector.

Acknowledgment

The authors thank the anonymous reviewers for providing valuable comments, which considerably improved the quality of this paper.

References

Bellegarda, J. R., "A Multi-span Language Modeling Framework for Large Vocabulary Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, 6(5) 1998, pp. 456-467.

Bellegarda, J. R., "A statistical language modeling approach integrating local and global constraints," *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997, pp. 262-269.

Bellegarda, J. R., "Exploiting latent semantic information in statistical language modeling," *Proceeding of IEEE*, 88(8) 2000a, pp. 1279-1296.

Bellegarda, J. R., "Large vocabulary speech recognition with multi-span statistical language models," *IEEE Transactions on Speech and Audio Processing*, 8(1) 2000b, pp. 76-84.

Berry, M. W., S. T. Dumais and G. W. O'Brien, "Using Linear algebra for Intelligent Information Retrieval," *Society for Industrial and Applied Mathematics (SIAM): Review*, 37(4) 1995, pp. 573-595.

Berry, M. W., "Large scale singular value computations," *International Journal of Supercomputer Applications*, vol. 6, 1992, pp. 13-49.

Chen, S. and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," *Computer Speech and Language,* 13(4) 1999, pp. 359-394.

Deerwester, S., S. T. Dumais, T. K. Landauer, G. W. Furnas and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, 41(6) 1990, pp. 391-407.

Ding, C. H. Q., "A similarity-based probability model for latent semantic indexing," *Proc. 22nd Annual International ACM SIGIR Conference*, 1999, pp. 58-65.

Federico, M., "Bayesian estimation methods for n-gram language model adaptation," *Proc. of the International Conference on Spoken Language Processing,* vol. 1, 1996, pp. 240-243.

Golub, G. and C. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins, 1989.

Jelinek, F., "Self-Organized Language Modeling for Speech Recognition," *Readings in Speech Recognition*, Morgan-Kaufmann Publishers, 1990, pp. 450-506.

Jelinek, F., "Up From Trigrams," *Proc. European Conference on Speech communication and Technology*, vol. 3, 1991, pp. 1037-1040.

Jelinek, F. and R. Mercer, "Interpolated estimation of Markov source parameters from sparse data," *Pattern Recognition in Practice*, 1980, pp. 381-397.

Katz, S.M., "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3) 1987, pp. 400-401.

Kawabata, T. and M. Tamoto, "Back-off Method for N-gram Smoothing based on Binomial Posteriori Distribution," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 1996, pp.192-195.

Lau, R., R. Rosenfeld and S. Roukos, "Trigger-based language models: A maximum entropy approach," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 1993, pp. 45-48.

Masataki, H., Y. Sagisaka, K. Hisaki and T. Kawahara, "Task adaptation using MAP estimation in *n*-gram language modeling," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 1997, pp.783-786.

Novak, M. and R. Mammone, "Use of non-negative matrix factorization for language model adaptation in a lecture transcription task," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 2001, pp. 541-544.

Ricardo, B.-Y. and R. -N. Berthier, *Modern information retrieval*, Addison-Wesley, 2000.

Written, I. H. and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Transaction on Information Theory*, 37(4) 1991, pp. 1085-1094.

Zhai, C. and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," *Proc. 24th Annual International ACM SIGIR Conference*, 2001, pp. 334-342.

Zhou, G. D. and K. T. Lua, "Interpolation of n-gram and mutual-information based trigger pair language models for Mandarin speech recognition," *Computer Speech and Language,* 13(2) 1999, pp.125-141.

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 45-62

Multi-Modal Emotion Recognition from Speech and Text

Ze-Jing Chuang[*] and Chung-Hsien Wu[*]

Abstract

This paper presents an approach to emotion recognition from speech signals and textual content. In the analysis of speech signals, thirty-three acoustic features are extracted from the speech input. After Principle Component Analysis (PCA) is performed, 14 principle components are selected for discriminative representation. In this representation, each principle component is the combination of the 33 original acoustic features and forms a feature subspace. Support Vector Machines (SVMs) are adopted to classify the emotional states. In text analysis, all emotional keywords and emotion modification words are manually defined. The emotion intensity levels of emotional keywords and emotion modification words are estimated based on a collected emotion corpus. The final emotional state is determined based on the emotion outputs from the acoustic and textual analyses. Experimental results show that the emotion recognition accuracy of the integrated system is better than that of either of the two individual approaches.

1. Introduction

Human-machine interface technology has been investigated for several decades. Recent research has placed more emphasis on the recognition of nonverbal information, and has especially focused on emotion reaction. Many kinds of physiological characteristics are used to extract emotions, such as voice, facial expressions, hand gestures, body movements, heartbeat and blood pressure. Scientists have found that emotion technology can be an important component in artificial intelligence [Salovey *et al.* 1990], especially for human-human communication. Although human-computer interaction is different from human-human communication, some theories show that human-computer interaction shares

[*] Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, ROC
E-mail: {bala, chwu}@csie.ncku.edu.tw

basic characteristics with human-human interaction [Reeves *et al.* 1996]. In addition, affective information is pervasive in electronic documents, such as digital news reports, economic reports, e-mail, etc. The conclusions reached by researchers with respect to emotion can be extended to other types of subjective information [Subasic *et al.* 2001]. For example, education assistance software should be able to detect the emotions of users and; therefore; choose suitable teaching courses. Moreover, the study of emotions can apply to some assistance systems, such as virtual babysitting systems or virtual psychologist systems.

In recent years, several research works have focused on emotion recognition. Cohn and Katz [Cohn *et al.* 1998] developed a semi-automated method for emotion recognition from faces and voices. Silva [Silva *et al.* 2000] used the HMM structure to recognize emotion from both video and audio sources. Yoshitomi [Yoshitomi *et al.* 2000] combined the hidden Markov model (HMM) and neural networks to extract emotion from speech and facial expressions. Other researchers focused on extracting emotion from speech data only. Fukuda and Kostov [Fukuda *et al.* 1999] applied a wavelet/cepstrum-based software tool to perform emotion recognition from speech. Yu [Yu *et al.* 2001] developed a support vector machine (SVM)-based emotion recognition system. However, few approaches have focused on emotion recognition from textual input. Textual information is another important communication medium and can be retrieved from many sources, such as books, newspapers, web pages, e-mail messages, etc. It is not only the most popular communication medium, but also rich in emotion. With the help of natural language processing techniques, emotions can be extracted from textual input by analyzing punctuation, emotional keywords, syntactic structure, semantic information, etc. In [Chuang *et al.* 2002], the authors developed a semantic network for performing emotion recognition from textual content. That investigation focused on the use of textual information in emotion recognition systems. For example, the identification of emotional keywords in a sentence is very helpful to decide the emotional state of the sentence. A possible application of textual emotion recognition is the on-line chat system. With many on-line chat systems, users are allowed to communicate with each other by typing or speaking. A system can recognize a user’s emotion and give an appropriate response.

In this paper, a multi-modal emotion recognition system is constructed to extract emotion information from both speech and text input. The emotion recognition system classifies emotions according to six basic types: happiness, sadness, anger, fear, surprise and disgust. If the emotion intensity value of the currently recognized emotion is lower than a predefined threshold, the emotion output is determined to be neutral. The proposed emotion recognition system can detect emotions from two different types of information: speech and text. To evaluate the acoustic approach, a broadcast drama, including speech signal and textual content, is adopted as the training corpus instead of artificial emotional speech. During feature selection, an initial acoustic feature set that contained 33 features is first analyzed and extracted. These acoustic features contain several possible characteristics, such as intonation, timbre, acoustics, tempo, and rhythm. We also extract some features to represent special intonations, such as trembling speech, unvoiced speech, and crying speech. Finally, among these diverse features, the most significant features are selected by means of principle component analysis (PCA) to form an acoustic feature vector. The acoustic feature vector is fed to the Support Vector Machines (SVMs) to determine the emotion output according to hyperplanes determined by the training corpus.

For emotion recognition via text, we assume that the emotional reaction of an input sentence is essentially represented by its word appearance. Two primary word types, "emotional keywords" and "emotion modification words," are manually defined and used to extract emotion from the input sentence. All the extracted emotional keywords and emotion modification words have their corresponding "emotion descriptors" and "emotion modification values." For each input sentence, the emotion descriptors are averaged and modified using the emotion modification values to give the current emotion output. Finally, the outputs of the textual and acoustic approaches are combined with the emotion history to give the final emotion output.

The rest of the paper is organized as follows. Section 2 describes the module for recognizing emotions from speech signals. The details of SVM classification model is also provided in this section. Then the textual emotion recognition module and the integration of these two modules are presented in sections 2.3 and 3, respectively. Finally, experimental results obtained using the integrated emotion recognition system are provided in section 5, and some conclusions are drawn in section 6.

2. Acoustic Emotion Recognition Module

Deciding on appropriate acoustic features is a crucial step in emotion recognition. As in similar research, this study adopts the pitch and energy features and their derivatives. In addition, some additional characteristics may be found in emotional speech, such as trembling speech, unvoiced speech, varying speech duration, and hesitation. These features are also extracted in our approach.

2.1 Feature Extraction

A diagram of the acoustic feature extraction approach is shown in Figure 1. In the proposed approach, four basic acoustic features, pitch, energy, formant 1 (F1), and the zero crossing rate (ZCR), are estimated first. Previous research has shown that emotional reactions are strongly related to the pitch and energy of the speech. For example, the pitch of speech associated with anger or happiness is always higher than that associated with sadness or disappointment, and the energy associated with surprise or anger is also greater than that associated with fear.

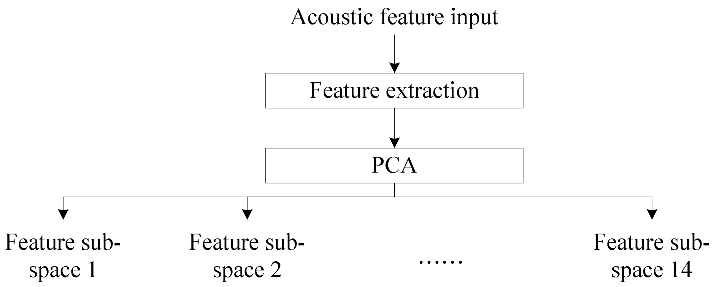


Figure 1. Diagram of the acoustic feature extraction module.

To extract an appropriate feature set, a short-time processing technique is first applied. The contours of the acoustic features are used to represent the time-varying feature characteristics. Each contour can be represented by its mean, slope, and slope difference. The Legendre polynomial [Abramowitz *et al*. 1972] is adopted to represent the contours of these four features.

In feature extraction, we adopt several parameters that are based on pitch and energy. We extract 33 acoustic features in the following 13 categories:

(1) 4th-order Legendre parameters for the pitch contour;

(2) 4th-order Legendre parameters for the energy contour;

(3) 4th-order Legendre parameters for the formant one (F1) contour;

(4) 4th-order Legendre parameters for the zero crossing rate (ZCR) contour;

(5) maximum energy;

(6) maximum smoothed energy;

(7) minimum, median, and standard deviation of the pitch contour;

(8) minimum, median, and standard deviation of the energy contour;

(9) minimum, median, and standard deviation of the smoothed pitch contour;

(10) minimum, median, and standard deviation of the smoothed energy contour;

(11) ratio of the sample number of the upslope to that of the downslope for the pitch contour;

(12) ratio of the sample number of upslope to that of the downslope for the energy contour;

(13) pitch vibration.

The features in categories (1) to (8) are statistical parameters of four basic acoustic features. In order to remove discontinuities from the contour, the pitch and energy features in categories (9) and (10) are smoothed using window method.

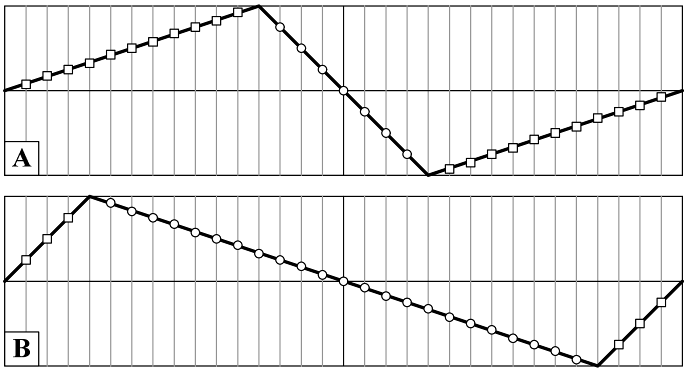


Figure 2. The ratio of up-slope sample number to the down-slope sample number. Two contours with the same wavelength are shown in parts A and B; the square symbols indicate the up-slope sample, and the circle symbols indicate the down-slope sample.

The ratios described in categories (11) and (12) represent not only the slope but also the shape of each vibration in the contour. Figure 2 shows the difference between these parameters. In this figure, each part shows the vibration of a contour. In order to show how the parameters are used, we assume that the length and the amplitude of these two contours are the same. In part A, the length of the upslope contour is longer than that of the downslope contour, while the opposite is shown in part B. The ratio of upslope to downslope is 3.14 (22 upslope samples to 7 downslope samples) in part A and 0.26 (6 upslope samples to 23 downslope samples) in part B.

