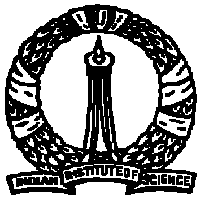
# Automatic Language Identification based on Acoustic Sub-word units

A Dissertation
Submitted in Partial Fulfilment of the
Requirements for the Degree of
**Master of Technology**
in Industrial Electronics

By
**Mone Thirumala Raju**




Department of Electronics and Communication Engineering
National Institute of Technology Karnataka, Surathkal
P.O. Srinivasnagar - 575 025
India

February 2003

# Acknowledgements

This stylistic preference is a bit unfair, however since there have actually been quite a few people involved in seeing these results and reaching these conclusions. This seems like a good place to thank these people.

At a risk of being vague, it suffice to say that my attention has been divided and detracted by various diversions, academic and otherwise. I owe a huge debt of gratitude to Professor T. V. Sreenivas for helping me put and keep a focus on my work. His guidance, regardless of studies, has been invaluable, and collaborating with him has been both privilege and a joy.

I would also like to thank Dr. Ramasubramanian for his unflinching support of my work. He opened the door to this world of possibilities and gave me a freedom to explore it at will. In addition, his standards of quality and clarity have been nothing short of inspirational.

I would like to thank Dr. Sumam David, who had taught me Speech Processing and at the same I also like to thank all faculty members of NITK - Surathkal and too my class mates.

Furthermore, I would like to thank the Speech and Audio Group (SAG) members for their support during my stay in the IISc. Along the same lines, I am grateful to Sai Jayram for his intruding help and too Chandrasekhar without whom Indian Language Database (ILDB) collection might have been impossible. Finally, it would be a drastic omission if this account did not pay proper homage to the people who have helped me shuffle through the everyday stuff; thanks especially to my colleague Vishweshwar, Vamshi, Ramachandra, Mahesh and last but most of all, my family.

# Contents

# List of Figures

# Abstract

Automatic language identification (LID) is an important research problem with application in multi-lingual speech input-output systems. Several promising solutions for LID have been reported over the last decade. Among these, the phone recognition based approach is one of the more effective methodology for LID. We identify three main frameworks in phone recognition approaches, namely, Parallel Phone Recognition (PPR), Phone Recognition and Language Modeling (PRLM) and Parallel-PRLM (P-PRLM). All these three systems suffer from the limitation of requiring phonetically labeled data for the training of the front-end phone recognizer (PR). In this dissertation, we circumvent this problem by using a sub-word recognizer (SWR) in the place of phone recognizer. The SWR is obtained from training data without phonetic transcription in any of the languages in the LID task. The SWR performs a front-end recognition in terms of sub-word units, which are obtained by automatic segmentation , segment clustering and segment HMM modeling. This also provides means of controlling the resolution of the acoustic space represented by the sub-word units which can be made to match broad phonetic units or sub-phonetic units.

In this dissertation we study three sub-word based systems, namely, Parallel Sub-word Recognition (PSWR), Sub-word Recognition and Language Modeling (SWRLM) and Parallel-SWRLM (P-SWRLM), which are the sub-word equivalents of PPR, PRLM and P-PRLM systems. We reported results for these systems and show that the sub-word based system can perform as well as the phone based systems.

# Chapter 1

# Introduction

Automatic language identification (LID) has become an important research problem over the last decade with several promising solutions [16], [34]. An $N$ language LID task is to classify an input speech (typically spoken by an unknown speaker and of unknown text) as belonging to one of $N$ languages $\mathcal{L}_1, \ldots, \mathcal{L}_N$. Over past three decades, significant effort has been focused on the automatic extraction of information from speech signals. Many techniques were aimed at obtaining either a transcription of speech signal or an identification of the speaker's identity and gender [33]. More recently, LID research has been experiencing a renaissance, spurred by research activities in multilingual speech recognition and understanding for which an efficient means for identifying the language being spoken has definite benefits.

There are several important applications for automatic language identification [15] [33] [34]. For example, consider an automatic multi-lingual voice controlled travel-information retrieval system, which accepts voice as the only input and from which it identifies the language of an unknown speaker and also responds to him/her in that language, which is an ultimate goal of multi-lingual speech dialog systems. Telephone companies can provide better services to customers speaking different languages if an LID front-end can route calls to appropriate operators with various applications such as, handling emergency calls, checking into hotel, arranging meeting for non-native speakers. More importantly, current research into multilingual speech recognition and synthesis system will benefit directly with the use of an LID front-end for further processing of the input speech by multilingual speech-to-speech translation systems.

## 1.1 Perceptual experiments

As with speech recognition, humans are the most accurate language identification systems. Within seconds of hearing speech, people are able to determine whether it is a language they know. Even they can make subjective judgments if they are not familiar with the language spoken. e.g., "Sounds like German"

Humans are able to identify languages using very short excerpts of speech, drawing upon several different sources of information. The fact that humans are so adept at language identification task, illustrates the considerable gap between our perceptual capabilities and our attempt at automating them. If we are to shorten this gap, it is imperative that we study human performance on LID task.

The understanding we get from the perceptual experiments done by Muthusamy [16] is that many listeners learn to discriminate among languages by using combination of phoneme and word spotting strategies and prosodic cues. These experiments also showed that increased exposure to each language and longer training sessions contribute to improved identification performance. Listeners who knew more languages tended to perform better, on the average, than subjects who knew just one language. So, priori knowledge definitely helped and the subject have learned to develop their own cues as the experiment progressed.

## 1.2 LID cues

In mono-lingual spoken language identification systems, the objective is to determine the content of the speech, typically realized by phoneme recognition, word recognition and sentence recognition. This requires that systems cue in on small portions of speech-frames, phonemes, syllables, sub-word units, and so on, to determine what the speaker said. In contrast, in content-independent language identification, phonemes and other sub-word units alone are not sufficient cues, since several phonemes and syllables are common across different languages. One also needs to examine the sentences as a whole to determine the "acoustic signature" of the language, the unique characteristics that make one language

sound distinct from another.

The basic LID cues which are used to decode the "acoustic signature" are,

- **Acoustic Phonetics:** Phonetic inventories differ from language to language. Even if languages have identical phones, the frequency of occurrence of phones differ across languages.
- **Prosodics :** Duration characteristics, pitch contours and stress patterns are different from one language to another.
- **Phonotactics :** Phonotactics refer to the rules that govern the combination of the different phones in a language. There is a wide variance in phonotactic rules across languages.
- **Vocabulary :** Conceptually, the most important difference between languages is that they use different sets of words. That is, their vocabularies differ.

A successful LID algorithm would exploit information from all of the above cues to arrive at its identification decision.

## 1.3 LID approaches/systems

Fig 1.1 shows the two phases of LID. During the "training" phase, the typical system is presented with examples of speech from a variety of languages. Each training speech utterance is converted into a stream of feature vectors. These feature vectors are computed from short windows of the speech waveform (e.g. 20 ms) during which the speech signal is assumed to be stationary. The feature vectors are computed regularly (e.g. every 10 ms) and contain spectral or cepstral information about the speech signal (the cepstrum is the inverse Fourier transform of the log magnitude spectrum; it is used in many speech processing applications). The training algorithm analyzes a sequence of such vectors and produces one or more models for each language. These models are intended to represent a set of language dependent, fundamental characteristics of the training speech to be used during the recognition phase of the LID process.

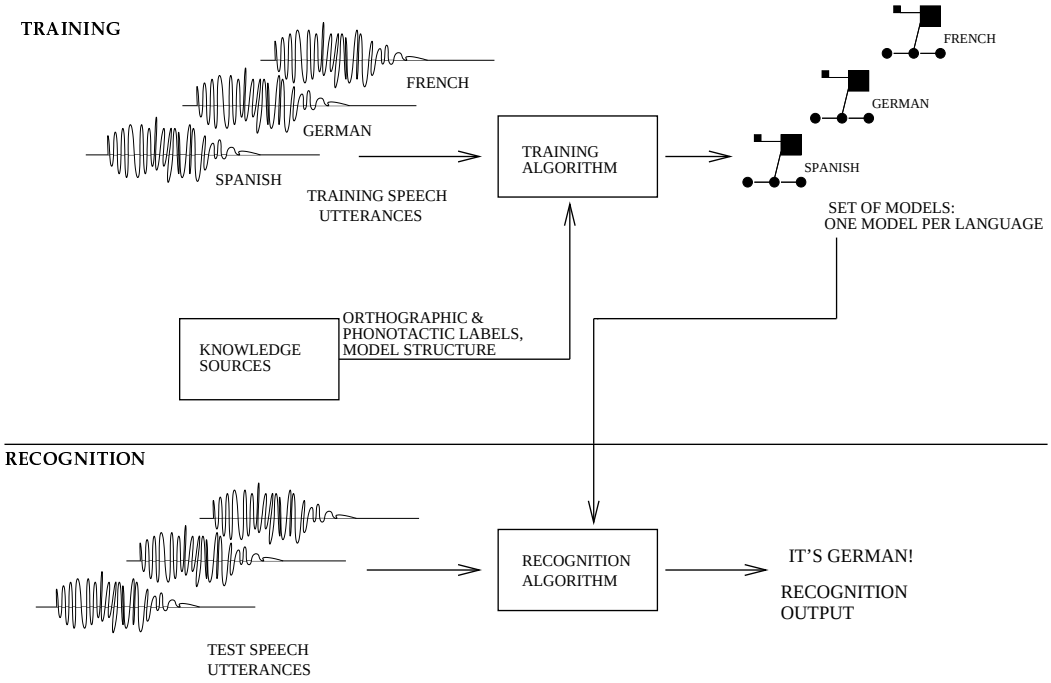



Figure 1.1: General LID system

During the "recognition" phase of LID, feature vectors computed from a new utterance are compared to each of the language-dependent models. The likelihood that the new utterance was spoken in the same language as the speech used to train each model is computed and the maximum-likelihood model is found. The language of the speech that was used to train the model yielding maximum likelihood is hypothesized as the language of the utterance.

We can categorize much of LID work [32], [6], [16], [34] into the following main approaches

- Spectral similarity approaches
- Prosody based approaches
- Phone-recognition approaches
- LID using multi-lingual speech units
- Word level approaches

- LID based on continuous speech recognition

### 1.3.1 Spectral similarity approaches

Spectral similarity approach in LID is one of the earliest in the design of automatic LID systems. Here, LID is performed by computing the difference in spectral content among languages, exploiting the fact that speech spoken in different languages contains different phonemes and phones. To train these systems, a set of prototypical short-term spectra were computed and extracted from training speech utterances. During recognition, test speech spectra were computed and compared to the training prototypes. The language of the test speech was hypothesized as the language having training spectra that best matched the test spectra.