Trembling speech can be characterized by means of pitch vibration. For category (13), the pitch vibration is defined and calculated as follows:

$$P_r = \frac{1}{N}\sum_{i=0}^{N-1}\delta\left[\left(P(i)-\bar{P}\right)\times\left(P(i+1)-\bar{P}\right)\right],\quad \delta[x]=\begin{cases}1 & x<0\\ 0 & x\geq 0\end{cases}, \qquad (1)$$

where $\bar{P}$ is the mean value of the pitch contour.

2.2 Principle Component Analysis

Principal component analysis (PCA) is a standard statistical approach that can be used to extract the main components from a set of parameters. As described in the previous section, an initial set of 33 features is firstly extracted. After PCA is performed, 14 dimensions of principle components are chosen to capture over 90% of the total variance.

Traditionally, the 14 dimensions of principle components are used to perform classification directly. But in our approach, the principle components are used to select a more detailed subspace. In PCA, each principle component is the linear combination of the original features. If a principle component is selected, the features that have larger combination weights are also selected and form a feature subspace. The combination weights of the original features are represented in the transformation matrix, which is calculated in PCA. By setting the threshold of the combination weights to a value of 0.2, we can select the significant features for each principle component to form a feature set. Therefore, we have 14 feature subspaces.

Table 1 shows an example of feature subspace generation. Suppose that F_1 to F_5 are the original features, that P_1 and P_2 are the selected principle components in PCA, and that the values indicate the combination weights. By selecting the original features according to values that are greater than the threshold of 0.2, we can select $\{F_1, F_3, F_4\}$ as the first feature subspace from P_1 and $\{F_2, F_4\}$ as the second feature subspace from P_2.

Table 1. An example of feature subspace generation. When a threshold value of 0.2 is applied, the generated feature subspaces are {F_1, F_3, F_4} and {F_2, F_4}.

	P_1	P_2
F_1	**0.2**	0.1
F_2	0.15	**0.3**
F_3	**0.2**	0.1
F_4	**0.3**	**0.4**
F_5	0.15	0.1

2.3 Emotion Recognition Using SVM Models

The support vector machine (SVM) [Cristianini *et al.* 2001] has been widely applied in many research areas, such as data mining, pattern recognition, linear regression, and data clustering. Given a set of data belonging to two classes, the basic idea of SVM is to find a hyperplane that can completely distinguish two different classes. The hyperplane is decided by the maximal margin of two classes, and the samples that lie in the margin are called “support vectors.” The equation of the hyperplane is described in Eq. (2):

$$D(x)=\sum_{i=1}^{N}\alpha_i y_i(x\cdot x_i)+w_0 \ . \tag{2}$$

Traditional SVMs can construct a hard decision boundary with no probability output. In this study, SVMs with continuous probability output are proposed. Given the test sample x', the probability that x' belongs to class c is $P(class_c|x')$. This value is estimated based on the following factors:

the distance between the test input and the hyperplane,

$$R = \frac{D(x')/\|w\|}{1/\|w\|} = D(x') \;; \tag{3}$$

the distance from the class centroid to the hyperplane,

$$R' = \frac{R}{D(\bar{x})} = \frac{D(x')}{D(\bar{x})} \;; \tag{4}$$

where $\bar{x}$ is the centroid of the training data in a class;

the classification confidence of the class;

the classification accuracy evaluated based on the training data is used to define the classification confidence of class c,

$$P_c = \frac{\textit{Number of sentences correctly recognized as class c}}{\textit{Total number of sentences in class c}} \;. \tag{5}$$

Finally, the output probability is defined as follows according to the above factors:

$$P(class_c|x') = \frac{P_c}{1+\exp(1-R')} = \frac{P_c}{1+\exp\left(1-\dfrac{D(x')}{D(\bar{x})}\right)} \;. \tag{6}$$

As described above, the acoustic feature set is divided into 14 feature sub-spaces. For each sub-space, an SVM model is applied to decide on the best class of the speech input. The final output is the combination of these different SVM outputs, and shown as follows:

$$\begin{aligned} P(class_c|x') &= \left(\prod_{i=1}^{S} P_i(class_c|x')\right)^{\frac{1}{S}} \\ &= \left(\prod_{i=1}^{S}\left(\frac{P_c}{1+\exp\left(1-D(x')/D(\bar{x})\right)}\right)\right)^{\frac{1}{S}} \end{aligned} \;, \tag{7}$$

where the probability $P_i(class_c|x')$ is the output of SVM in the i-th feature subspace and S (=14) is the number of sub-spaces.

3. Textual Emotion Recognition Module

The most popular method for performing emotion recognition from text is to detect the appearance of emotional keywords. Generally, not only the word level but also the syntactic and semantic levels may contain emotional information. Figure 3 shows a diagram of the textual emotion recognition module. A front-end speech recognizer is first used to convert the speech signal into textual data. To extract the emotional state from the text input, we assume that every input sentence includes one or more emotional keywords and emotion modification words. The emotional keywords provide a basic emotion description of the input sentence, and the emotion modification words can enhance or suppress the emotional state. Finally, the final emotional state is determined by combining the recognition results from both textual content and speech signal.

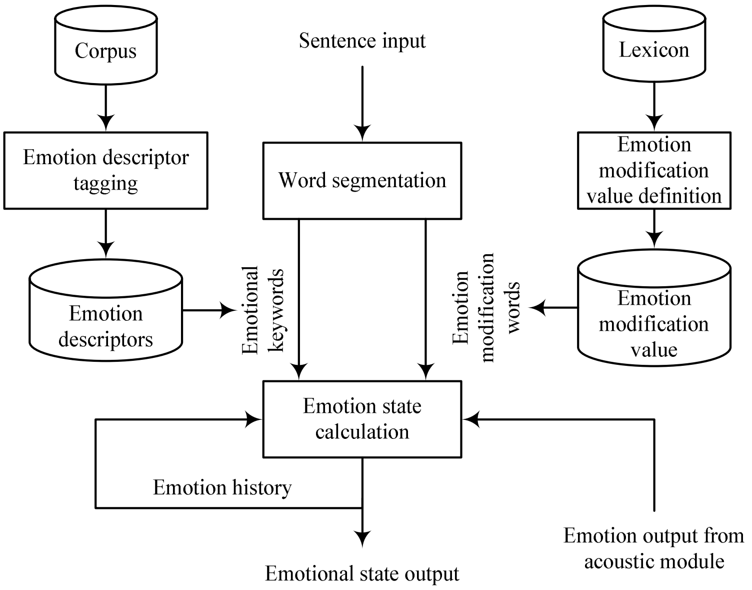

Corpus
Sentence input
Lexicon
Emotion descriptor tagging
Word segmentation
Emotion modification value definition
Emotion descriptors
Emotional keywords
Emotion modification words
Emotion modification value
Emotion state calculation
Emotion history
Emotional state output
Emotion output from acoustic module

Figure 3. Diagram of textual emotion recognition module.

3.1 Front-end Processor

In order to transform the speech signal into textual data, a keyword spotting system [Wu *et al.* 2001] is applied first. The hidden Markov models (HMM) are adopted to perform keyword spotting, and the Mel-frequency cepstrum coefficients (MFCC) are extracted as the acoustic features. Obviously, the speech recognizer plays a very important role in the textual emotion recognition module. In our approach, since we consider only the emotional keywords and extract their corresponding information using HowNet, a keyword spotting system is adopted to spot the emotional keywords and emotion modification words.

3.2 Emotional Keyword Definition

For each emotional keyword, the corresponding emotion descriptor is manually defined. The emotion descriptor is a set of descriptions of the emotion reactions corresponding to the keywords. Basically, it contains an emotional state label and an intensity value, which ranges from 0 to 1. The emotional state label can be one of the following six labels: happiness, sadness, anger, fear, surprise, and disgust. The intensity value describes how strongly the keyword belongs to this emotional state. In many cases, however, a word may contain one or more emotional reactions. Accordingly, there may be more than one emotion descriptor for each emotional keyword. For example, two emotional states, sadness and anger, are involved in the keyword "disappointed." However, the keyword "depressed" is annotated with only one emotional state: sadness. After the tagging process is completed, the emotion descriptors of the word "disappointed" are {(2, 0.2), (3, 0.6)}, and the emotion descriptor of the word "depressed" is {(3, 0.6)}. The numbers 2 and 3 in the parentheses indicate the emotional states anger and sadness, respectively. The numbers 0.2 and 0.6 represent the degree of the emotional states. In the following, we describe how the emotional state is calculated. Consider the following input sentence at time t:

S_t : "We felt very ***disappointed*** and ***depressed*** at the results."

Here, the i^{th} emotional keyword is represented by k_i^t, $1 \le i \le M_t$, and M_t is the number of keywords in sentence S_t. In this example, k_1^t and k_2^t represent the words "***disappointed***" and "***depressed,***" respectively, and the value of M_t is 2. For each emotional keyword k_i^t, the corresponding emotion descriptor is $\left(l_r^{ti}, v_r^{ti}\right)$, $1 \le r \le R_i^t$, where R_i^t represents the number of emotion descriptors of k_i^t. The variable l_r^{ti} is the r^{th} emotional state label, and v_r^{ti} is the r^{th} intensity value of k_i^t. The value of the emotional state label can range from 1 to 6, corresponding to six emotional states: happiness, sadness, anger, fear, surprise, and disgust. In this case, the values of R_1^t and R_2^t are 2 and 1, respectively. For k_1^t, the values of l_1^{t1}, l_2^{t1}, v_1^{t1}, and v_2^{t1} are 2, 3, 0.2, and 0.6, respectively. For k_2^t, the values of l_1^{t2} and v_1^{t2} are 3 and 0.6, respectively.

The emotion descriptors of each emotional keyword are manually defined based on a Chinese lexicon containing 65620 words. In order to eliminate errors due to subjective judgment, all the words are firstly tagged by three people individually and then cross validated by the other two people. For each word, if the results tagged by different people are close, the average of these values will be set as the emotion descriptors of the word. If the three people cannot reach a common consensus, an additional person will be asked to tag the word, and the result will be taken into consideration. Based on experience, only a few words need additional suggestions.

The final tagged results for the emotion descriptors are shown in Table 2. A total of 496 words are defined as emotional keywords, and there are some ambiguities. Only 423 of them have unique emotional label definitions, 64 words have 2 emotional label definitions, and 9 words have 3 emotional label definitions. Most of the ambiguities occur in the anger and sadness categories. For example, the word "unhappy" may indicate an angry emotion or a sad emotion, according to the individual's personality and situation.

Table 2. The ambiguity of tagged emotion labels for an emotional keyword.

Number of tagged emotion labels of an emotional keyword			**Total**
1	**2**	**3**	
423	64	9	496

3.3 Emotion Modification Value

Besides, emotional keywords, emotion modification words also play an important role in emotion recognition. For example, the following three phrases have different emotional states and emotion degrees: “happy,” “very happy,” and “not happy.” The only difference between these three phrases is in the emotion modification words “very” and “not.” In order to quantify the emotional effect for different emotion modification words, we define an emotion modification value. According to the previous analysis of emotions [Lang. 1990], all emotion modification words can be classified into two groups: positive emotion modification words and negative emotion modification words. Positive emotion modification words strengthen the current emotional state, while negative emotion modification words reverse the current emotional state. For example, “very happy” is stronger than “happy” because of the use of word “very,” but “not happy” may be sad or angry.

The emotion modification value is manually defined for each emotion modification word. It consists of a sign and a number. The sign indicates the positive or negative state of the emotion modification word, and the number indicates the modification strength of the emotion modification word. For example, the emotion modification values of the words or phrases “a little,” “very,” and “extremely” are +1, +2, and +3, respectively. And the emotion modification values of the words or phrases “not at all,” “not,” and “never” are -1, -2, and -3, respectively. The degree ranges from 1 to 3. For the example, in previous section, in the case of $\boldsymbol{S_t}$: *“We felt **very** disappointed and depressed at the results,”* the emotion modification word is represented by g_j^t, $1 \le j \le N_t$, where N_t is the number of emotion modification words in sentence S_t. The corresponding emotion modification value of g_j^t is represented by u_j^t. In this example, g_1^t represents the word “***very.***” The values of N_t and u_1^t are 1 and +2, respectively.

4. Final Emotional State Determination

The final emotional state is the combination of the three outputs: the emotion recognition result obtained from acoustic features, the emotion descriptors of the emotional keywords, and the emotion modification values of the emotion modification words in this sentence. Given an input sentence S_t at time t, the final emotion reaction obtained from the textual content of sentence S_t is represented by E_t^C, which is a six dimension vector, $E_t^C = \left(e_1^{tC}, e_2^{tC}, e_3^{tC}, e_4^{tC}, e_5^{tC}, e_6^{tC}\right)$.

The six elements in E_t^C represent the relationship between sentence S_t and the six emotional states: happiness, sadness, anger, fear, surprise, and disgust, respectively. Each value is calculated as follows:

$$e_o^{tC} = \frac{1}{3}\left(\prod_{x=1}^{N} u_x^t\right)^{\frac{1}{N}} \left(\frac{\sum_{y=1}^{M}\sum_{z=1}^{R_y^t} S\left(l_z^{ty},o\right)v_z^{ty}}{\sum_{y=1}^{M}\sum_{z=1}^{R_y^t} S\left(l_z^{ty},o\right)} \right) \quad ,1 \le o \le 6 \quad . \tag{8}$$

The value in the first pair of parentheses is a geometric mean of all the emotion modification values, and the value in the second pair of parentheses is the average of intensity values that belong to emotional state o. The function $S\left(l_z^{ty},o\right)$ is a step function with a value of 1 when $l_z^{ty} = o$ and a value of 0 when $l_z^{ty} \ne o$. The constant 1/3 is used to normalize the emotion intensity value to the range from -1 to 1.