There were several variations on this spectral similarity theme. The training and testing spectra could be used directly as feature vectors, or they could be used instead to compute formant-based or cepstral feature vectors. The training examplers could be chosen either directly from the training speech or could be synthesized through the use of $K$-means clustering. The spectral similarity could be calculated by the Euclidean, Mahalanobis, or some other distance metric [34], [21].

### 1.3.2 Prosody based approaches

Prosody based approach is based on the features that carry prosodic (e.g., pitch, duration, intonation) information. This method was motivated because studies revealed that humans can use prosodic features for identifying the language of speech utterances. Here, parameters were designed to capture pitch and amplitude contours on a syllable-by-syllable basis. They were normalized to be insensitive to overall amplitude, pitch and speaking rate. The accuracy of these systems of any particular set of feature is highly language-pair specific [25],[16].

### 1.3.3 Phone-recognition approaches

Phone-recognition approach is derived from the fact that different languages have different phone inventories. LID systems can hypothesize exactly which phones are being spoken as a function of time and determine the language based on the statistics of that phone sequence. To make these systems easier to train, one can use single language phone recognizer as front-end to a system that uses phonotactic scores to perform LID.

The novelty of these phone based systems is that we can incorporate more of knowledge into the LID system. Three main frameworks can be identified within the phone recognition approaches [34], [33], [32], [6], [16] viz.,

- Phone recognizer followed by Language Modeling (PRLM)
- Parallel- PRLM (P-PRLM)
- Parallel Phone Recognizer (PPR)

### 1.3.4 LID using multi-lingual speech units

Instead of training language-dependent phoneme recognizer multi-lingual speech units can be built [34]. These are derived by either a mixture of language-dependent and language-independent phones [19], [30] or by deriving tokens automatically from training data can be used instead of phone recognizers. Advantages of this approach include data sharing and discriminant training between phonemes across languages and easy bootstrapping to unseen languages.

Research has also focused on the problem of identifying and processing only those phones that carry the most language discriminating information. These phones are known as mono-phones [34], [19]. The language-independent phones called as poly-phones [20], can be trained on data from more than one language without loss of accuracy.

### 1.3.5 Word level approaches

Word level approach [34] to LID falls between phone level systems described previously and the large-vocabulary speech recognition systems [22]. These systems use more sophisticated

sequence modeling than the phonetic models of the phone-level systems. Even though these systems incorporate more knowledge into the LID system they require huge training data.

### 1.3.6 Continuous speech recognition

By adding even more knowledge to the system, researchers hope to obtain even better LID performance [21], [34], [22]. In such systems, one speech recognizer per language is created in training phase. During testing, each of these recognizer is run in parallel, and the one yielding output with highest likelihood is selected as the winning recognizer. The language used to train that recognizer is the hypothesized language of the utterance. Such systems hold the promise of high-quality identification, because they use higher-level knowledge; like words and word sequences rather than phone and phone sequences to make LID decision.

## 1.4 Present work

Among various LID approaches/systems, the phone recognition approach offers considerable promise, as it incorporates sufficient knowledge of the phonology of the language to be identified, without incurring the significantly higher cost of word-based approaches. The front-end in all the three phone based approaches, have to be trained on manually labeled training data. In PPR, the front-end PR for each language in the task has to be trained from labeled training data for all the $N$ languages in the LID task. A PPR system is therefore is the most difficult to implement and also best in LID performance.

It is important to note that labeled training data is difficult to obtain. Manually transcribing a database in any language to obtain phonetic labels is a time consuming, tedious, error prone and expensive process. It requires skilled linguists in each language to be labeled.

We propose sub-word based LID systems, which does not require manually labeled data in any of the languages to be recognized. The sub-word recognizer (SWR) used in the PSWR system can be obtained from training data without phonetic transcription in any of the languages in the task. The new approach performs a front-end tokenization in terms of sub-word units which are designed by automatic segmentation, segment clustering and

segment HMM modeling. The SWR can replace the front-end phone recognizer (PR) in the PPR system as well as in the PRLM and P-PRLM systems which constitute two other well accepted frameworks in LID system design. This allows easy expansion of these systems to a large number of languages without requiring tedious manually labeled training speech data in any of the languages in the task.

In this thesis we will be discussing about different sub-word LID systems namely Parallel Sub-word Recognizer (PSWR), Sub-word Recognition and Language Model (SWRLM) and Parallel-SWRLM (P-SWRLM). The PSWR is an alternative to the PPR system. In PSWR we will have $N$ SWRs for $N$ language task and each SWR has a language dependent sub-word unit inventory (SWUI). The SWR and the back-end language model (LM) of sub-word unit sequences are tokenized by the front-end SWR. PSWR yields three types of scores, namely, the acoustic score, joint acoustic-language score and the language model score. PSWR with different segment rates and sub-word unit inventory sizes is also discussed which will reveal the effects of different time resolutions and granularity. A PSWR requires SWUs in all the languages in LID task but SWRLM will have only one front-end with language dependent SWUs which will also be discussed in this thesis with all segment rates and sub-word unit inventory sizes. SWRLM may not span the entire acoustic space so we can run multiple SWRLMs in parallel or we can use language independent SWUI. In this thesis we also focus on th use of multiple SWRLM, termed as Parallel-SWRLM (P-SWRLM).

Parallel-SWRLM (P-SWRLM) differ from SWRLM and PSWR in the way the front-ends are used. In P-SWRLM, we will have more than one SWRLMs in parallel. This parallelism is studied across languages, time resolutions and sub-word unit inventories. We also discuss about various classifiers like maximum likelihood classifier and Gaussian classifier. All the experiments are done using OGI-TS database. We evaluate PSWR, SWRLM and P-SWRLM systems using 6 languages of the OGI-TS database.

## 1.5 Dissertation overview

The rest of the thesis is organized as follows. Chapter 2 describes the multi-lingual speech corpus for Indian languages collected as a part of this thesis. This is to facilitate further research in LID on Indian languages. Chapter 3 describes Parallel sub-word recognition (PSWR) system in detail. The chapters 4 discusses Sub-word recognition with language modeling (SWRLM) and chapter 5 discusses P-SWRLM system. Chapter 6 gives scope for further research and Chapter 7 give the conclusions.

# Chapter 2

# Speech corpus development for Indian Languages

Research in automatic language identification requires a large corpus of multi-lingual speech data to capture the many sources of variability within and across the languages [15]. These include variability due to speakers of different ages, gender, dialect, microphones, background noise and the language being spoken. It is also important that the corpus contain a wide variety of speech from each speaker, like fixed vocabulary to natural, continuous speech. This makes it useful for both content-dependent and content-independent language identification.

The availability of such a corpus in the public-domain would enable researchers to study languages and to develop and compare multi-language recognition algorithms. Unfortunately, there was no such corpus of data available for Indian languages. Consequently, a significant part of this dissertation was also devoted towards collecting and developing a multilingual speech corpus for Indian languages. This was taken up after a close study of speech corpuses such as OGI-TS and TIMIT databases.

This chapter describes the OGI-TS database and the Indian language high quality speech corpus, which contain a mix of fixed-vocabulary utterances and natural continues speech.

## 2.1 OGI-TS database

The Oregon Graduate Institute Multi-language Telephone Speech Corpus (OGI-TS) was designed specifically for language ID research [15] [16]. It currently consists of spontaneous and fixed-vocabulary utterances in eleven languages: English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil and Vietnamese. These utterances were produced by 90 native speakers in each language over telephone lines. These utterances ranged in duration from one second to 50 seconds, with an average duration of 13.4 seconds. Hindi is a recent addition to this corpus. Among the eleven languages, phonetic transcription is available for six languages - English, German, Hindi, Japanese, Mandarin and Spanish.

## 2.2 Indian language high quality speech corpus

The acquisition of a multi-language high-quality speech corpus poses several unique problems. The collection process was slow and required considerable human supervision. The selection of languages is heavily dependent on the availability of native speakers of that language in the vicinity of the recording laboratory.

High quality speech is recorded in an anechoic chamber, and is digitized at a high sampling rate of 16 kHz which preserves the high-frequency information associated with obstruents like stops, nasals and fricatives in speech.

### 2.2.1 Language selection

Among various Indian Languages we have selected six languages: Hindi, Kannada, Malayalam, Marathi,Tamil and Telugu. English, as spoken by Indian speakers is the seventh language. The choice of languages is based on the most commonly used languages across India and availability of native speakers of the languages in the Bangalore metropolitan area, particularly in the IISc campus. Hindi is included as it is India's national language, Marathi is included because it is widely used in North-western part of India. Among the 6 Indian languages four are Dravidian languages widely spoken in Southern India and some

part of South-east Asia.

### 2.2.2 Speakers

For each language, twenty adult speakers of different age, gender are selected. Care was taken to ensure that a speaker was chosen for Indian language, only if he/she had that language as a native language particularly during childhood. The age of female speakers ranged from 20 to 45 years while those of the males ranged from 22 to 50 years. There were approximately equal number of male and female speakers in each language.

### 2.2.3 Recording Equipment

Speech was collected using a Sennheiser HMD 224 noise-canceling microphone and low pass filtered at 7.6 kHz. A PC was programmed to say prompts in each of the seven languages and to collect speech samples at 16 kHz at 16-bit resolution with gain control so that the digitized speech covers the full dynamic range. The recording was done in an anechoic chamber with no background noise. With this recording equipment, the observed a posteriori and peak segmental SNRs are 32dB and 64dB respectively.

### 2.2.4 Recording protocol and data acquisition

The recording protocol was designed to obtain

1. **Useful vocabulary**
    - Personal details with name, age, native language, profession and about family (30 seconds)
    - Digits and days of the week (40 seconds) (31 words)
    - Numbers in English (40 seconds) (24 words)
    - English alphabets and special characters (50 seconds) (30 words)
    - Commonly used words in native language (40 seconds) (20 words)
2. **Task specific words (fixed-vocabulary)**

- Railway reservation words in native language (40 seconds) (20 words)
- Railway reservation words in English (40 seconds) (20 words)
- Banking words in native language (40 seconds) (20 words)
- Banking words in English (40 seconds) (20 words)

3. **Elicited free speech**

- Passage reading part-1 in native language (60 seconds)
- Passage reading part-2 in native language (60 seconds)
- Picture description in native language (60 seconds)
- Passage reading in English (60 seconds)
- Passage reading in Hindi (60 seconds) (Optional)

Elicited free speech was obtained by asking him/her to read 2 paras of duration 1 minute each in his native language and giving a picture of his choice and asking him/her to describe for maximum of 1 minute. Every speaker has to read an English passage which is of 1 minute duration and if speaker is fluent in reading Hindi he can also have Hindi passage reading of 1 minute, which is an optional because so many speakers are not fluent in reading Hindi even though they speak. The duration of data recorded for one speaker is approximately 8 minutes. After recording 8 minutes of speech for a single speaker all digitized files were played back and judgments were made on the quality and content of speech in each utterance. If it was not satisfactory manner then the speaker was requested to repeat the recording.

### 2.2.5 Recording system

The system we are using is an Interactive Voice Recording System with Voice Activity Detection (IVRS-VAD). This system plays prompts and after listening to these prompts speaker has to start speaking with an initial pause of 3 seconds. During the initial 2 seconds of recording it calculates a threshold ($\epsilon$) and $1.25\epsilon$ is kept as silence-to-speech threshold.

The IVRS-VAD calcultates threshold $\delta$ from one second of speech samples and if $\delta$ is greater than the silence-to-speech threshold $1.25\epsilon$, then it is considered as speech; else those samples are neglected and thrown as silence regions. If such silence regions were found successively then it is replaced by the most recent one second silence region; recording for that particular prompt is automatically terminated if there are five such successive silence regions. The flow chart of the IVRS-VAD is shown in Figure 2.1

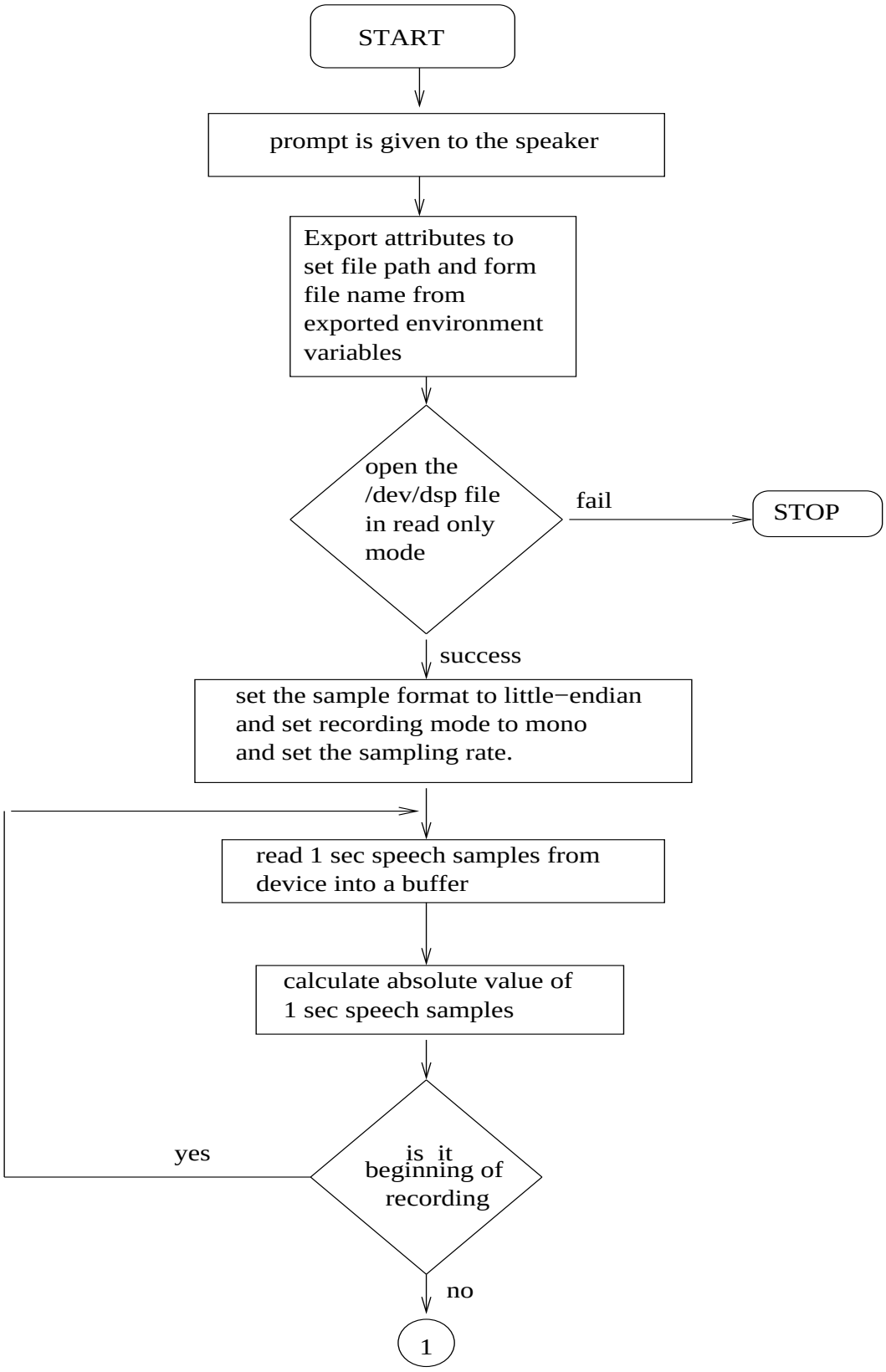



Figure 2.1: Flow chart of IVRS-VAD type recording system continued

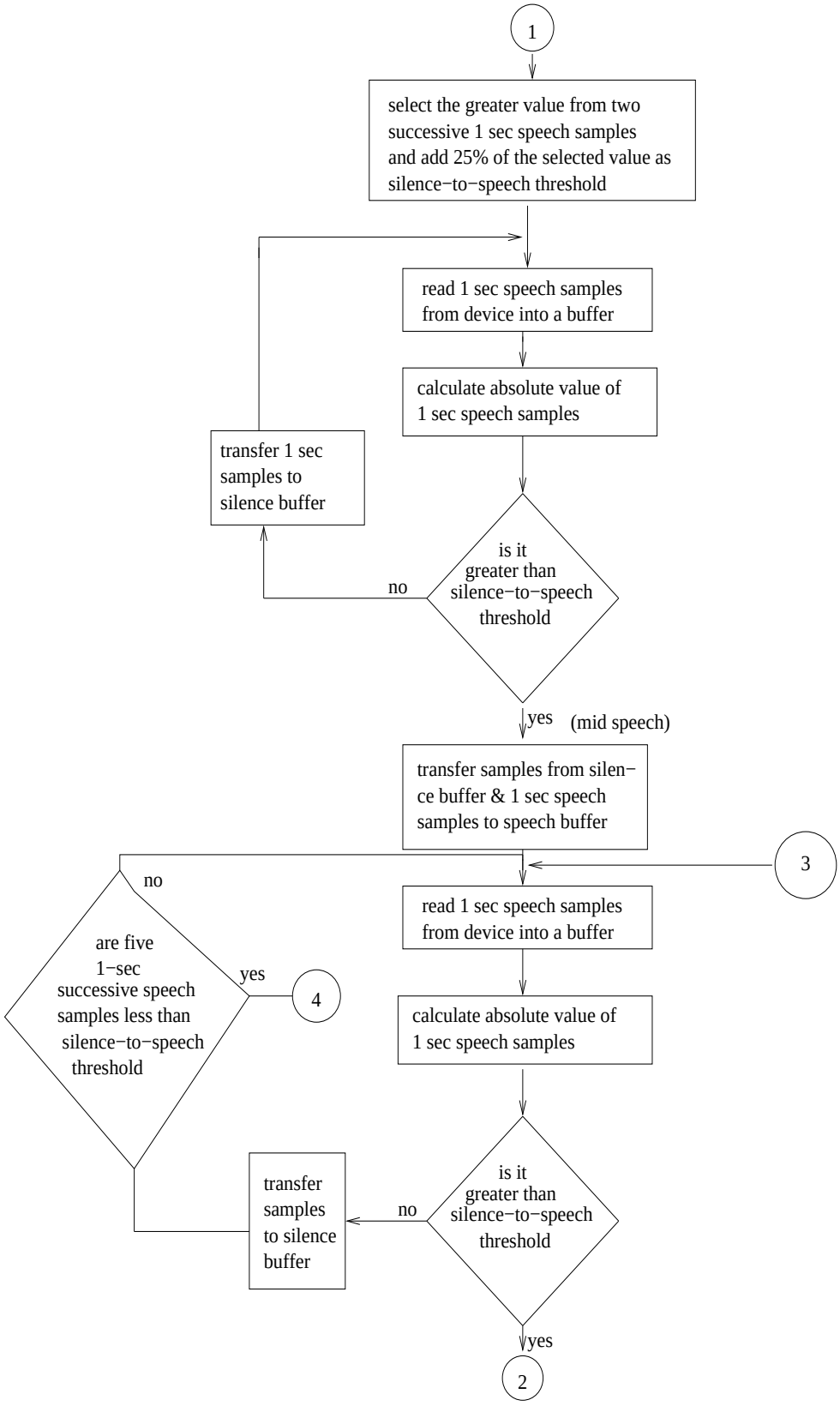



Figure 2.2: Flow chart of IVRS-VAD type recording system continued

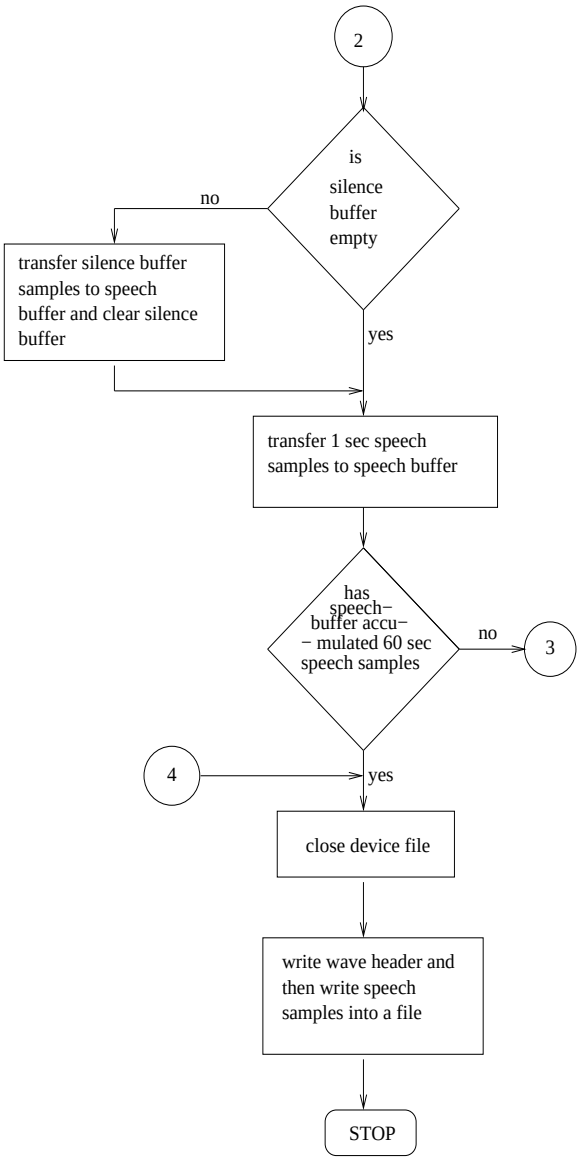



Figure 2.3: Flow chart of IVRS-VAD type recording system

# Chapter 3

# Parallel sub-word recognition (PSWR)

## 3.1 Introduction

Automatic language identification (LID) has become an important research problem over the last decade with several promising solutions [16], [34]. Among the various approaches to LID [34], the phone-recognition approach offers considerable promise, as it incorporates sufficient knowledge of the phonology of the languages to be identified. There are at least three typical configurations of phone recognition approach for LID. These are i) Phone Recognition followed by Language Modeling (PRLM), ii) Parallel PRLM (P-PRLM) and, iii) Parallel Phone Recognition (PPR) systems [33]. Most systems reported recently in the literature fall within these categories [3], [4], [5], [6], [29], [30], [31], [32], [26], [16], [34].

One of the main frameworks in the phone-recognition approaches is Parallel Phone Recognition (PPR) [33]. An $N$ - language LID task is to classify an input speech utterance as belonging to one of $N$ languages $\mathcal{L}_1, \ldots, \mathcal{L}_N$. The PPR system for this task has $N$ paths, each with a front-end phone-recognizer (PR) followed by a back-end language model (LM) in that language. The front-end PR tokenizes the input utterance into a sequence of phone symbols. The back-end LM performs phonotactic analysis on the resulting phone sequence. Phonotactics refers to the language-dependent constraints on the sequences of phones and is modeled by an $n$-gram analysis, with typical systems using a bigram ($n = 2$) statistics. An input utterance is classified by a maximum likelihood decision on the $N$ scores obtained by the front-end PR [8] or by the back-end LM or jointly by PR and LM of each language

[33], [27].

In the PPR system, the front-end PR has to be trained on phonetically labeled data, usually obtained by manual labeling, to generate a 'phone HMM inventory' of that language. Thus, a PPR requires labeled training data for all the $N$ languages in the task to train each of its $N$ front-end PRs. A PPR system is therefore the most difficult to implement [33], [27]. It is important to note that labeled training data is difficult to obtain. Manually transcribing a database in any language to obtain phonetic labels is a time-consuming, tedious, error-prone and expensive process. It requires skilled linguists in each language to be labeled.

Recently, a sub-word based LID system called 'parallel sub-word recognition' (PSWR) has been proposed [8], [10] which operates in a PPR framework but without requiring manually labeled phonetic data in any of the languages in the task.

We study the PSWR system with both the front-end sub-word recognizer (SWR) and the back-end language model (LM) of sub-word unit sequences tokenized by the front-end SWR. The resulting PSWR system can yield three types of scores, namely, the acoustic score, joint acoustic-language score and the language model score. We examine the effectiveness of these scores for a LID task of 6 languages in the OGI-TS database. Considering various combinations of the statistical evaluation scores, it is found that PSWR can perform as well as PPR, even with broad acoustic sub-word tokenization, thus making it an efficient alternative to the PPR system.

### 3.1.1 Advantages of SWR over PR

The sub-word recognizer (SWR) used in the PSWR system can be obtained from training data without phonetic transcription in any of the languages in the task. The new approach performs a front-end tokenization in terms of sub-word units which are designed by automatic segmentation, segment clustering and segment HMM modeling. The SWR can replace the front-end phone recognizer (PR) in the PPR system as well as in the PRLM and P-PRLM systems which constitute two other well accepted frameworks in LID system design. This allows easy expansion of these systems to a large number of languages without