After the emotion reaction from the textual content has been calculated, the final emotion output E_t is the combination of E_t^A and E_t^C,

$$\begin{aligned} E_t &= \left(e_1^t, e_2^t, e_3^t, e_4^t, e_5^t, e_6^t\right) \\ e_o^t &= \alpha e_o^{tA} + \left(1-\alpha\right)e_o^{tC} \quad , \text{ where } \alpha = \max_{1\le o\le 6}\left(e_o^{tA}\right) \end{aligned} \quad . \tag{9}$$

The emotion output of acoustic module e_i^{tA} ranges from 0 to 1, and the emotion output of textual module e_i^{tC} ranges from -1 to +1.

According to the assumption that the current emotional state is influenced by the previous emotional states, the output of the current emotion vector E_t must be modified by means of its previous emotion vector E_{t-1}. The recursive calculation of the emotion history is defined as follows:

$$E_t' = \delta E_t + \left(1-\delta\right)E_{t-1}, \quad t \ge 1, \tag{10}$$

where E_t is the t-th emotion vector calculated in as described in the previous section; E_t' indicates that the final output considers the emotion history, and the initial value E_0 is the output without any modification. The combination coefficient δ is empirically set to 0.75.

5. Experimental Results

For the purpose of system evaluation, in order to obtain real emotional states from natural speech signals, we collected the training corpus from broadcast dramas. There were 1085 sentences in 227 dialogues from leading man and 1015 sentences in 213 dialogues from leading woman. The emotional states of these sentences were tagged manually. The emotion tagging results are listed in Table 3.

Table 3. Tagged emotion labels in the testing corpus.

	Number of tagged sentences	
	Male	**Female**
Happiness	126	121
Sadness	121	92
Anger	98	80
Fear	60	58
Surprise	196	172
Disgust	106	113
Neutral	1617	1530

The system was implemented on a personal computer with a Pentium IV CPU and 512 MB of memory. A high-sensitivity microphone was connected to the computer and provided real-time information about speech signals.

5.1 Experiment on Acoustic Feature Extraction with PCA

As described in section II, 33 acoustic features are analyzed using PCA with thresholds of 90% and 0.2, which are the thresholds for deciding on the important principle components and the significant features of each component, respectively. The PCA process also divides the original feature space into 14 feature sub-spaces. The value of the threshold and the number of feature sub-spaces are decided experimentally.

For acoustic feature evaluation, an SVM classification system was constructed for this experiment. The threshold for deciding on the important principle components (R^2) was set to be within a range of from 85% to 100% with a step size of 2%, and the threshold for deciding on the significant features of components (T) was set to be with a range of from −1 to 1 with a step size of 0.1. The experimental results are shown in Figure 4.

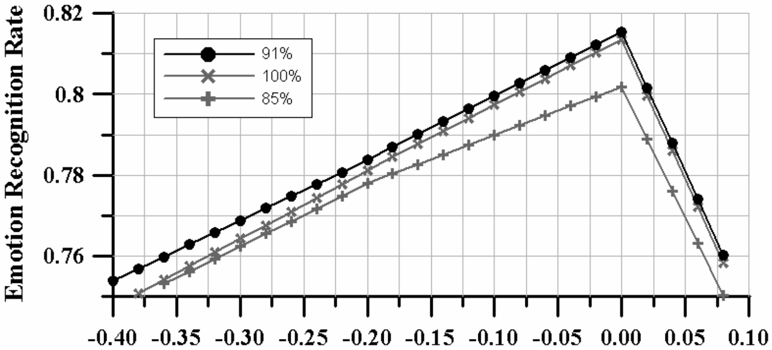


Figure 4. Emotion recognition rates for acoustic features under different PCA thresholds. The black line indicates the results obtained when R^2 = 91%, and the two gray lines indicate the results obtained when R^2 = 85% and 100%.

As shown in Figure 4, the achieved recognition rate was 63.33% when $T = -1$. When R^2 = 91% and $T = 0$, the achieved recognition rate was 81.55%, the highest rate obtained in all the tests. The results show that after PCA was performed, the orthogonal feature space was extracted from the original feature sets when $R^2 = 91\%$ and $T = 0$, and the emotion recognition rate also increased due to the elimination of dependency.

Based on the results, we could decide on the appropriate number of feature sub-spaces. Figure 5 shows the relation between the number of sub-spaces and R^2. Since the previous experiment indicated that an appropriate value of R^2 was 91%, the appropriate number of sub-spaces was chosen as 14 based on the curve in Figure 5.

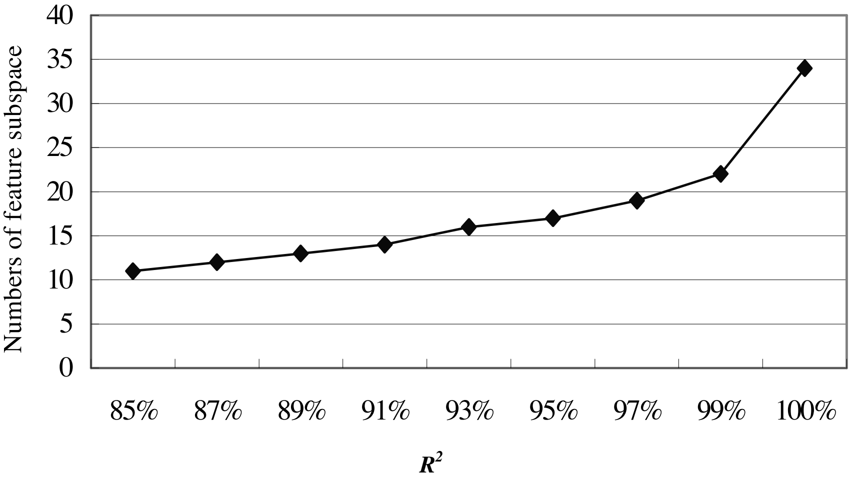


Figure 5. The relationship between R2 and the number of feature sub-spaces.

5.2 Experiment on Keyword Spotting

Since the emotion recognition rate of the textual module depends on the recognition rate of the keyword spotting system, the aim of this experiment was to identify the relationship between the keyword spotting system and textual emotion recognition module. The test corpus is first prepared with all emotional keywords are annotated manually, and then all the emotional keywords in test corpus was selected randomly and fed to the textual emotion recognition module. The emotion recognition rates of the textual module according to varying ratio of the number of emotional keywords are illustrated in Figure 6.

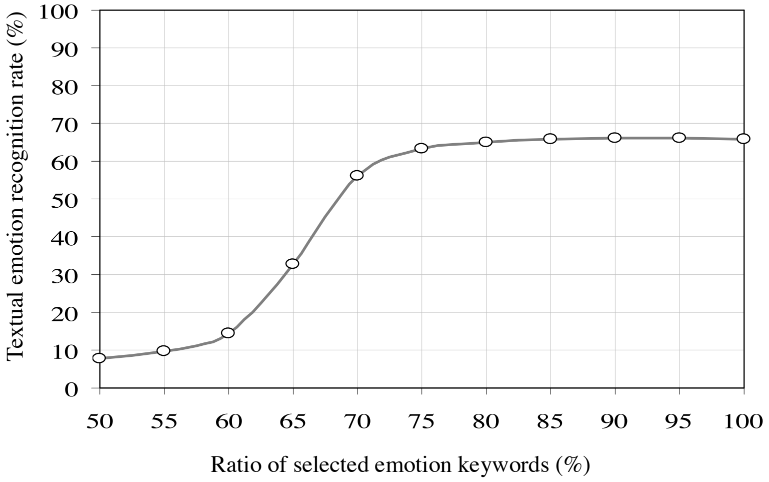


Figure 6. The relationship between the keyword recognition rate and the emotion recognition rate.

As shown in Figure 6, the emotion recognition rate of the textual module did not increase after the ratio of selected keywords reached an accuracy rate of 75%. That means if the keyword recognition rate is higher than 75%, the output of the textual emotion recognition module will reach an upper bound. Since the keyword recognition rate of the system can reach 89.6%, this keyword spotting system is suitable for the textual emotion recognition module.

5.3 Emotion Recognition Results Obtained from Acoustic Information

In this experiment, 14 feature subspaces were adopted. The radial basis function was chosen as the kernel function in the SVM model. Table 4 shows the results obtained by acoustic module. Since the dramatic dialogues were spoken by professional actors, the variation of speech intonation was very large with, therefore, decreased the recognition rate. In addition, the recognition rates for neutral and sadness were a little higher than those for other emotions. Checking the speech corpus, we found that the intonation patterns for neutral and sadness are more stable than those for other emotions. This was the main reason why these experimental results were obtained.

Table 4. Emotion recognition results obtained, based on acoustic information.

	Recognition rate		
	Male	**Female**	***Average***
Happiness	78.85%	71.90%	75.37%
Sadness	85.40%	88.04%	86.72%
Anger	81.52%	75.00%	78.26%
Fear	72.13%	70.18%	71.16%
Surprise	73.55%	62.54%	68.05%
Disgust	76.32%	68.79%	72.56%
Neutral	88.38%	77.53%	82.96%
Average	79.45%	73.43%	76.44%

The acoustic module is based on the assumption that the speech information is too complicated to be classified using only one SVM. Thus, PCA is used to generate the feature subspace. In order to test this assumption, we compared the recognition results for speech input obtained using the classifier with a single SVM and multiple SVMs. Table 5 shows the comparison and confirms the assumption.

Table 5. A comparison of the results obtained using the acoustic module with a single SVM and multiple SVMs.

	Multiple SVM	**Single SVM**
Happiness	75.37%	68.13%
Sadness	86.72%	75.91%
Anger	78.26%	66.57%
Fear	71.16%	60.55%
Surprise	68.05%	55.62%
Disgust	72.56%	64.54%
Neutral	82.96%	70.01%
Total	76.44%	65.90%

5.4 Emotion Recognition Results Obtained from Textual Content

The experimental results obtained by the textual emotion recognition module are listed in Table 6. From these results, we can find that the recognition rate cannot achieve the same level in the case of the acoustic module, i.e., the keyword-based approach cannot achieve satisfactory performance. The reasons of these results are two twofold. Firstly, owing to the complexity of natural language, sentences with the same emotional state may not contain the same emotional keywords. Secondly, as mentioned above, less than 500 words are labeled as emotional keywords from a total of 65620 words. This leads to the low occurrence rate of the occurrence of emotional keywords. But when emotional keywords appear in a sentence, the emotional reaction of the sentence is always strongly related to these keywords. The keyword-based approach is still helpful for improving performance when integrated with the acoustic module.

Table 6. Emotion recognition results obtained based on textual content.

	Recognition rate		
	Male	**Female**	***Average***
Happiness	66.35%	63.64%	64.99%
Sadness	59.12%	61.96%	60.54%
Anger	76.09%	72.50%	74.29%
Fear	71.03%	65.51%	68.27%
Surprise	66.85%	58.46%	62.66%
Disgust	57.12%	55.34%	56.23%
Neutral	77.98%	64.76%	71.37%
Average	67.79%	63.17%	65.48%

5.5 Emotion Recognition Results Obtained Using the Integrated System

Finally, the experimental results obtained using the integrated system are shown in Table 7. The outside test was performed using an extra corpus collected from the same broadcast drama. There were a total of 200 sentences in 51 dialogues in this corpus. When the integration strategy was used, the performance of the integrated system is better than that any of the individual modules. Compared with the results obtained by the acoustic module, the results obtained with the integrated system were 5.05% higher. In order to understand the results, we verified the test corpus manually and found that when a sentence was recognized as having one emotional state, it usually contained either emotional keywords or no keywords. Only a few sentences contained emotional keywords with opposite the emotional states. Thus when the output of the acoustic module was reliable, the output of the textual module could slightly support the results obtained by the acoustic module. But if the acoustic module could not identify the emotional state of an input sentence, the emotional keywords played an important role in the final calculation.

Table 7. Emotion recognition results obtained using the integrated system.

	Inside	**Outside**
Happiness	84.44%	66.67%
Sadness	82.98%	73.91%
Anger	79.66%	67.65%
Fear	78.24%	62.37%
Surprise	80.33%	69.52%
Disgust	76.51%	70.43%
Neutral	88.24%	76.84%
Average	81.49%	69.63%

6. Conclusion

In this paper, an emotion recognition system with multi-modal input has been proposed. When PCA and the SVM model are applied, the emotional state of a speech input can be classified and fed into the textual emotion recognition module. This approach to recognizing emotions from textual information is based on pre-defined emotion descriptors and emotion modification values. After all the emotion outputs have been integrated, the final emotional state is further smoothed by mean of the previous emotion history. The experimental results show that the multi-modal strategy is a more promising approach to emotion recognition than the single module strategy.

In our study, we investigated a method of textual emotion recognition and also tested the combination of the two emotion recognition approaches. Our method can extract emotions from both speech and textual information without the need for a sophisticated speech recognizer. However, there are still many problems that remain to be solved. For example, in the textual emotion recognition module, syntactic structure information is important for natural language processing but cannot be obtained using HowNet alone. An additional parser may be needed to solve this problem. In the acoustic module, crying and laughing sounds are useful for deciding on the current emotional state but are hard to extract. A sound recognizer may, thus, be useful for improving the emotion recognition performance.