requiring tedious manually labeled training speech data in any of the languages in the task.

We list below the three frameworks with the use of a SWR in the place of PR in each of the three PR frameworks PPR, PRLM and P-PRLM.

1. **PSWR:** It allows extendibility of the system to large number of languages without requiring labeled training data in any language; this results in easy incorporation of new languages in the LID task.

2. **SWRLM:** This refers to Sub-word recognition followed by language modeling. this is the sub-word equivalent of the conventional PRLM [33]. It helps a PRLM framework by allowing design of a single front-end SWR obtained either in a language dependent manner or a language independent manner from a collection of several languages encompassing the acoustic-phonetic space of all the languages in the task. A single multi-language front-end SWR is better than a single-language front-end SPR, in terms of coverage of acoustic-phonetic space and the resultant sub-word label sequence as seen by each of the language models in the task.

3. **P-SWRLM:** This refers to Parallel-SWRLM and is equivalent of P-PRLM [33]. It allows a P-PRLM system to have as many front-end SWR systems as possible; a large number of $M$ ($M < N$) front-end SWRs gives adequate coverage of the acoustic space without requiring labeled training data in any language.

In this context, it has to be noted that Tucker [26] has bootstrapped PPR systems by using labeled training speech in only one language (TIMIT for English). This system is not truly sub-word based, in the same sense used in this thesis as well as in the work on sub-word based speech recognition systems [12], [11], [23], [18], [1]. It should also be noted that the work reported earlier in [14] addresses the issue of using untranscribed training data for LID and is similar to the sub-word based approach discussed here in making use of automatically derived acoustic units based on automatic segmentation and clustering.

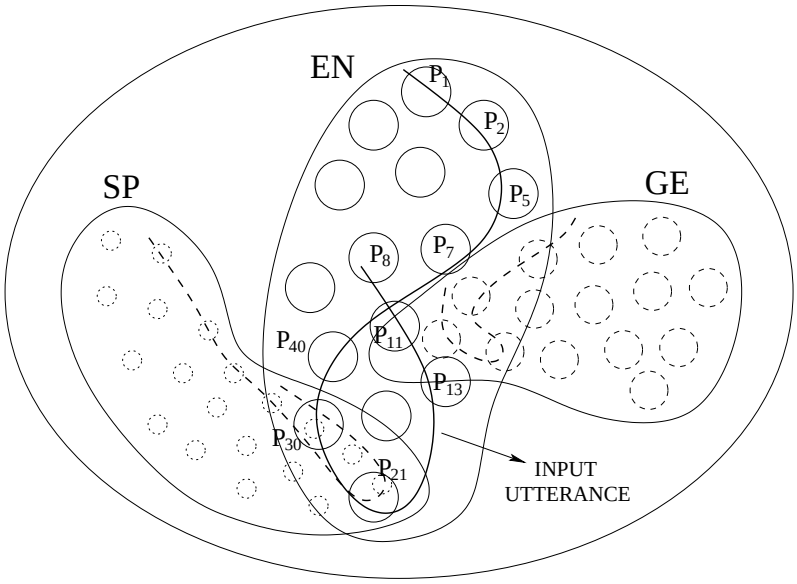



Figure 3.1: PSWR acoustic-space and sub-word decoding

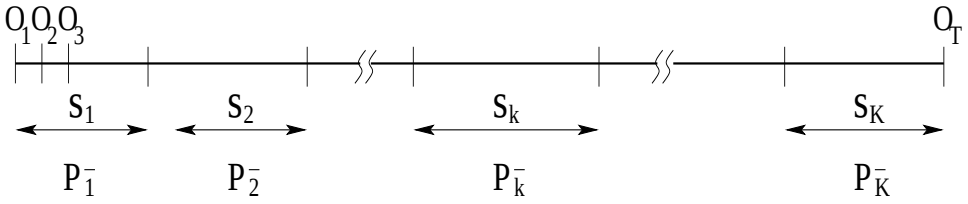



Figure 3.2: PSWR sub-word decoding

## 3.2 Parallel sub-word recognition (PSWR)

The parallel sub-word recognition (PSWR) approach can be visualized as shown in Fig. 3.1. Consider a 3-language LID task consisting of 3 languages English (EN), German (GE) and Spanish (SP). Fig. 3.1 illustrates the acoustic space (feature space) spanned by the 3 languages. Each language $\mathcal{L}_l$ is represented by a set of sub-word units $\mathcal{P}_l = \{P_1, P_2, \ldots, P_L\}$, where $L$ is the number of sub-word units in any language $\mathcal{L}_l$. Clearly, the 3 languages would have common acoustic regions which are indicated by the overlapping acoustic spaces. The overlapping acoustic regions have sub-word units which are common across the 3 languages.

Given an input utterance (as shown by the continuous dark line), PPR decodes the utterance into a sequence of sub-word units (which best match the input utterance) for each of the languages. While the input utterance is largely contained in the acoustic space of

English (clearly belonging to English), the best sequence decoding by German and Spanish are shown by the dashed lines. A typical decoding of the example shown will be

**EN:** $P_8\ P_{11}\ P_{13}\ P_{21}\ P_{30}\ P_{40}\ P_{11}\ P_7\ P_5\ P_2\ P_1$
**GE:** $P_{21}\ P_{27}\ P_{33}\ P_8$
**SP:** $P_{20}\ P_{31}\ P_{37}\ P_{32}\ P_{21}\ P_7\ P_1$

A generic decoding is as shown in Fig. 3.2, yielding an optimum segmentation and labeling of the input utterance $\mathbf{O} = (O_1, O_2, \ldots, O_T)$ into a sequence of sub-word units $(P_{\bar{1}}, P_{\bar{2}}, \ldots, P_{\bar{k}}, \ldots, P_{\bar{K}})$, where $P_{\bar{k}} \in \mathcal{P}_l$. Such a decoding enables computation of a measure of the likelihood that $\mathbf{O}$ is from each of the languages, i.e., $P(\mathbf{O}|\mathbf{EN})$, $P(\mathbf{O}|\mathbf{GE})$ and $P(\mathbf{O}|\mathbf{SP})$. By maximum a posteriori (MAP) classification, the language of the utterance is decided as

$$\arg\max(P(\mathbf{O}|\mathbf{EN}), P(\mathbf{O}|\mathbf{GE}), P(\mathbf{O}|\mathbf{SP})) \tag{3.1}$$

for equiprobable prior probabilities $P(\mathbf{EN})$, $P(\mathbf{GE})$, and $P(\mathbf{SP})$. Clearly, since the input utterance is contained well within the English acoustic-space, its likelihood $P(\mathbf{O}|\mathbf{EN})$ will be greater than $P(\mathbf{O}|\mathbf{GE})$ and $P(\mathbf{O}|\mathbf{SP})$ thus resulting in correct classification by the MAP or maximum-likelihood (ML) decision.