References

Salovey, P. and J. Mayer, "Emotional Intelligence," *Imagination, Cognition and Personality*, vol. 9, no. 3, 1990, pp.185-211.

Reeves, B. and C. Nass, "The Media Equation : How People Treat Computers, Television and New Media Like Real People and Places," *Cambridge Univ. Press*, 1996.

Subasic, P. and A. Huettner, "Affect Analysis of Text Using Fussy Semantic Typing," *IEEE Transactions on Fussy System*, vol. 9, no. 4, 2001, pp.483-496.

Cohn, J.F. and G.S. Katz, "Bimodal Expression of Emotion by Face and Voice," *Proceedings of the sixth ACM international conference on Multimedia: Face/gesture recognition and their applications*, 1998, pp.41-44.

Silva, L. C De and N.P. Chi, "Bimodal Emotion Recognition," *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp.332-335.

Yoshitomi, Y., S.I. Kim, T. Kawano, and T. Kitazoe, "Effect of Sensor Fusion for recognition of Emotional States Using Voice, Face Image and Thermal Image of Face," *Proceedings of the ninth IEEE International Workshop on Robot and Human Interactive Communication*, 2000, pp.173-183.

Fukuda, S. and V. Kostov, "Extracting Emotion from Voice," *Proceedings of IEEE International Workshop on Systems, Man, and Cybernetics*, vol. 4, 1999, pp.299-304.

Yu, F., E. Chang, Y.Q. Xu, and H.Y. Shum, "Emotion Detection from Speech to Enrich Multimedia Content," *Proceedings of IEEE Pacific Rim Conference on Multimedia*, 2001, pp.550-557.

Chuang, Z.J. and C.H. Wu, "Emotion Recognition from Textual Input using an Emotional Semantic Network," *Proceedings of IEEE International Conference on Spoken Language Processing*, 2002, pp.2033-2036.

Abramowitz, M. and I.A. Stegun, "Legendre Functions and Orthogonal Polynomials," in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Dover, 1972, pp.331-339.

Cristianini, N. and J. Shawe-Taylor, "An Introduction to Support Vector Machines," *Cambridge University Press*, 2001.

Wu, C.H. and Y.J. Chen, "Multi-Keyword Spotting of Telephone Speech Using Fuzzy Search Algorithm and Keyword-Driven Two-Level CBSM," *Speech communication*, Vol.33, 2001, pp.197-212.

Lang, P.J., M.M. Bradley, and B.N. Cuthbert, "Emotion, atten Lang tion, and the startle reflex," *Psychological Review*, 97, 1990, pp.377-395.

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 63-76

Multiband Approach to Robust Text-Independent Speaker Identification

Wan-Chen Chen[*+], Ching-Tang Hsieh[*], and Eugene Lai[*]

Abstract

This paper presents an effective method for improving the performance of a speaker identification system. Based on the multiresolution property of the wavelet transform, the input speech signal is decomposed into various frequency bands in order not to spread noise distortions over the entire feature space. To capture the characteristics of the vocal tract, the linear predictive cepstral coefficients (LPCCs) of each band are calculated. Furthermore, the cepstral mean normalization technique is applied to all computed features in order to provide similar parameter statistics in all acoustic environments. In order to effectively utilize these multiband speech features, we use feature recombination and likelihood recombination methods to evaluate the task of text-independent speaker identification. The feature recombination scheme combines the cepstral coefficients of each band to form a single feature vector used to train the Gaussian mixture model (GMM). The likelihood recombination scheme combines the likelihood scores of the independent GMM for each band. Experimental results show that both proposed methods achieve better performance than GMM using full-band LPCCs and mel-frequency cepstral coefficients (MFCCs) when the speaker identification is evaluated in the presence of clean and noisy environments.

Keywords: speaker identification, wavelet transform, linear predictive cepstral coefficient (LPCC), mel-frequency cepstral coefficient (MFCC), Gaussian mixture model (GMM).

1. Introduction

In general, speaker recognition can be divided into two parts: speaker verification and speaker

[*] Department of Electrical Engineering, Tamkang University, Taipei, Taiwan, Republic of China
[+] Department of Electronic Engineering, St. John's & St. Mary's Institute of Technology, Taipei, Taiwan, Republic of China
E-mail: steven@mail.sjsmit.edu.tw, hsieh@ee.tku.edu.tw, elai@ee.tku.edu.tw

identification. Speaker verification refers to the process of determining whether or not the speech samples belong to some specific speaker. However, in speaker identification, the goal is to determine which one of a group of known voices best matches the input voice sample. Furthermore, in both tasks, the speech can be either text-dependent or text-independent. Text-dependent means that the text used in the test system must be the same as that used in the training system, while text-independent means that no limitation is placed on the text used in the test system. Certainly, the method used to extract and model the speaker-dependent characteristics of a speech signal seriously affects the performance of a speaker recognition system.

Many researches have been done on the feature extraction of speech. The linear predictive cepstral coefficients (LPCCs) were used because of their simplicity and effectiveness in speaker/speech recognition [Atal 1974, White and Neely 1976]. Other widely used feature parameters, namely, the mel-frequency cepstral coefficients (MFCCs) [Vergin *et al*. 1999], were calculated by using a filter-bank approach, in which the set of filters had equal bandwidths with respect to the mel-scale frequencies. This method is based on the fact that human perception of the frequency contents of sounds does not follow a linear scale. The above two most commonly used feature extraction techniques do not provide invariant parameterization of speech; the representation of the speech signal tends to change under various noise conditions. The performance of these speaker identification systems may be severely degraded when a mismatch between the training and testing environments occurs. Various types of speech enhancement and noise elimination techniques have been applied to feature extraction. Typically, the nonlinear spectral subtraction algorithms [Lockwood and Boudy 1992] have provided only minor performance gains after extensive parameter optimization. Furui [1981] used the cepstral mean normalization (CMN) technique to eliminate channel bias by subtracting off the global average cepstral vector from each cepstral vector. Another way to minimize the channel filter effects is to use the time derivatives of cepstral coefficients [Soong and Rosenberg 1988]. Cepstral coefficients and their time derivatives are used as features in order to capture dynamic information and eliminate time-invariant spectral information that is generally attributed to the interposed communication channel.

Conventionally, feature extraction is carried out by computing acoustic feature vectors over the full band of the spectral representation of speech. The major drawback of this approach is that even partial band-limited noise corruption affects all the feature vector components. The multiband approach deals with this problem by performing acoustic feature analysis independently on a set of frequency subbands [Hermansky *et al.* 1996]. Since the resulting coefficients are computed independently, a band-limited noise signal does not spread over the entire feature space. In our previous works [Hsieh and Wang 2001, Hsieh *et al*. 2002, 2003], we proposed a multiband feature extraction method in which features from various subbands and the full band are combined to form a single feature vector. This feature extraction method was evaluated in a speaker identification system using vector quantization (VQ), group vector quantization, and the Gaussian mixture model (GMM) as identifiers. The experimental results showed that this multiband feature is more effective and robust than the full-band LPCC and MFCC features, particularly in noisy environments.

In past studies on recognition models, VQ [Soong *et al.* 1985, Buck *et al.* 1985, Furui 1991], dynamic time warping (DTW) [Furui 1981], the hidden Markov model (HMM) [Poritz 1982, Tishby 1991], and GMM [Reynolds and Rose 1995, Alamo *et al.* 1996, Pellom and Hansen 1998, Miyajima *et al.* 2001] were used to perform speaker recognition. The DTW technique is effective in text-dependent speaker recognition, but it is not suitable for text-independent speaker recognition. HMM is widely used in speech recognition, and it is also commonly used in text-dependent speaker verification. It has been shown that VQ is very effective for speaker recognition. Although the performance of VQ is not as good as that of GMM [Reynolds and Rose 1995], VQ is computationally more efficient than GMM. GMM [Reynolds and Rose 1995] provides a probabilistic model of the underlying sounds of a person's voice. It is computationally more efficient than HMM and has been widely used in text-independent speaker recognition.

In this study, the multiband linear predictive cepstral coefficients (MBLPCCs) proposed previously [Hsieh and Wang 2001, Hsieh *et al.* 2002, 2003] are used as the front end of the speaker identification system. Then, cepstral mean normalization is applied to these multiband speech features to provide similar parameter statistics in all acoustic environments. In order to effectively utilize these multiband speech features, we use feature recombination and likelihood recombination methods in the GMM recognition models to evaluate the task of text-independent speaker identification. The experimental results show that the proposed multiband methods outperform GMM using full-band LPCC and MFCC features.

This paper is organized as follows. The proposed algorithm for extracting speech features is described in section 2. Section 3 presents the multiband speaker recognition models. Experimental results and comparisons with the conventional full-band GMM are presented in section 4. Concluding remarks are made in section 5.

2. Multiband Features Based on Wavelet Transform

The recent interest in the multiband feature extraction approach has mainly been attributed to Allen's paper [Allen 1994], where it is argued that the human auditory system processes features from different subbands independently, and that the merging is done at some higher point of processing to produce a final decision. The advantages of using multiband processing are multifold and have been described in earlier publications [Bourlard and Dupont 1996, Tibrewala and Hermansky 1997, Mirghafori and Morgan 1998]. The major drawback of a pure subband-based approach may be that information about the correlation among various subbands is lost. Therefore, we suggest that full-band features should not be ignored, but should be combined with subband features to maximize recognition accuracy. A similar approach that combines information from the full band and subbands at the recognition stage was found to improve recognition performance [Mirghafori and Morgan 1998]. It is not a trivial matter to decide at which temporal level the subband features should be combined. In the multiband approach [Bourlard and Dupont 1996, Tibrewala and Hermansky 1997], different classifiers for each band are used, and likelihood recombination is done at the HMM state, phone or word level. In another approach [Okawa *et al.* 1998, Hariharan *et al.* 2001], the individual features of each subband are combined into a single feature vector prior to decoding. In our approach, the full band and subband features are also used in the recognition model.

Based on time-frequency multiresolution analysis, the effective and robust MBLPCC features are used as the front end of the speaker identification system. First, the LPCCs are extracted from the full-band input signal. Then the wavelet transform is applied to decompose the input signal into two frequency subbands: a lower frequency subband and a higher frequency subband. To capture the characteristics of an individual speaker, the LPCCs of the lower frequency subband are calculated. There are two main reasons for using the LPCC parameters: their good representation of the envelope of the speech spectrum of vowels, and their simplicity. Based on this mechanism, we can easily extract the multiresolution features from all lower frequency subband signals simply by iteratively applying the wavelet transform to decompose the lower frequency subband signals, as depicted in Figure 1. As shown in Figure 1, the wavelet transform can be realized by using a pair of finite impulse response (FIR) filters, h and g, which are low-pass and high-pass filters, respectively, and by performing the down-sampling operation (↓2). The down-sampling operation is used to discard the odd-numbered samples in a sample sequence after filtering is performed.

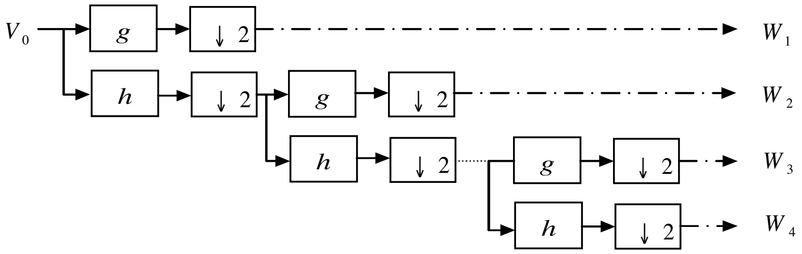


Figure 1. Two-band analysis tree for a discrete wavelet transform

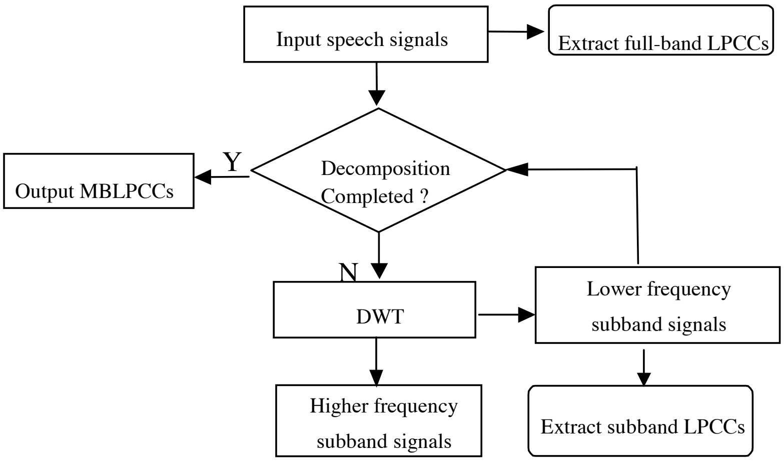

Input speech signals
Extract full-band LPCCs
Y
Output MBLPCCs
Decomposition Completed ?
N
DWT
Lower frequency subband signals
Higher frequency subband signals
Extract subband LPCCs

Figure 2. Features extraction algorithm of MBLPCCs

The schematic flow of the proposed feature extraction method is shown in Figure 2. After the full-band LPCCs are extracted from the input speech signal, the discrete wavelet transform (DWT) is applied to decompose the input signal into a lower frequency subband, and the subband LPCCs are extracted from this lower frequency subband. The recursive decomposition process enables us to easily acquire the multiband features of the speech signal. Based on the concept of the proposed method, the number of MBLPCCs depends on the level of the decomposition process. If speech signals bandlimited from 0 to 4000 Hz are decomposed into two subbands, then three bands signals, (0-4000), (0-2000), and (0-1000) Hz, will be generated. Since the spectra of the three bands will overlap in the lower frequency region, the proposed multiband feature extraction method focuses on the spectrum of the speech signal in the low frequency region similar to extracting MFCC features.