In this paper, we study three types of likelihood measures for $P(\mathbf{O}|\mathcal{L}_l)$ for languages $\mathcal{L}_l, l = 1, \ldots, N$ in a $N$-language LID task with a maximum-likelihood classifier which makes a classification decision given the $N$ language scores $P(\mathbf{O}|\mathcal{L}_l), l = 1, \ldots, N$ for each of the three different likelihood scores.

## 3.3 PSWR system

A typical PSWR system is shown in Fig. 3.3. The PSWR system uses a front-end sub-word recognizer (SWR) for each language in the task. For an $N$-language task, the PSWR system has $N$ front-end SWRs. Each SWR has a language dependent sub-word unit (SWU) inventory. The important point to note is that the SWU inventory is obtained without the

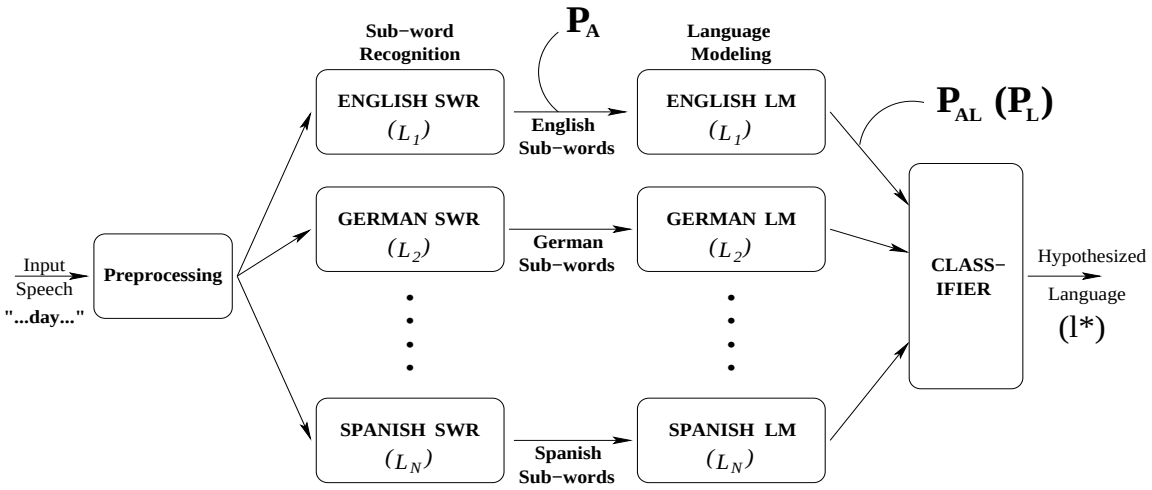



Figure 3.3: Parallel sub-word recognition (PSWR) system

need for manually labeled training data. Each front-end SWR is followed by a back-end language model (LM) which performs phonotactic analysis. The LM is typically an $n$-gram analyzer of the SWU label sequence output by the SWR front-end. For a given input utterance, the PSWR system yields $N$ scores which can be of the following types: i) Acoustic score, obtained by the front-end SWR, ii) Joint acoustic-language score, obtained by a joint decoding using both the front-end SWR and back-end LM, and iii) Language-model score, obtained from only the back-end LM. The input utterance is classified into one of $N$ languages based on a maximum likelihood decision on the $N$ scores of any of these three types.

The PSWR system thus consists of three components: i) Sub-word recognizer (SWR), ii) Language model (LM), and iii) Maximum - likelihood (ML) classifier. These are described in detail in the following sections with emphasis on the three types of scores stated above.

## 3.4 Sub-word recognizer (SWR)

### 3.4.1 Sub-word unit (SWU) inventory

Each of the $N$ SWRs in a PSWR system has an inventory of sub-word units (SWUs) $\mathcal{P}_l = \{P_1, P_2, \ldots, P_L\}$ and corresponding sub-word HMM models [21] $\mathcal{H}_l = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$

for each language $\mathcal{L}_l$, $l = 1, \ldots, N$. In the PSWR system, the training phase involves the design of the SWU inventory $\mathcal{H}_l$ for a set of SWUs $\mathcal{P}_l$ of each language $\mathcal{L}_l$ in the LID task. The procedure for generating the SWU inventory is essentially that used for acoustic SWU based speech recognition [12] and constitutes the training of the SWR in the PSWR system [8], [10]. The SWR training phase consists of the following steps:

1. Automatic segmentation of linear prediction (LP) vector sequence into acoustic segments.

2. Clustering of acoustic segments into $L$ clusters and labeling of acoustic segments into sub-word classes.

3. Generation of hidden Markov model (HMM) for each SWU using the acoustic segments in its cluster.

#### 3.4.1.1 Automatic segmentation

We use the maximum-likelihood (ML) segmentation for automatic segmentation [24], [9]. Here, the training utterances (in the form of MFCC vector sequence) is segmented into acoustic segments based on an objective criterion for a required number of segments/second.

ML segmentation is based on using the piecewise stationarity of speech as the acoustic criterion. The main criteria to be satisfied in the segmentation problem is to obtain segments which exhibit maximum acoustic homogeneity within their boundaries. The amount of acoustic inhomogeneity of a segment is measured in terms of an 'intra-segment distortion'; this is given as a sum of distances from the frames that span the segment, to the centroid of these frames comprising the segment. The general approach is to obtain a segmentation (of say, $m$ segments) with the minimum sum of intra-segment distortion over all possible segment boundaries. This segmentation problem can be solved efficiently using a dynamic programming (DP) procedure [24]. This is briefly described below.

A speech utterance is given by $\mathbf{X}_1^T = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, which is a discrete observation sequence of $T$ speech frames, where, $\mathbf{x}_n$ is a $p$-dimensional acoustic vector at frame '$n$',

denoted by $\mathbf{x}_n = [x_1(n), x_2(n), \ldots, x_p(n)]'$. A partial sequence extending from frame $i$ to frame $j$ is denoted by $\mathbf{X}_i^j = (\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_j)$.

The segmentation problem is to find '$m$' consecutive segments in the observation sequence $\mathbf{X}_1^T$. Let the segment boundaries be denoted by the set of integers $\mathcal{B} = \{b_o, b_1, \ldots, b_m\}$. The $i^{\text{th}}$ segment starts at frame $b_{i-1} + 1$ and ends at frame $b_i$; the beginning and end points of the sampled speech data are given and fixed, i.e., $b_0 = 0$, and $b_m = T$. The segmentation is thus a problem of finding segment boundaries $\{b_0, b_1, \ldots, b_m\}$ that minimize the total distortion

$$D(m, T) = \sum_{i=1}^{m} \sum_{n=b_{i-1}+1}^{b_i} d(\mathbf{x}_n, \mu_i) \tag{3.2}$$

where, $D(m, T)$ is the total distortion of a $m$ - segment segmentation of $\mathbf{X}_1^T = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$; $\mu_i$ is the generalized centroid of the $i^{\text{th}}$ segment consisting of the spectral sequence $\mathbf{X}_{b_{i-1}+1}^{b_i} = \{\mathbf{x}_{b_{i-1}+1}, \ldots, \mathbf{x}_{b_i}\}$ for a specific distance measure $d(\cdot, \cdot)$. The centroid can be viewed as a maximum-likelihood estimate of the frames in the segment $\mathbf{X}_{b_{i-1}+1}^{b_i} = \{\mathbf{x}_{b_{i-1}+1}, \ldots, \mathbf{x}_{b_i}\}$ (under the assumption that the frames in the segment are modeled by a multi-variate Gaussian, whose mean $\mu_i$ is the centroid being estimated) as,

$$\mu_i = \arg\min_{\mathbf{y}} \left[ \frac{1}{b_i - b_{i-1}} \sum_{n=b_{i-1}+1}^{b_i} d(\mathbf{x}_n, \mathbf{y}) \right] \tag{3.3}$$

For the Euclidean distance '$d$', $\mu_i$ is the average of the frames in the segment $\mathbf{X}_{b_{i-1}+1}^{b_i}$.

The segment boundaries can be solved efficiently using a dynamic programming (DP) procedure. The $i^{th}$ intra-segment distortion is given by

$$\Delta(b_{i-1} + 1, b_i) = \sum_{n=b_{i-1}+1}^{b_i} ||\mathbf{x}_n - \mu_i|| \tag{3.4}$$

Let the minimum accumulated distortion upto the $i^{th}$ segment (which ends in frame $b_i$) be denoted as $D(i, b_i)$, i.e., $D(i, b_i)$ is the minimum distortion of a segmentation of $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{b_i}\}$ into $i$ segments. The dynamic programming problem is to find the minimum of

$$D(i, b_i) = \min_{b_{i-1}}[D(i-1, b_{i-1}) + \Delta(b_{i-1}+1, b_i)] \tag{3.5}$$

for all possible $b_{i-1}$. The segmentation problem is then one of obtaining the minimum total distortion $\min\{D(m,T)\}$ (3.2). This is computed efficiently by a trellis realization and the optimal segmentation boundaries $(b_0, b_1, \ldots, b_m)$ are found by backtracking on the trellis after the optimal alignment path is determined corresponding to $\min\{D(m,T)\}$.

Though the ML segmentation was formulated for LP vectors under likelihood ratio (LR) distortion [24], it can also use other parametric representations with appropriate distortion measures. We have shown that the MFCC parameters with liftering performs best (and robustly for SNRs upto -10 dB), under Euclidean distance measure [9]. We use the same approach here.

#### 3.4.1.2 Segment clustering

ML segmentation of training data produces a large number of acoustic segments $\mathcal{S} = \{S_1, S_2, \ldots, S_M\}$ which span the speech segment space. This segment corpus is to be partitioned into $L$ clusters.