Finally, cepstral mean normalization is applied to normalize the feature vectors so that their short-term means are normalized to zero as follows:

$$\hat{X}_k(t) = X_k(t) - \mu_k , \tag{1}$$

where $X_k(t)$ is the *k*th component of feature vector at time (frame) *t*, and μ_k is the mean of the *k*th component of the feature vectors of a specific speaker's utterance.

In this paper, the orthonormal basis of DWT is based on the 16 coefficients of the quadrature mirror filters (QMF) introduced by Daubechies [1988] (see the Appendix).

3. Multiband Speaker Recognition Models

As explained in section 1, GMM is widely used to perform text-independent speaker recognition and achieves good performance. Here, we use GMM as the classifier. Our initial strategy for multiband speaker recognition is based on straightforward recombination of the cepstral coefficients from each subband (including the full band) to form a single feature vector, which is used to train GMM. We call this identifier model the feature combination Gaussian mixture model (FCGMM). The structure of FCGMM is shown in Figure 3. First, the input signal is decomposed into L subbands. In the "extract LPCC" block, the LPCC features extracted from each band (including the full band) are further normalized to zero mean by using the cepstral mean normalization technique. Finally, the LPCCs from each subband (including the full band) are recombined to form a single feature vector that is used to train GMM. The advantages of this approach are that: (1) it is possible to model the correlation among the feature vectors of each band; (2) acoustic modeling is simpler.

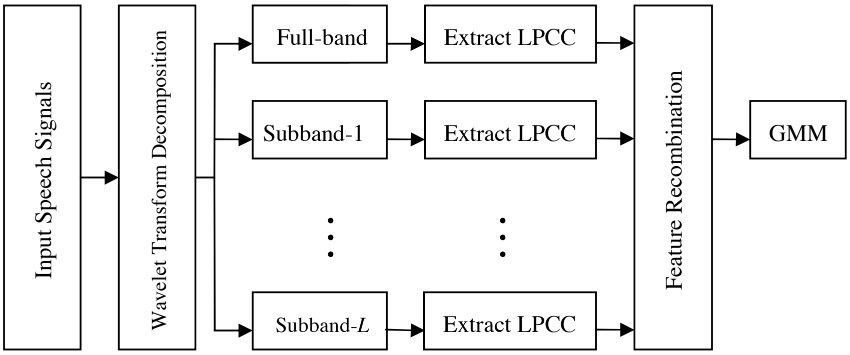


Figure 3. Structure of FCGMM

Our next approach combines the likelihood scores of the independent GMM for each band, as illustrated in Figure 4. We call this identifier model the likelihood combination Gaussian mixture model (LCGMM). First, the input signal is decomposed into L subbands. Then the LPCC features extracted from each band are further normalized to zero mean by using the cepstral mean normalization technique. Finally, different GMM classifiers are applied independently to each band, and the likelihood scores of all the GMM classifiers are combined to obtain the global likelihood scores and a global decision.

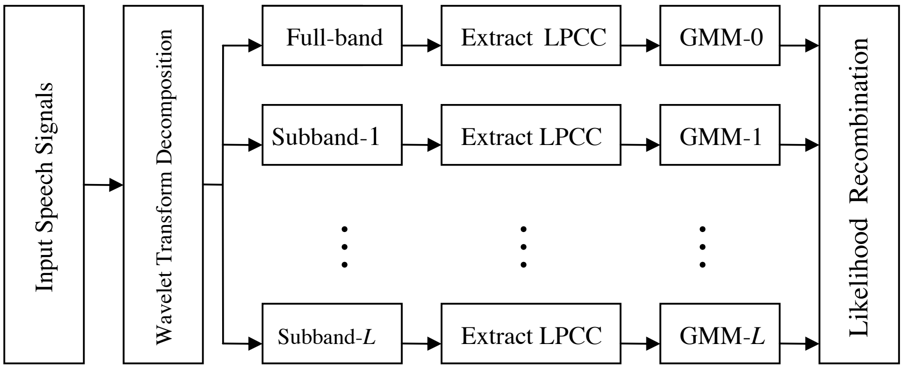


Figure 4. Structure of LCGMM

For speaker identification, a group of S speakers is represented by LCGMMs, λ_1, λ_2,..., λ_S. A given speech utterance X is decomposed into L subbands. Let X_i and λ_{ki} be the feature vector and the associated GMM for band i, respectively. After the log-likelihood $\log P(X_i/\lambda_{ki})$ of band i for a specific speaker k is evaluated, the combined log-likelihood $\log P(X/\lambda_k)$ for the LCGMM of a specific speaker k is determined as the sum of the log-likelihood $\log P(X_i/\lambda_{ki})$ for all bands as follows:

$$\log P(X \mid \lambda_k) = \sum_{i=0}^{L} \log P(X_i \mid \lambda_{ki}), \tag{2}$$

where L is the number of subbands. When $L = 0$, the functions of LCGMM and the conventional full-band GMM are identical. For a given speech utterance X, X is classified to belong to the speaker $\hat{S}$ who has the maximum log-likelihood $\log P(X \mid \lambda_{\hat{S}})$:

$$\hat{S} = \arg \max_{1 \le k \le S} \log P(X \mid \lambda_k). \tag{3}$$

4. Experimental Results

This section presents experiments conducted to evaluate application of FCGMM and LCGMM to text-independent speaker identification. The first experiment studied the effect of the decomposition level. The next experiment compared the performance of FCGMM and LCGMM with that of the conventional GMM using full-band LPCC and MFCC features.

4.1 Database Description and Parameter Setting

The proposed multiband approaches were evaluated using the KING speech database [Godfrey *et al.* 1994] for text-independent speaker identification. The KING database is a collection of conversational speech from 51 male speakers. For each speaker, there are 10 sections of conversational speech that were recorded at different times. Each section consists of about 30 seconds of actual speech. The speech from a section was recorded locally using a microphone and was transmitted over a long distance telephone link, thus providing a high-quality (clean) version and a telephone quality version of the speech. The speech signals were recorded at 8 kHz and 16 bits per sample. In our experiments, noisy speech was generated by adding Gaussian noise to the clean version speech at the desired SNR. In order to eliminate silence segments from an utterance, simple segmentation based on the signal energy of each speech frame was performed. All the experiments were performed using five sections of speech from 20 speakers. For each speaker, 90 seconds of speech cut from three clean version sections provided the training utterances. The other two sections were divided into nonoverlapping segments 2 seconds in length and provided the testing utterances.

In both experiments conducted in this study, each frame of an analyzed utterance had 256 samples with 128 overlapping samples. Furthermore, 20 orders of LPCCs for each frequency band were calculated, and the first order coefficient was discarded. For our multiband approach, we used 2, 3 and 4 bands as follows:

2 bands: (0-4000), (0-2000) Hz;

3 bands: (0-4000), (0-2000), (0-1000) Hz;

4 bands: (0-4000), (0-2000), (0-1000), (0-500) Hz.

4.2 Effect of the Decomposition Level

As explained in section 2, the number of subbands depends on the decomposition level of the wavelet transform. The first experiment evaluated the effect of the number of bands used in the FCGMM and LCGMM recognition models with 50 mixtures in both clean and noisy environments. The experimental results are shown in Table 1. One could see that the 3-band FCGMM achieved better performance under low SNR conditions (for example, 15 dB, 10 dB and 5 dB), but poorer performance under clean and 20 dB SNR conditions, compared with the 2-band FCGMM. Since the 2-band FCGMM used (0-4000) and (0-2000)Hz features, and the 3-band FCGMM used (0-4000), (0-2000) and (0-1000)Hz features, the feature derived from the lower frequency region (below 1kHz) was more robust than the feature derived from the higher frequency region under low SNR conditions.

The best identification rate of LCGMM could be achieved in both clean and noisy environments when the number of bands was set to be three. Since the features were extracted from (0-4000), (0-2000) and (0-1000) Hz subbands and the spectra of the subbands overlapped in the lower frequency region (below 1kHz), the success achieved using the MBLPCC features could be attributed to the emphasis on the spectrum of the signal in the low-frequency region.

It was found that increasing the number of bands to more than three for both models not only increased the computation time but also decreased the identification rate. In this case, the signals of the lowest frequency subband were located in the very low frequency region, which put too much emphasis on the lower frequency spectrum of speech. In addition, the number of samples within the lowest frequency subband was so small that the spectral characteristics of speech could not be estimated accurately. Consequently, the poor result in the lowest frequency subband degraded the system performance.

Table 1. Effect of number of bands on the identification rates for FCGMM and LCGMM recognition models in both clean and noisy environments.

Model \ SNR		clean	20 dB	15 dB	10 dB	5 dB
FCGMM	2 bands	93.45%	85.55%	72.10%	50.25%	30.76%
	3 bands	91.09%	83.87%	76.64%	60.50%	46.22%
	4 bands	88.07%	81.18%	74.29%	63.03%	43.36%
LCGMM	2 bands	93.28%	86.39%	76.47%	53.78%	28.24%
	3 bands	94.96%	92.10%	86.89%	68.07%	43.53%
	4 bands	94.12%	89.41%	84.87%	71.76%	43.19%

4.3 Comparison with Conventional GMM Models

In this experiment, the performance of the FCGMM and LCGMM recognition models was compared with that of the conventional GMM using full-band LPCC and MFCC features under Gaussian noise corruption. For all three models, the number of mixtures was set to be 50.

Here, the parameters of FCGMM and LCGMM were the same as those discussed in section 4.2 except that the number of bands was set to be three. The experimental results shown in Table 2 indicate that the performance of both GMM recognition models using full-band LPCC and MFCC features was seriously degraded by Gaussian noise corruption. On the other hand, LCGMM achieved the best performance among all the models in both clean and noisy environments, and maintained robustness under low SNR conditions. GMM using full-band MFCC features achieved better performance under clean and 20 dB SNR conditions, but poorer performance under lower SNR conditions, compared with the 3-band FCGMM. GMM using full-band LPCC features achieved the poorest performance among all the models. Based on these results, it can be concluded that LCGMM is effective in representing the characteristics of individual speakers and is robust under additive Gaussian noise conditions.

Table 2. Identification rates for GMM using full-band LPCC and MFCC features, FCGMM, and LCGMM with white noise corruption.

SNR Model	Clean	20 dB	15 dB	10 dB	5 dB
GMM using full-band LPCC	88.40%	77.65%	61.68%	35.63%	19.50%
GMM using full-band MFCC	92.61%	85.88%	73.11%	51.60%	32.77%
3-band FCGMM	91.09%	83.87%	76.64%	60.50%	46.22%
3-band LCGMM	94.96%	92.10%	86.89%	68.07%	43.53%

5. Conclusions

In this study, the effective and robust MBLPCC features were used as the front end of a speaker identification system. In order to effectively utilize these multiband speech features, we examined two different approaches. FCGMM combines the cepstral coefficients from each band to form a single feature vector that is used to train GMM. LCGMM recombines the likelihood scores of the independent GMM for each band. The proposed multiband approaches were evaluated using the KING speech database for text-independent speaker identification. Experimental results show that both multiband schemes are more effective and robust than the conventional GMM using full-band LPCC and MFCC features. In addition, LCGMM is more effective than FCGMM.

Acknowledgements

This research was financially supported by the National Science Council, Taiwan, R. O. C., under contract number NSC 92-2213-E032-026.

References

Alamo, C. M., F. J. C. Gil, C. T. Munilla, and L. H. Gomez, "Discriminative training of GMM for speaker identification," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,* 1 1996, pp. 89-92.

Allen, J. B., "How do humans process and recognize speech?," *IEEE Transactions on Speech and Audio Processing*, 2(4) 1994, pp. 567–577.

Atal, B., "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *Journal of Acoustical Society America*, 55 1974, pp. 1304-1312.

Bourlard, H., and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," *Proceedings of International Conference on Spoken Language Processing*, 1996, pp. 426–429.

Buck, J. T., D. K. Burton, and J. E. Shore, "Text-dependent speaker recognition using vector quantization," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,* 10 1985, pp. 391-394.

Daubechies, I., "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics,* 41 1988, pp. 909-996.

Furui, S., "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2) 1981, pp. 254-272.

Furui, S., "Comparison of speaker recognition methods using statistical features and dynamic features," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 29(3) 1981, pp. 342-350.

Furui, S., "Vector-quantization-based speech recognition and speaker recognition techniques," *Proceedings of Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems and Computers,* 4-6 Nov., 2 1991, pp.954-958.

Godfrey, J., D. Graff, and A. Martin, "Public databases for speaker recognition and verification," *Proceedings of ESCA Workshop Automatic Speaker Recognition, Identification, Verification,* 1994, pp. 39-42.

Hariharan, R., I. Kiss, I. Viikki, "Noise robust speech parameterization using multiresolution feature extraction," *IEEE Transactions on Speech and Audio Processing*, 9(8) 2001, pp. 856-865.

Hermansky, H., S. Tibrewala, and M. Pavel, "Toward ASR on partially corrupted speech," *Proceedings of 4th International Conference on Spoken Language Processing*,1 1996, pp. 462–465.

Hsieh, C. T., and Y. C. Wang, "A robust speaker identification system based on wavelet transform," *IEICE Transactions on Information and Systems*, E84-D(7) 2001, pp.839-846.

Hsieh, C. T., E. Lai, and Y. C. Wang, "Robust Speech Features based on Wavelet Transform with application to speaker identification", *IEE Proceedings – Vision, Image and Signal Processing*, 149(2) 2002, pp.108-114.

Hsieh, C. T., E. Lai, and Y. C. Wang, "Robust speaker identification system based on wavelet transform and Gaussian mixture model," *Journal of Information Science and Engineering,* 19 2003, pp. 267-282.

Lockwood, P., and J. Boudy, "Experiments with a nonlinear spectral subtractor (NSS), hidden Markov models and the projection, for robust speech recognition in cars," *Speech Communication*, 11(2-3) 1992, pp. 215–228.

Mirghafori, N., and N. Morgan, "Combining connectionist multiband and full-band probability streams for speech recognition of natural numbers," *Proceedings of International Conference on Spoken Language Processing*, 3 1998, pp. 743–747.

Miyajima, C., Y. Hattori, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Text-independent speaker identification using Gaussian mixture models based on multi-space probability distribution," *IEICE Transactions on Information and Systems*, E84-D(7) 2001, pp. 847-855.

Okawa, S., E. Bocchieri, and A. Potamianos, "Multi-band speech recognition in noisy environments," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,*2 1998, pp. 641–644.

Pellom, B. L., and J. H. L. Hansen, "An effective scoring algorithm for Gaussian mixture model based speaker identification," *IEEE Signal Processing Letters,* 5(11) 1998, pp. 281-284.

Poritz, A., "Linear predictive hidden Markov models and the speech signal," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,* 7 1982, pp. 1291-1294.

Reynolds D. A., and R. C. Rose, "Robust test-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, 3(1) 1995, pp. 72-83.

Soong, F. K., A. E. Rosenberg, L. R. Rabiner, and B. H. Juang, "A vector quantization approach to speaker recognition," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,* 10 1985, pp. 387-390.

Soong, F. K., and A. E. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 36(6) 1988, pp. 871-879.

Tibrewala, S., and H. Hermansky, "Sub-band based recognition of noisy speech," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2 1997, pp. 1255–11258.

Tishby, N. Z., “On the application of mixture AR hidden Markov models to text independent speaker recognition,” *IEEE Transactions on Signal Processing*, 39(3) 1991, pp. 563-570.

Vergin, R., D. O’Shaughnessy, and A. Farhat, “Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition,” *IEEE Transactions on Speech and Audio Processing*, 7(5) 1999, pp. 525-532.

White, G. M., and R. B. Neely, “Speech recognition experiments with linear prediction, bandpass filtering, and dynamic Programming,” *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 24(2) 1976, pp.183-188.

Appendix

The low-pass QMF coefficients h_k used in this study are listed in Table 3. The coefficients of the high-pass filter g_k are calculated from h_k coefficients as follows:

$$g_k = (-1)^k h_{n-1-k} \quad k = 0,1,...,n\,, \tag{4}$$

where n is the number of QMF coefficients.

Table 3. QMF coefficients h_k

h_0	*0.766130*	h_8	*0.008685*
h_1	*0.433923*	h_9	*0.008201*
h_2	*-0.050202*	h_{10}	*-0.004354*
h_3	*-0.110037*	h_{11}	*-0.003882*
h_4	*0.032081*	h_{12}	*0.002187*
h_5	*0.042068*	h_{13}	*0.001882*
h_6	*-0.017176*	h_{14}	*-0.001104*
h_7	*-0.017982*	h_{15}	*-0.000927*

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 77-94

An Innovative Distributed Speech Recognition Platform for Portable, Personalized and Humanized Wireless Devices

Yin-Pin Yang[*]

Abstract

In recent years, the rapid growth of wireless communications has undoubtedly increased the need for speech recognition techniques. In wireless environments, the portability of a computationally powerful device can be realized by distributing data/information and computation resources over wireless networks. Portability can then evolve through personalization and humanization to meet people's needs. An innovative distributed speech recognition (DSR) [ETSI, 1998],[ETSI, 2000] platform, configurable DSR (C-DSR), is thus proposed here to enable various types of wireless devices to be remotely configured and to employ sophisticated recognizers on servers operated over wireless networks. For each recognition task, a configuration file, which contains information regarding types of services, types of mobile devices, speaker profiles and recognition environments, is sent from the client side with each speech utterance. Through configurability, the capabilities of configuration, personalization and humanization can be easily achieved by allowing users and advanced users to be involved in the design of unique speech interaction functions of wireless devices.

Keywords: Distributed, speech recognition, configurable, wireless, portable, personalized, humanized.

1. Introduction

In the current wireless era, cellular phones have become daily-life necessities. People carry their own handsets and make phone calls anytime, everywhere, while public payphones have

[*] Ph.D, Senior Researcher, Advanced Technology Center, Computer and Communications Research Laboratories, Industrial Technology Research Institutes
E-mail: YinPinYang@itri.org.tw TEL: 886-3-5914830 FAX: 886-3-5820098
Address: E000 CCL/ITRI Rm.712, Bldg.51, 195 Sec.4, Chung Hsing Rd., Chutung, Hsinchu 310, Taiwan.

almost disappeared. Inspired by this vast number of mobile phone users, the wireless communication industry is developing wireless data services to create more profit. Wireless devices can be treated as terminals of an unbounded information/data network – the Internet. However, the small screen sizes of mobile devices discourage users from surfing the Internet in mobile situations. Wireless data services are not as attractive as was expected, and this is one of the major reasons for the so-called “3G Bubble” [Baker, 2002][Reinhardt *et al*, 2001].

On the other hand, the handset market is still blooming. Personal, stylish and fashionable features, such as ring tones, color screen displays, covers, and so on, are all very popular, especially among teenagers. Functionally speaking, portable devices, such as PDAs, pocket/palm PCs and digital cameras, are now integrated with handsets. Many interesting applications, such as portable electronic dictionaries, map navigators, and mobile learning, can be built into mobile devices. However, these functions or services still cannot create serious business opportunities for telecom companies.

What will future appealing services for cell phones be? “Talking to a machine,” or interacting with a machine, might be a candidate. That is, besides talking to human-beings through voice channels, people may like to talk to machines and access the Internet through data channels. The possibilities are unlimited. Handsets may thus evolve into personal “intimate pets” that people will use from childhood to grownup. In this scenario, speech interaction will play an important part in humanizing devices [Hiroshi et al. 2003]. However, due to the limitations of the current state-of-art speech recognition techniques, the robustness issue [Deng *et al*. 2003][Wu *et al*. 2003][Lee 1998] is always a bottleneck in commercializing speech recognition products. This imperfection reveals the importance of configurability. In the following paragraphs, the relationships among configurability, personalization, and wireless environments will be explored.

Speech Recognition and Wireless Environments

How does a speech recognition system fit into a the wireless network? In this paper, we will primarily highlight two key terms “distributed” and “configurable.” The term “distributed” can be interpreted as follows: computation distributed and data distributed. As for the former, normally speech recognition functions are needed in mobile situations, and devices are usually thin and lacking in computational power. It would be much easier to design speech recognition functions if the computation involved in recognition processes would be distributed over wireless networks by means of a client-server architecture. As for the latter, speech recognition is by nature a pattern matching process, which needs to acquire utterances within a given application domain. For example, a speaker-independent (SI) continuous digit recognizer targeting the Taiwanese market needs to acquire a great large number of sample continuous digit utterances from all dialects in this market. The representation and quality of the sample utterances used for training or adaptation can seriously dominate the performance of a speech recognizer. If a wireless network is used, speech data acquisition can be done in a much more efficient and systematical way. More importantly, the acquired speech data, labeled by means of a speaker profile, recognition environment, and device/microphone type, can be kept on the server. Speech data will thus not be abandoned when particular applications or services are discontinued.

From the above, we can conclude that we need a centralized speech recognition server embedded in the wireless infrastructure. When we say "talking to a machine", the "machine" is actually an entire wireless network. People talk to the same lifetime recognizer, and the recognizer evolves continuously. This speech recognition server can provide any types speech recognition services (computation distributed) for all classes of wireless mobile devices. These services continuously acquire speech data from all locations (data distributed), and adapt the engine performance all the time. For each recognition task, there is a "configuration file" (or, say, a tag) to record all of the information regarding types of services, speaker profiles, recognition environments, etc. We call this type of server a configurable distributed speech recognition (C-DSR) server.

In the following, the history of DSR developed by ETSI/Aurora will be briefly described. Then, the innovative C-DSR platform will be introduced.

Distributed Speech Recognition (DSR) developed by ETSI/Aurora

Instead of squeezing the whole recognizer into a thin device, it seems more reasonable to host recognition tasks on a server and exchange information between the client and server. However, due to the low bit-rates of speech coders (note that coders are designed for humans, not recognizers), the speech recognition performance can be significantly degraded. The DSR, proposed by ETSI Aurora, overcomes these problems by distributing the recognition process between the client and server, by using an error protected data channel to send parameterized speech features.

From DSR to Configurable DSR (C-DSR)

Aurora DSR can be seen as a speech "coder" [Digalakis *et al.* 1999] designed to enable handset users to talk to their recognizers. Besides handsets, there are many other mobile devices that need DSR services, and they all operate in different environments and in different recognition modes. Each combination or, say, configuration, needs its own "coder" to achieve better performance. Based on these needs, C-DSR was built as an integrated client-server platform which not only offers a convenient way to construct speech recognition functions on various client devices, but also provides powerful utilities/tools to assist each configuration to obtain its own coder in order to increase the overall recognition task completion rate. To achieve these goals, C-DSR maximizes the advantages of data channels and centralized servers by means of its "configurable" capability - *configurability*. The configurability can be considered from two points of views.

The C-DSR Client

From the client side viewpoint, speech recognition processing is configurable to work with: (1) various kinds of thin to heavy mobile devices, ranked according to their computational power, noting that most of devices do not have sufficient computational power to perform the feature extraction process proposed by ETSI Aurora; (2) various types of recognition environments, such as offices, homes, streets, cars, airports, etc; this information about recognition environments can help recognition engines achieve greater accuracy; (3) various types of recognition services, such as command-based, grammar-based, speaker independent/ de--pendent mixed mode, and dialogue style services; (4) various speaker profiles since speaker information can help recognizers achieve higher recognition rates[1] and are required by recognition applications, such as speaker adaptation [Lee *et al*.1999][Chen *et al*.1990], speaker verifications identification [Siohan *et al*.1998]. The C-DSR platform provides a faster and more flexible way to construct various speech recognition functions for various mobile devices used in various recognition environments. One of the major missions of C-DSR is to increase the frequency of speech recognition use in daily life.

The C-DSR Server

From the viewpoint of the centralized server, the C-DSR server collects from all of the registrant clients speech utterances or formatted speech feature arrays along with their configuration tags. The basic idea is to formalize the life cycle of a speech recognition product/task from the deployment phase, to the diagnostic phase, tuning phase, and upgrading phase. Also, similar tasks can share corresponding information and adaptation data located on the server. The C-DSR server offers the following mechanisms to fully take advantage of these categorized speech and configuration data: (1) the C-DSR server can decide which recognition engine or which acoustic HMM model to employ according to the history log; (2) the C-DSR server can balance the trade-offs among communication bandwidth, system load and recognition accuracy; (3) the categorized and organized speech database can be utilized to create diagnostic tools that can be used to tune-up recognition engines and to perform all kinds

[1] For example, if we may provide gender information from speaker profile to speech recognizer, even a first-time speaker can obtain higher recognition accuracy.

of adaptation, such as speaker adaptation, channel adaptation [Siohan *et al.*1995] and background noise adaptation [Kristjansson *et al.*2001].

In summary, C-DSR is a generic speech recognition engine, in which all of the information, or parameters, concerning application-dependent user profiles and device profiles are kept in a configuration file which is initiated at the client side. In technical terms, C-DSR is a platform, while from customer's point of view, C-DSR is a personally owned, lifetime speech recognition engine. The C-DSR platform is embedded in the wireless network in contrast to conventional speech recognizers that are treated as input interfaces for portable devices (see Figure 1).

Overview

In the following, the architecture of C-DSR is described in section II. Then in section III, a demo system is presented to demonstrate the unique capability, that is, configurability, of C-DSR. Some conclusions are drawn in the final section.

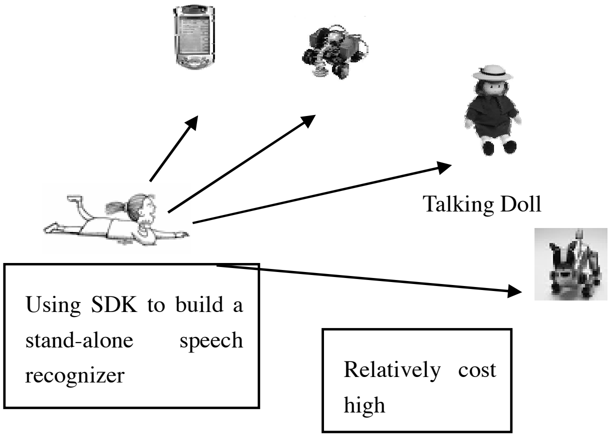


Figure 1(A). Conventionally, speech recognition is simply one of the available input interfaces for portable devices.