Assume that the intra-segmental spectral variation is so small that each segment is spectrally well defined by its centroid. Given the segments $\mathcal{S} = \{S_1, S_2, \ldots, S_M\}$ and the corresponding centroids $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_M\}$, design a codebook of $L$ codevectors $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_L\}$ such that the total distortion

$$D = \sum_{m=1}^{M} \min_{i=1,\ldots,L} d(\mathbf{s}_m, \mathbf{c}_i) \tag{3.6}$$

is minimized. This is a standard $K$-means [13] algorithm. The final codebook $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_L\}$ partitions the segment space $\mathcal{S}$ into $L$ clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_L\}$ such that

$$C_i = \{S_m : d(\mathbf{s}_m, \mathbf{c}_i) < d(\mathbf{s}_m, \mathbf{c}_j), \forall j = 1, \ldots, L, j \neq i\} \tag{3.7}$$

Thus each codevector $\mathbf{c}_i$ is associated with a partition $C_i$ with its corresponding cluster

of segments. The above partition implicitly labels each acoustic segment $S_m \in \mathcal{S}$ into partition $C_i$ (3.7).

#### 3.4.1.3 Sub-word HMM inventory

Each cluster $C_i \in \mathcal{C}$ defines a class of acoustically similar segments and is treated as representing a notional sub-word unit $P_i$. The acoustic segments belonging to each sub-word class $P_i, i = 1, \ldots, L$ are modeled by an HMM [21]. This results in an inventory of $L$ sub-word HMMs, $\mathcal{H}_l = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$ with the corresponding inventory of SWUs $\mathcal{P}_l = \{P_1, P_2, \ldots, P_L\}$ for the language $\mathcal{L}_l$. Each language has a different $\mathcal{P}_l$.

### 3.4.2 Sub-word tokenization

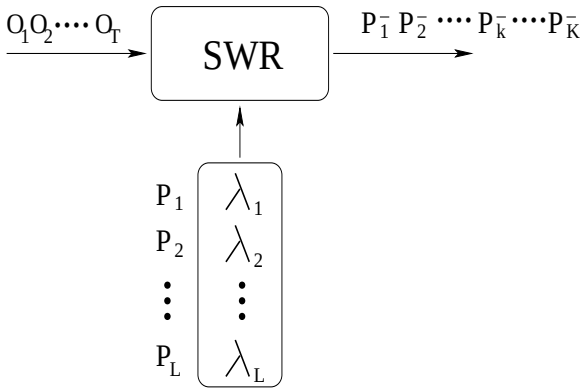



Figure 3.4: Sub-word recognition (SWR) front-end in PSWR

Fig. 3.4 shows a typical sub-word recognition (SWR) front-end in the PSWR system. The front-end SWR in path $l$ uses the sub-word inventory $\mathcal{H}_l$ to tokenize an input utterance into a sequence of sub-word unit labels by optimal decoding as in connected word recognition [21]. This decoding is an optimum 'connected sub-word recognition' problem, which generates a decoded sub-word sequence and the associated acoustic likelihood score corresponding to the decoded best path by the Viterbi search. This SWR decoding is performed for each of the $N$ languages in PSWR.

Fig. 3.5 shows the optimal sub-word decoding by the SWR. Let the input utterance be a sequence of feature vectors $\mathbf{O} = (O_1, O_2, \ldots, O_T)$. $\mathbf{O}$ is tokenized by the SWR into

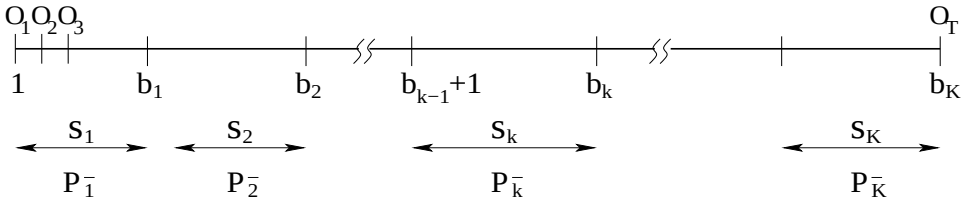



Figure 3.5: Optimal sub-word decoding

an optimal sequence of $K$ SWUs $(P_{\bar{1}}, \ldots, P_{\bar{k}}, \ldots, P_{\bar{K}})$, where $P_{\bar{k}} \in \mathcal{P}_l$. The likelihood associated with this optimal string of sub-word units is calculated as $\mathbf{P}_A(l) = P(\mathbf{O}|\mathcal{H}_l)$ (acoustic score) given by,

$$\mathbf{P}_A(l) = \max_{B,K} \sum_{k=1}^{K} \log(p(s_k|\lambda_{\bar{k}})) \tag{3.8}$$

$B = (b_0, b_1, \ldots, b_K)$, with $b_0 = 0$ and $b_K = T$, are the segment boundaries for any segmentation of the $T$ frames of $\mathbf{O}$. The $k^{th}$ segment $s_k = (O_{b_{k-1}+1}, \ldots, O_{b_k})$ is associated with the SWU model $\lambda_{\bar{k}}$, where, $\lambda_{\bar{k}}$ is the HMM model which has the maximum likelihood of generating segment $s_k$, from among $\mathcal{H}_l = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$; $p(s_k|\lambda_{\bar{k}})$ is the corresponding HMM likelihood and $P_{\bar{k}}$ is the corresponding SWU.

## 3.5 Language-model (LM)

The back-end LM in each of the PSWR system paths performs phonotactic analysis on the SWU label sequence generated by the front-end SWR; typically, it evaluates the likelihood of the SWU sequence using a bigram distribution.

Let the front-end SWR of language $\mathcal{L}_l$ tokenize the input utterance into a sequence of $K$ SWU labels $(P_{\bar{1}}, \ldots, P_{\bar{k}}, \ldots, P_{\bar{K}})$, as given by (3.8). The LM likelihood $\mathbf{P}_L(l) = P(\mathbf{O}|\mathcal{B}_l)$, using a bigram model $\mathcal{B}_l$ of language $\mathcal{L}_l$, is given by,

$$\mathbf{P}_L(l) = \sum_{k=2}^{K} \log p(P_{\bar{k}}|P_{\overline{k-1}}, \mathcal{L}_l) \tag{3.9}$$

where $P_{\bar{k}}$ and $P_{\overline{k-1}}$ are consecutive symbols observed in the tokenized SWU stream. The bigram model for language $\mathcal{L}_l$ is given by $\mathcal{B}_l = \{p_l(i,j)\} = \{p(P_j|P_i)\}, i, j = 1, \ldots, L$, where $p_l(i,j)$ is used to evaluate $p(P_{\bar{k}}|P_{\overline{k-1}}, \mathcal{L}_l)$ in (3.9) when $P_{\bar{k}} = P_j$ and $P_{\overline{k-1}} = P_i$. $\mathcal{B}_l$ is

learnt from the SWU labels obtained from tokenization of training utterances of language $\mathcal{L}_l$ using the SWR front-end of language $\mathcal{L}_l$.

## 3.6 Joint acoustic-phonotactic decoding

While sub-word recognition (SWR) and language modeling (LM) are described above as done independently, they can be combined into one step in the PSWR configuration; i.e., it is possible to integrate the acoustic/phonotactic models so that language-specific phonotactic constraints can be used during the Viterbi decoding process of SWR rather than applying the LM (phonotactic) constraints after the sub-word recognition is complete. This is referred to as *joint acoustic-phonotactic decoding* or simply as *joint decoding*. This was first used in [33] and is being extended here to joint decoding with sub-word units. The most likely sub-word sequence obtained by joint decoding and the corresponding acoustic - phonotactic likelihood measure optimally combines both the acoustic likelihood and the language model likelihood (obtained by bigram models).

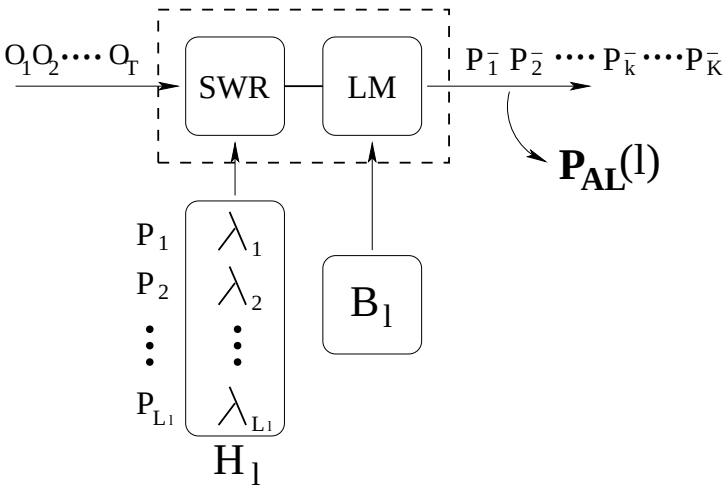



Figure 3.6: Joint acoustic-phonotactic decoding

Fig. 3.6 shows a typical joint-decoding by SWR and LM. In the case of joint acoustic-phonotactic decoding, the input utterance $\mathbf{O} = (O_1, O_2, \ldots, O_T)$ is tokenized by the SWR and LM jointly for each language $\mathcal{L}_l, l = 1, \ldots, N$. The optimal sequence of $K$ SWUs ($P_{\bar{1}}$, ..., $P_{\bar{k}}$, ..., $P_{\bar{K}}$), obtained by joint decoding by the SWR and LM of language

$\mathcal{L}_l$, maximizes the joint acoustic-phonotactic likelihood (or acoustic-language score) $\mathbf{P}_{AL}(l)$ $= P(\mathbf{O}|\mathcal{H}_l, \mathcal{B}_l)$, as given by,

$$\mathbf{P}_{AL}(l) = \max_{B,\hat{\lambda},K} \left\{ \mathbf{P}_1 + \sum_{k=2}^{K} \log[p(s_k|\lambda_{\hat{k}}) \cdot p(P_{\hat{k}}|P_{\widehat{k-1}})] \right\} \tag{3.10}$$

where $B$ and $s_k$ are as given in (3.8); $\mathbf{P}_1 = \log p(s_1|\lambda_{\hat{1}})$, $\hat{\lambda} = (\lambda_{\hat{1}}, \ldots, \lambda_{\hat{k}}, \ldots, \lambda_{\hat{K}})$ is any sequence of $\lambda_{\hat{k}} \in \mathcal{H}_l, k = 1, \ldots, K$ and $(P_{\hat{1}}, \ldots, P_{\hat{k}}, \ldots, P_{\hat{K}})$ is the corresponding sub - word sequence. $(P_{\bar{1}}, \ldots, P_{\bar{k}}, \ldots, P_{\bar{K}})$ is the SWU sequence corresponding to the optimal $\bar{\lambda} = \{\lambda_{\bar{k}}\}_{k=1}^{K}$ which maximizes $\mathbf{P}_{AL}(l)$ in (3.10). The variables in (3.10) are as shown in Fig. 3.5 for optimal sub-word decoding.