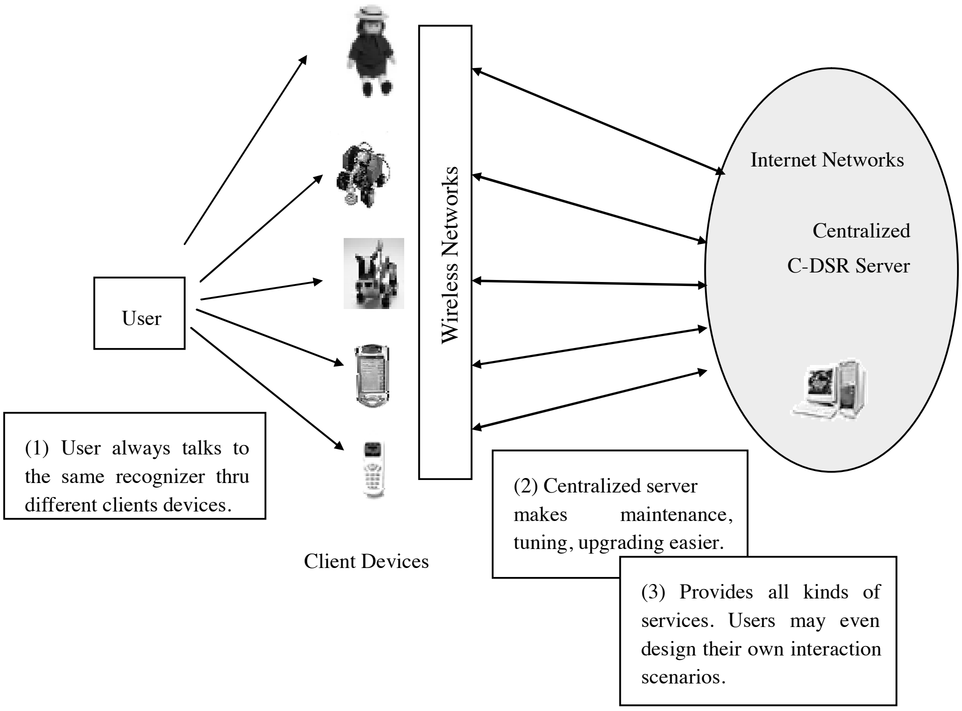

User
Client Devices
Wireless Networks
Internet Networks
Centralized
C-DSR Server
(1) User always talks to the same recognizer thru different clients devices.
(2) Centralized server makes maintenance, tuning, upgrading easier.
(3) Provides all kinds of services. Users may even design their own interaction scenarios.

Figure 1(B). Illustration of the innovative C-DSR platform.

2. The C-DSR Architecture

The function blocks of the C-DSR development platform are shown in Figure 2. A wireless device equipped with the C-DSR Client connects to a remote C-DSR Server using the C-DSR Protocol through a wireless network. The C-DSR Protocol sends speech data and parameters. The speech data can be in the form of PCM raw speech, ADPCM or pre-defined compressed speech features, depending on the computation power and bit-rate (communication bandwidth). The configuration file with speech data prepared by the client is, thus, transmitted by the C-DSR Protocol thru wireless networks. For now, the C-DSR Protocol is implemented on top of TCP/IP or RTP (Real Time Transit Protocol). After parsing the received protocol, the Configuration Controller (CC) decides how to configure the recognition engine (C-DSR Engine) and Dialogue System (DS) to accomplish the recognition task. The C-DSR engine and DS engine are composed of modulized components such that switches inside the engines can be shifted to corresponding components to perform the functionalities requested by the configuration. The recognition results are then logged and organized in the History Log Center

(HLC), resulting in a formatted package, or a database. The package is then passed to the Diagnostic Center (DC), where, diagnostic tools are used to tune-up the engines and provides adaptation data for various kinds of adaptation schemes, such as speaker/channel adaptation.

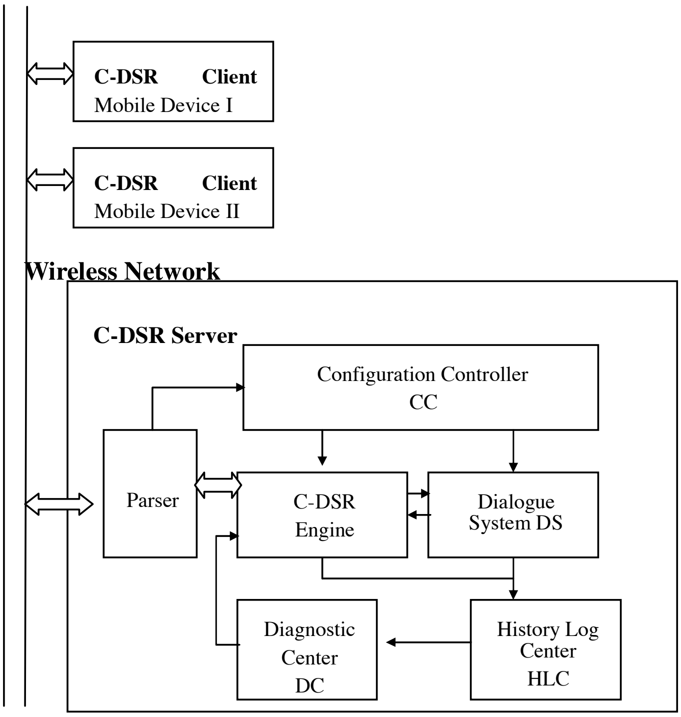


Figure 2. The function blocks of the C-DSR Platform.

A. Configuration File and Configuration Controller, CC

The configuration of the C-DSR platform is stored in a Configuration File (CF). Each field in the CF can be categorized into three attributes, SR, DSR, and Interactive Design-It-Yourself (I-DIY), which are explained below.

i. SR is the difference between the perfect and a current practical state-of-art speech recognition engine. The SR engine is never perfect. However, if we can restrict the speaking styles of the users, the recognition accuracy will be higher.

ii. DSR refers to those configurable parameters which can minimize the degradation due to wireless transmission loss.

iii. IDIY means Interactive Design-It-Yourself. We can never expect a machine to act exactly like a human being. The philosophy behind C-DSR is to make human-machine interaction simple and easy. The best way to achieve this goal is to involve users in design. Thus, we provide DIY tools to enable users to design their own ways of talking to machines.

Table 1. Configuration file

Configuration, Configurable Parameter		Attribute
SrCoder	Bit-rate: 0.1/1/4/16/64/128 Kbps	DSR
computationPower (deviceType)	very-thin/thin/medium/heavy	
Expandable	…	
NoiseType	home/office/street/in-vehicle	SR
microphonelType	US5/US20/US100/US200	
searchEngine	Full/Fast-mode	
voiceActivated	Yes/No	
endPointDetect	Yes/No	
speakingSpeed	Fast/medium/slow	
speakingAccent	Taiwan/china/foreigner	
Expandable	…	
vocabularySetUp	Vocabulary	I-DIY
grammarSetUp	Grammar (ABNF/cdsr_format)	
dialogueSetUp	Dialogue Script (VXML/AIML/cdsr_format)	
PersonalitySettings	Talktive/Quiet/Shy/…	
Expandable…	…	

The original design principle behind the C-DSR platform is to remotely configure the speech recognition engine on the server from the client side. It is the client device that initially prepares all of the configuration files. However, some of the configuration parameters may not be fully determined by the client device or may even be totally empty. In this case, the CC of the server should be able to append or modify these parameters by utilizing all available resources, including the historical configurations or statistics located in the server.

B. Configurable Engine

The configurable engine is the heart of the C-DSR server. As the name indicates, a configurable engine is an SR engine which is modulized and can be configured according to different requests requested from the various types of clients; Figure 3 shows the modules of the engine, which is a generalized SR engine on which state-of-art SR techniques can be employed. These typical modules (in a traditional/generalized SR engine) are listed below:

Table 2. Configurable modules in the C-DSR Engine with their parameters

Configurable Module	**Parameter**
Energy Normalization	None / FRAME_ENG_NORM
Front-end filter	FF_NONE / FF_LOW_PASS / FF_1POLE_1ZERO
Feature Extraction (if needed)	FE_NONE / FE_MFCC / FE_LPC_CEP / FE_8051_CLASS
End Point Detection	EP_NONE / EP_VFR / EP_ENG
Engine Type	EG_DIGITSTRING/EG_COMMAND/EG_KEYWORDSPOT/G_LVCSR
Mean Subtraction Computing	MS_NONE / MS_STD
HMM Adjustments	HJ_NONE / HJ_PMC
HMM Adaptations	HP_NONE / HP_ADAPT_DEV / HP_ADAPT_SPKR
Viterbi Searching	VS_FULL_PATH / VS_NBEST / VS_BEAM
Post Operations	PO_NONE / PO_STD

As shown in Table 2 above, each module has a well-defined interface and, for a particular module, several implementations are available. To each implementation, one CF name is attached, and it can be switched or configured. For instance, in the End Point Detection (EPD) module, there are three options, EPD_NONE, EPD_VFR and EPD_ENG, each representing a different algorithm used to implement the EPD function. The C-DSR platform also allows the system maintainer to adopt a new method for each module.

Intermediate data between modules are also generated to provide “symptoms” useful for diagnostic purposes. These symptoms include:

speech segmentation boundaries,

the Viterbi resulting path,

likelihood trajectories along the time axis on the resulting path,

a histogram of observations (feature vectors) for a particular Gaussian mixture,

a histogram of the observing likelihood of a particular HMM state

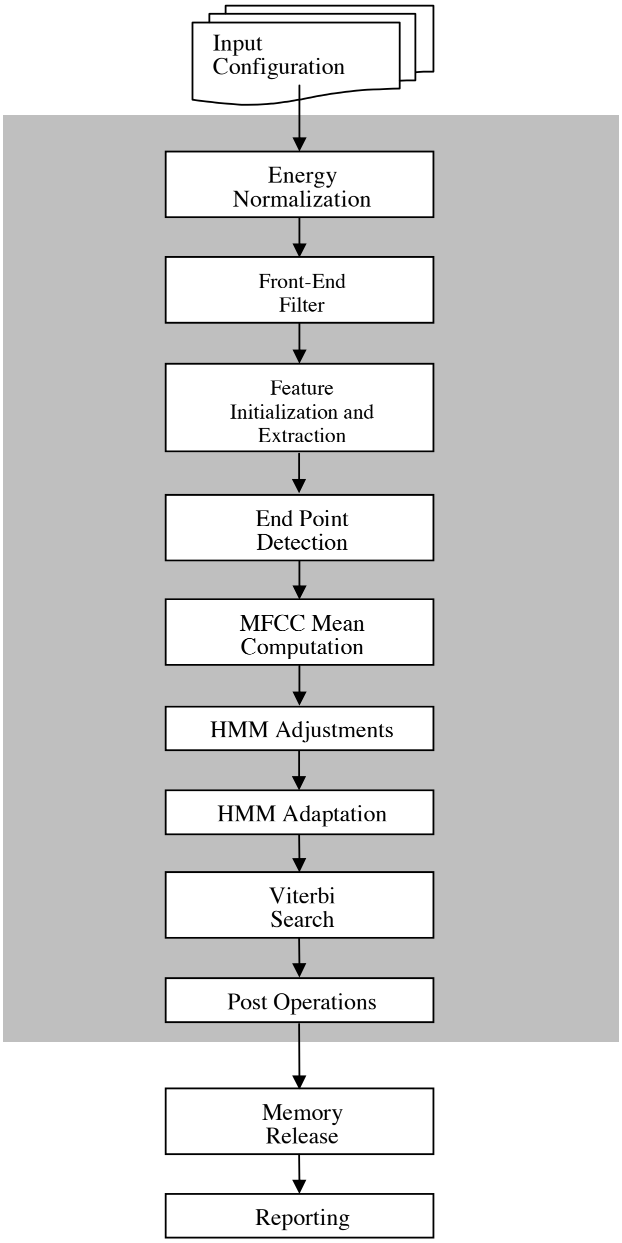


Figure 3. Modules of the C-DSR engine.

C. The Dialogue System, DS

A generic DS is, firstly, responsible for preparing a grammar, including vocabulary needed for the next speech recognition process. Then, the grammar with the incoming utterance is fed to the recognizer. The recognition result, a recognized key word, is then sent back to the DS. The DS then updates the "dialog status" records and determines the grammar for the next recognition.

Currently, we support AIML (Artificial Intelligence Markup Language, www.alicebot.org) and the simplified VoiceXML format used to describe dialogue scripts (see Figure 4).

D. The Diagnostic Center, DC

As described earlier, the so-called symptoms are intermediate data obtained while the engine is running and sent to the DC. The main purpose of the DC is to analyze and diagnose these symptoms in order to make suggestions. Thus, the C-DSR server is faced with various types of services, various environment configurations, and various types of client devices and speakers, so we want to make the engine core to be as generalized as possible. Currently, all of the diagnostics are done manually, which means that the DC only display to users or C-DSR server maintainers. We plan to make the DC automatically in the next generation of C-DSR.