## 3.7 Language-model (LM) scores

The bigram model $\mathcal{B}_l$ used in (3.9) is estimated from SWU labels obtained from decoding of training utterances of language $\mathcal{L}_l$ by the front-end SWR of language $\mathcal{L}_l$, as given by (3.8); this does not use any LM (bigram) constraint in the SWU decoding as done in (3.10). The LM score in (3.9) using this $\mathcal{B}_l$ on the SWU sequence obtained by (3.8), is referred here as the "Language-Model score – Decoupled" and is denoted by $\mathbf{P}_{LD}(l)$.

For joint decoding (3.10), the bigram model $\mathcal{B}_l$ (which provides $p(P_{\hat{k}}|P_{\widehat{k-1}})$ in (3.10)) is estimated from the 'reference SWU labels' of training utterances, obtained as follows during the SWR training for language $\mathcal{L}_l$ (Sec. 3.4.1): The training utterances are segmented using ML-segmentation into a sequence of segments $(S_1, S_2, \ldots, S_M)$; each of these segments belongs to some cluster from $\mathcal{C} = \{C_1, C_2, \ldots, C_L\}$ and is labeled by the corresponding SWU from $\mathcal{P}_l = \{P_1, P_2, \ldots, P_L\}$. The resulting SWU label sequence gives the 'reference SWU labels' of the training utterances. Use of the $\mathcal{B}_l$ (estimated from these reference SWU labels) in (3.10) constrains the joint decoding to decode an utterance into a SWU sequence which better matches the reference SWU label sequence of the utterance. The LM score computed by (3.9) when applied on the SWU sequence obtained by (3.10) is referred here as "Language-Model score – Joint" and is denoted by $\mathbf{P}_{LJ}(l)$.

In a PPR system, the bigram model $\mathcal{B}_l$ used for joint decoding (3.10) is estimated

from the manually labeled phone sequence of the training data. In the PSWR system, the reference SWU labels, derived by automatic segmentation and labeling of training data, serves the role of the manual phone labels in the PPR system.

## 3.8 Maximum-likelihood classifier (MLC)

We have described four types of scores in PSWR for an input utterance: i) Acoustic score $\mathbf{P}_A(l)$ (Eqn. (3.8)), ii) Joint acoustic-phonotactic (or acoustic-language) score $\mathbf{P}_{AL}(l)$ (Eqn. (3.10)), iii) Language-model score – Decoupled $\mathbf{P}_{LD}(l)$ (Sec. 3.7), and iv) Language-model score – Joint $\mathbf{P}_{LJ}(l)$ (Sec. 3.7). Denoting any of these four scores by $\mathbf{P}(l)$, the maximum - likelihood (ML) classifier identifies the language of the input utterance as $\mathcal{L}_{l^*}$ which has the highest likelihood (score) $\mathbf{P}(l)$, i.e.,

$$l^* = \arg \max_{l=1,\ldots,N} \mathbf{P}(l) \tag{3.11}$$

This is illustrated in Fig. 3.7 and Fig. 3.8 for a 2-language case. Fig. 3.7 shows the PSWR generating 2 scores $\mathbf{P}(l), l = 1, 2$ for input utterance $\mathbf{O}$. Let this be a 2-dimensional vector $\mathbf{x} = (x_1, x_2)$, where $x_l = \mathbf{P}(l), l = 1, 2$. $\mathbf{x}$ is referred to as the *score-vector* henceforth. $x_1 < 0$ and $x_2 < 0$, being log-likelihoods of product of probabilities. The score vectors of training utterances of $\mathcal{L}_1$ an $\mathcal{L}_2$ along with the score-vector $\mathbf{x}$ of a test utterance are shown in Fig. 3.8. For this 2-class problem, the maximum-likelihood (ML) decision surface is given by the line $x_1 = x_2$. The MLC classifies a test utterance (with score vector $\mathbf{x}$) as $l^* = 1$ or $l^* = 2$ by the ML decision

$$l^* = \arg \max_{l=1,2} x_l \tag{3.12}$$

i.e., $\mathbf{O} \in \mathcal{L}_1$ if $x_1 > x_2$ and $\mathbf{O} \in \mathcal{L}_2$ if $x_2 > x_1$.

### 3.8.1 Bias problem in MLC

It has been observed [33] that the log-likelihood scores $\mathbf{P}_{AL}(l)$ were biased in favor of one language over another or even over all other languages. The different language models

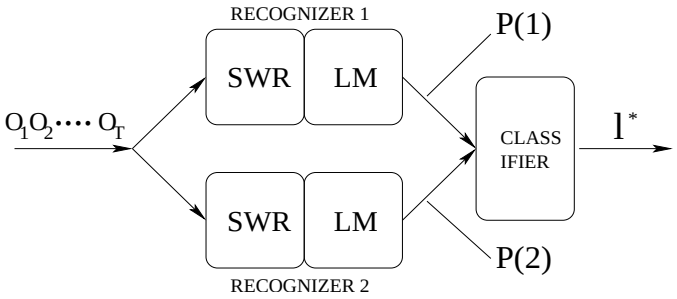



Figure 3.7: Score-vector from 2-language PSWR

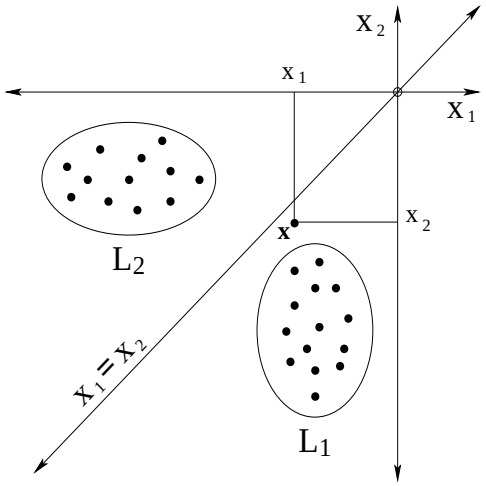



Figure 3.8: Maximum-likelihood classifier for 2-language example

(SWR and LM) are derived independently, each with its own parameters such as number of sub-word units, number of HMM states per SWU, number of mixtures per state etc.; because of these, there is a possibility that each language would have its own bias in the likelihood score that is generated.

Fig. 3.9 illustrates the bias problem geometrically for a 2-language case shown in Fig. 3.7 and Fig. 3.8. Ideally, for language $\mathcal{L}_1$, $x_1 > x_2$ and $\mathbf{x}$ is in the region marked $\mathbf{L}_1$ in box **A**. For language $\mathcal{L}_2$, $x_2 > x_1$ and $\mathbf{x}$ is in the region marked $\mathbf{L}_2$ in box **A**. The ML classifier is given by the line $x_1 = x_2$, which separates the two classes perfectly.

However, consider the case shown in box **B**. Here, the values of $\{x_1\}$ (for both $\mathcal{L}_1$ and $\mathcal{L}_2$ input) are consistently high, for the same range of values of $\{x_2\}$, i.e., the PSWR path (SWR and LM) for language $\mathcal{L}_1$ generates scores $\{x_1\}$ with a bias that increases its

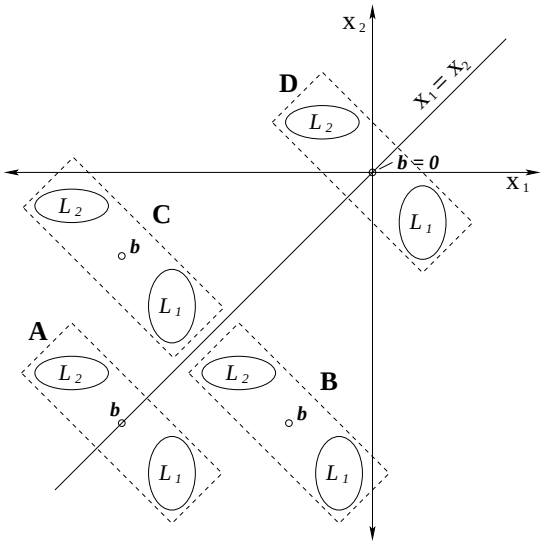



Figure 3.9: Bias and bias removal

value. This results in $x_1 > x_2$ for input from both languages $\mathcal{L}_1$ and $\mathcal{L}_2$, and all input utterances are classified as language $\mathcal{L}_1$.

A bias in the score $x_2$ which favors language $\mathcal{L}_2$ is shown in Fig. 3.9 as box **C**. Here, all input utterances generate scores $(x_1, x_2)$ where $x_2 > x_1$, resulting in a complete misclassification of utterances in language $\mathcal{L}_1$ as $\mathcal{L}_2$.

### 3.8.2 Bias-removal

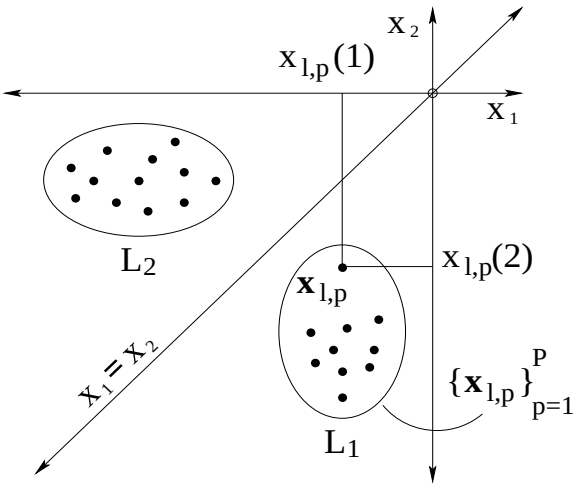



Figure 3.10: Training score-vectors for Zissman's bias-removal

For a general $N$-language LID task, PSWR produces $N$ scores $\{\mathbf{P}(n)\}, n = 1, \ldots, N$ for a given input utterance, where, $\mathbf{P}(n)$ is any of the four scores ($\mathbf{P}_A(n)$, $\mathbf{P}_{AL}(n)$), $\mathbf{P}_{LD}(n)$ and $\mathbf{P}_{LJ}(n)$ corresponding to the $n^{th}$ language path, i.e., output by the SWR, or LM, or jointly by SWR and LM, of language $\mathcal{L}_n$; any of these will henceforth be referred to as 'recognizer $n$', in general. Treating this as a vector $\mathbf{x} = [x(1), x(2), \ldots, x(N)]^t$, where $x(n) = \mathbf{P}(n)$, we shall refer to $\mathbf{x}$ as the *score-vector* of the input utterance. For a training data of $P$ utterances per language $\mathbf{U}^l = \{U_1^l, \ldots, U_P^l\}, l = 1, \ldots, N$, each language $\mathcal{L}_l$ produces $P$ score-vectors $\mathbf{X}_l = \{\mathbf{x}_{l,p}\}_{p=1}^P$, $l = 1, \ldots, N$, where, $\mathbf{x}_{l,p} = [x_{l,p}(1), \ldots, x_{l,p}(N)]^t$, $p = 1, \ldots, P$, in the $N$-dimensional score-space, i.e., $x_{l,p}(n)$ is the output score $\mathbf{P}(n)$ by the recognizer $n$ for input utterance $p$ of language $l$. Typically, $x_{l,p}(l)$ is higher than $x_{l,p}(n), n \neq l$, for perfect classification by MLC (3.11). Fig. 3.10 shows the training data score-vectors for a 2-class LID problem for languages $\mathcal{L}_1$ and $\mathcal{L}_2$.