E. The History Log Center (HLC)

The HLC is responsible for collecting and logging all of the corresponding information for each recognition service. The information collected includes speech utterances, formatted feature arrays, configuration files and the intermediate data, that is, symptoms and recognition results, and is saved to a corresponding user directory according to the user registration ID. The HLC serves as a database manager, whose job functions includes: (i) maintaining the database, if necessary, and creating a mechanism to eliminate garbage data; (ii) building data links to prepare adaptation data for various types of adaptation algorithms, for speaker or channel adaptation; (iii) preparing intermediate data for the DC to diagnose, so that the DC can provide data for the C-DSR engine to perform to tune algorithms and improve recognition accuracy.

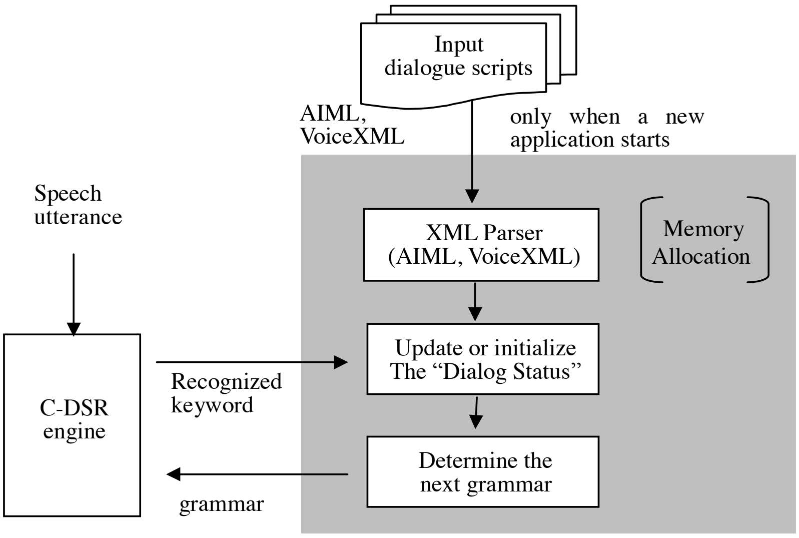


Figure 4. Diagram of the Dialogue System, DS.

3. C-DSR Demo System

In a laboratory, several speech recognition applications may be emulated by a PDA serving as a client on the C-DSR platform. These recognition applications are usually realized on stand-alone portable devices or server-based dialogue systems. Now, using the proposed C-DSR solutions, thin client devices can take advantage of powerful, wireless servers to perform sophisticated speech recognition functions (see Figure 5), including the following:

car agent – retrieving map/travel/hotel information through GPRS network in a car;

a personal inquiry system – a portable device which can retrieve stock/weather information anywhere through a GPRS network;

general-purpose remote control – in a WLAN 802.11b environment, a remote control which can be used to control a TV, stereo, air-conditioner, etc., through infrared rays by using natural language commands;

Sim-Librarian – a portable device, which, when a person walks into a library, can be used to ask for directions or for the location of a book the person is searching for.

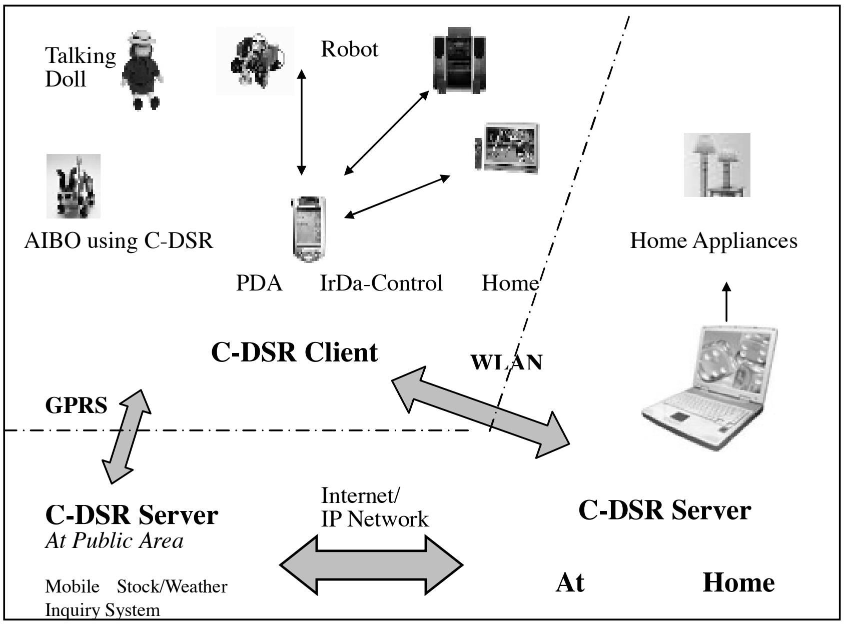


Figure 5. Illustration of the C-DSR implementation.

Each application uses its own configuration, specified according to (1) the device type: from thin to heavy, 8-bit 8051 class, DSP, PDA class; (2) the recognition style: command-based, natural, or dialogue; (3) the recognition environment: in a car, home, or open space. Two configuration files are presented below to illustrate how configuration files are used to realize a speech recognition application. Note that, normally, there are two types of speech recognition applications: voice command-based and dialogue style.

[Example 1] Voice-controlled home appliances

[Scenario] The user may use his wireless portable device with the installed C-DSR client to control a TV, lamp, or other home appliances within a WLAN environment.

[Configuration Settings]

1. Speech Feature Compression Format: this may be PCM, 8051-class, LPC-based Cepstrum, or MFCC (Mel-Frequency Cepstral Coefficients), depending on the computational cost and communication bandwidth (bit-rates) of the client device.
2. Environmental noise: this can be Quiet or Noisy. If this is skipped, the C-DSR Server will make a decision according to the history log.

3. Speaking speed: the speaking speed of a normal person is around five words per second. The user can determine his range of speaking speed, for instance, from three words per second to six words per second. If this is skipped, the C-DSR server will use default values.
4. Gender/Age/Accent: gender, age and accent information are very helpful for improving recognition performance. The C-DSR Client will retrieve and pass these pieces of information from the user/speaker profile to C-DSR Server for reference purposes. If this is skipped, the C-DSR Server will employ default models.
5. The Number of Results: the user may configure the number of recognition results, say N. The C-DSR Client will then display the first most likely candidates to the user.
6. Recognition Style: this can be Command-based or Dialogue. The grammar format for the Command-based style uses the ABNF format shown in the following:

```
#ABNF 1.0
$prefiller= 請 | 麻煩 | 你 | 我要
$action1= 開{open} | 關{close}
$keyword1= ( 燈{light} | 電燈{light} | 風扇{fan} | 電
風扇{fan} | 電視{tv} | 收音機{radio} )
```

[The Setup at the C-DSR Server]

When the configuration file and speech data are received from the client, the C-DSR Server performs recognition tasks according to the configuration. In the grammar example shown above, exactly one keyword from the $action1 group (open/close) and $keyword1 (light/fan/tv/radio) group will be generated. The Action Center on the C-DSR Server will perform a corresponding action, such as “turn on light.”

[Example No.2] Tourist Information Guide of Yu-Shan National Park

[Scenario] Users may use their own PDAs or smart phones to access this service when entering the area covered by the WLAN.

[Configuration Settings]

As in the previous case, we only need to change the field RecognitionStyle from Command-based to Dialogue-ProvidedByServer.

[The Setup at the C-DSR Server]

This example shows a dialogue system for a tourist guide. The content was provided by

Yu-Shan National Park. The C-DSR Platform provides several Dialogue Scripts. Here, we use VoiceXML as an example.

```
[CDSR_VXML]
<form_id=”tourinfo_agent”>
      <field name="hello">
        <prompt>您好，旅遊導覽精靈在此爲您服務</prompt>
        <grammar src="howRU.gram"    type="application/grammar+xml"/>
       </field>
      <field name="caragent">
        <prompt>很好，謝謝</prompt>
        <prompt>馬馬呼呼啦，謝謝</prompt>
        <prompt>可以啦，謝謝</prompt>
        <grammar src="tourinfo.gram" type="application"/>
        <filled>
        <if cond="caragent == 'resource'">
            <prompt>森林型態以柳杉，天然闊葉林樟樹，台灣杜鵑爲主</prompt>
            <prompt>動物型態有松鼠，穿山甲，台灣獼猴，台灣野兔等</prompt>
            <prompt>鳥類型態有綠繡眼，小鶯，巨嘴鴉，五色鳥等</prompt>
            <clear namelist="tourinfo.gram"/>
         </if>
         <if cond="caragent == 'facilities'">
            <prompt>遊客中心：內設餐飲部，會議室，多媒體簡報室及生態教育展示館</prompt>
            <prompt>餐廳：除可同時供一百人用餐外，並可作爲大型會議室，教室使用</prompt>
            <prompt>行政管理中心：爲本區工作人員處理行政事務的辦公地點</prompt>
            <clear namelist="tourinfo.gram"/>
         </if>
         <if cond="caragent == 'spot'">
             <prompt>化石區：此生痕化石是大約三萬年前蝦，蟹類進行築穴工事時所遺留而成
                     </prompt>
             <prompt>造林紀念石：爲前新竹山林管理所大溪分所，在民國四十四年爲紀念
                     東眼山造林工作實績而建</prompt>
             <prompt>親子峰：在林道終點上方，有大小雙峰，猶如慈母帶著小孩，故名親子峰
                     </prompt>
            <clear namelist="tourinfo.gram"/>
         </if>
         </filled>
       </field>
      <block>
         <submit next="theTop.vxml" namelist="city state"/>
         <prompt>好吧，掰</prompt>
         <prompt>好吧，下次再聊囉，掰</prompt>
      </block>
</form>
```

4. Conclusions

Speaking of wireless mobile devices, conventionally, speech recognition is considered to be one of the available input methods for these devices. In this paper, we have presented the client-server C-DSR platform which is a centralized speech recognition server embedded in the wireless infrastructure. By using C-DSR, people talk to the same lifetime speech recognition system. Speech data and the corresponding configuration, which keeps all the records about recognition environments, device information, dialogue scripts, recognition results, and so on, will not be abandoned when particular applications or services are discontinued. The speech recognition server provides many types of services for all classes of wireless mobile devices, and these services continuously acquire speech data from all locations, and adapt the engine performance all the time.

Personalization and humanization are essential. We have seen many successful products come on the market. A humanized device does not have to be "intelligent." As long as it "looks" intelligent and people find it interesting, we do not really need to make such a machine act exactly like a human being. People like to have their own ways to interact with their own personal devices/pets. Perhaps the Design-It-Yourself approach, getting people involved in the design process, is one good solution, and the "configurability" of C-DSR can surely provide such a platform to meet these needs.

References

ETSI Doc. No. ES 201 108, Ref. RES/STQ-00018, STQ Aurora, "DSR Front End".

ETSI ES, Version 0.1.1, Ref. DES/STQ-00030, STQ Aurora, "Front-End Extension for Tonal Language Recognition and Speech Reconstruction".

Baker, S. , "A Tale of A Bubble, "Business Week Magazine, International Cover Story, June 3, 2002.

Reinhardt, A. , W. Echikson, K. Carlisle, P. Schmidt, "Who Needs 3G Anyway?" Business Week Magazine, International – European Business, March 26, 2001.

Yamaguchi, H. , K. Suzuki, C. V. Ramamoorthy, "The Humanization, Personalization and Authentication Issues in the Design of Interactive Service System,“ 2003 Society for Design and Process Science, www.sdpsnet.org.

Deng, L. , J. Droppo, and A. Acero, "Recursive Estimation of Nonstationary Noise Using Iterative Stochastic Approximation for Robust Speech Recognition,“ in IEEE Transactions on Speech and Audio Processing. Volume: 11 Issue: 6 , Nov 2003.

Wu, J. , J. Droppo, L. Deng and A. Acero, "A Noise-Robust ASR Front-End Using Wiener Filter Constructed from MMSE Estimation of Clean Speech and Noise,“ in Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding. Virgin Islands, Dec, 2003.

Lee, C. H. , "On stochastic feature and model compensation approaches to robust speech recognition," Speech Communication, 25:29-47, 1998.

Digalakis, V. , L. Neumeyer and M. Perakakis, "Quantization of Cepstral Parameters for Speecg Recognition Over the World Wide Web," IEEE Journal on Selected Areas in Communications, Jan. 1999, volume 17, pp 82-90.

Lee, C. H. , C. H. Lin, and B. H. Juang, "A study on speaker adaptation of continuous density HMM parameters," Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pages 145-148, Albuquerque, New Mexico, April 1990. ICASSP'90.

Chen, K. T. and Hsin-min Wang, "Eigenspace-based Maximum A Posteriori Linear Regression for Rapid Speaker Adaptation," in Proc. IEEE Int. Conf. Acoustics, Speech, Signal processing (ICASSP'2001), Salt Lake City, USA, May 2001.

Siohan, O. , A. E. Rosenberg and S. Parthasarathy, "Speaker identification using minimum classification error training," In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Seattle, Washington, USA, May 1998.

Siohan, O. , Y. Gong, and J. P. Haton, "Channel adaptation using linear regression for continuous noisy speech recognition," IEEE Workshop on Automatic Speech Recognition, Snowbird, Utah, USA, December 1995.

Kristjansson, T. , B. Frey, L. Deng and A. Acero. "Towards Non-Stationary Model-Based Noise Adaptation for Large Vocabulary," in Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing. Salt Lake City, Utah, May, 2001.