Zissman [33] suggests the following bias removal technique (which will be referred as BR-Z):

1. Find the language-dependent bias $b(n)$ as the average of the log-likelihoods for all input utterances (from all languages) processed by recognizer $n$.

$$b(n) = \frac{1}{N} \sum_{l=1}^{N} \frac{1}{P} \sum_{p=1}^{P} x_{l,p}(n) \tag{3.13}$$

   The corresponding bias vector is $\mathbf{b} = [b(1), b(2), \ldots, b(N)]^t$ [1]. This is the centroid of the $N$ clusters of score-vectors where cluster $l$ is the collection of score-vectors $\{\mathbf{x}_{l,p}\}_{p=1}^P \in \mathcal{L}_l$.

2. Given a test utterance $V$, let its score vector be $\mathbf{y} = [y(1), \ldots, y(N)]^t$. Perform bias-removal on this score vector as $\mathbf{y}' = \mathbf{y} - \mathbf{b}$.

3. Perform ML classification on $\mathbf{y}' = [y'(1), \ldots, y'(N)]^t$ to hypothesize the correct language of $V$ as $\mathcal{L}_{l^*}$ given by,

[1] It should be noted that $P$ should be constant across different languages; otherwise, the bias estimate of (3.13) would itself be biased. It is also assumed that the utterance likelihoods are normalized by the length of the utterance (for acoustic and joint acoustic-language scores) and by the length of the decoded sub-word sequence (for the language model scores).

$$l^* = \arg \max_{n=1,\ldots,N} y'(n) \tag{3.14}$$

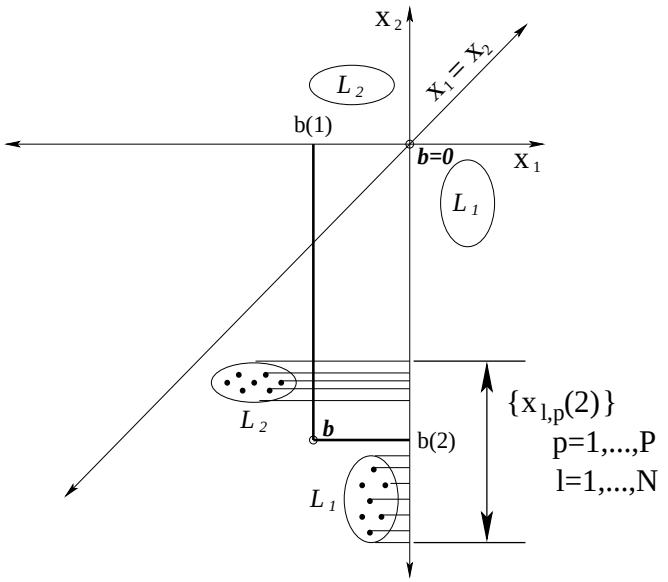



Figure 3.11: Bias-vector in Zissman's bias-removal

Fig. 3.11 illustrates the computation of the bias by Zissman's method for a 2-class problem. It shows the computation of the bias-vector **b** from the training data score-vectors of all languages. For example, the bias component $b(2)$ is the average of score-vector components $\{x_{l,p}(2)\}$ over $l = 1, 2$ and $p = 1, \ldots, P$.

In the example shown in Fig. 3.9, for the 2-language case, BR-Z (when applied on score-vectors of the training utterances) corresponds to first finding the centroid **b** for the biased case shown by box **B** (or box **C**), and translating the score vectors such that the new centroid is at the origin. This is shown as box **D**. Bias removal on test data translates it to box **D**, where the ML classifier can perform optimally in discriminating the 2-classes ($N$-classes in general). Note that bias-removal (BR-Z) makes use of the training data for estimating the bias **b** (the centroid of score-vectors from all language input) which is removed from score-vectors of test utterances before performing ML classification (3.14).

We have observed this bias problem in all four scores $\mathbf{P}_A(l)$, $\mathbf{P}_{AL}(l)$, $\mathbf{P}_{LD}(l)$ and $\mathbf{P}_{LJ}(l)$ to varying extents and have used the bias-removal method of [33] for all the four

scores in the PSWR system. Note that the PPR system also has the same four types of scores as discussed here for PSWR and in [27], though not dealt with in [33]. An alternate method for bias-removal has been proposed in [27].

## 3.9 Experiments and Results

We treat the PPR system [33], [27], [7] as a baseline system with which to compare the performance of the PSWR system. PPR is described briefly in Sec. 3.1. We present here results comparing the performance of the PPR and PSWR systems for the four different types of scores: Acoustic score ($\mathbf{P}_A$), Joint acoustic-language score ($\mathbf{P}_{AL}$), Language model score – Decoupled ($\mathbf{P}_{LD}$), and Language model score – Joint ($\mathbf{P}_{LJ}$).

### 3.9.1 Database

We use the Oregon Graduate Institute Multi-language Telephone Speech (OGI-TS) corpus [17] for evaluation. The OGI-TS has a total of 11 languages, out of which 6 languages – English (EN), German (GE), Hindi (HI), Japanese (JA), Mandarin (MA) and Spanish (SP) – have phonetic labels. We evaluate both the PPR and PSWR systems using these 6 languages of the OGI-TS database.

Both the PPR and PSWR systems are trained on the 50 story-bt (story-before-the-tone) utterances per language spoken by 50 different speakers. The PPR and PSWR systems are tested using 20 story-bt utterances per language outside the training data; these are spoken by 20 more different speakers. The database is also used to produce two sets of data [33]: i) '45 sec data' comprising 45 seconds of data in the story-bt utterance and, ii) '10 sec data' obtained as a set of 10 second cuts from the 45 sec utterances.

### 3.9.2 Preprocessing

Speech data is parameterized every 20ms with a frame shift of 20ms. Each frame of speech is first pre-emphasized by $(1 - 0.95z^{-1})$ and then windowed by a Hamming window. The pre-emphasized and windowed frame is then used for parameter estimation. A 12-dimensional MFCC parameter with lifter is found to perform optimally for ML segmentation (Sec.

3.4.1.1) [24], [9].

### 3.9.3 Parameters of PSWR system

In ML segmentation (Sec. 3.4.1.1), the segmentation performed using dynamic programming can segment the input speech into a pre-specified number of segments [24], [9]. We specify this as $m = Rt$, where $R$ is the number of segments per second and $t$ is the duration of the input utterance in seconds. $R$ is used as a parameter to control the segment rate of the ML segmentation. $R$ takes values as 2, 5, 10 and 20 seg/sec. $R = 2$ and 5 correspond to a coarse segmentation and can generate broad-phonetic segments and phone strings. HMM captures the dynamics within such phone strings. This could offer a promising possibility that the sub-word HMM inventory has approximated the phone strings characterizing a particular language. $R = 10$ gives phone-like segmentation as the phone rate in normal speech is about 10 phones/sec; $R = 20$ results in a fine segmentation and produces sub-phonemic segments.

The number of clusters $L$ in the segment clustering step (Sec. 3.4.1.2) determines the sub-word inventory size. We use $L$ as a parameter to control the resolution of the acoustic space; $L$ is varied as 10, 30, 50 and 100. Small $L$ such as 10, corresponds to a coarse clustering and generates HMM models which typically model broad-phonetic categories. Values of $L = 30$ and 50 generate phone-like units in the inventory; their clusters match the phonemic units as languages typically have phone set sizes in this range. $L = 100$ yields a finer clustering of the acoustic space. Note that smaller $R$ results in larger segments which requires a larger $L$ to capture the full variability of the acoustic space.

### 3.9.4 HMM parameters

Training of a PPR system consists of generating an inventory of phone HMM models of size determined by the number of different phonetic units in the corpus labeling scheme. Whereas in a PSWR system, the training consists of generating an inventory of sub-word HMM models of any desired size $L$ (Sec. 3.4.1.2 and 3.4.1.3) Both these training procedures are implemented using HTK.

ML segmentation (Sec. 3.4.1.1) and segment clustering (Sec. 3.4.1.2) are done using 12-dimensional MFCC parameters with a lifter [9]. Subsequent to this, HMM models (Sec. 3.4.1.3) are computed using a 26-dimensional parameter vector consisting of 12 MFCC, 12 delta-MFCC, energy and delta energy.

The HMMs for both the PPR and PSWR systems are 3 state, left to right models. For PPR system we use 6 Gaussian mixtures per state. For the PSWR system, different number of mixtures per state (1, 3, 6, 9 and 12) are used, for each $(R, L)$ from a range of $R = 2$, 5, 10, 20 seg/sec and $L = 10$, 30, 50 and 100. Only diagonal covariances are used for all mixture components.

## 3.9.5 Results

### 3.9.5.1 Acoustic score performance

First we show results of PSWR using only the acoustic score $\mathbf{P}_A$ in Table I and Table II. Table I shows the LID accuracy of PPR and PSWR systems for a 6 language task (EN, GE, HI, JA, MA, SP), for training (50 utterances/language) and test (20 utterances/language) data. Performance on 45 sec data and 10 sec data are shown in Table I(a) and I(b) respectively. The PSWR system parameters in this experiment are as follows: The HMMs of each SWU uses 9 Gaussian mixtures per state. Segmentation rate $R = 10$ seg/sec and sub-word inventory size $L = 30$. These parameters correspond to phone-like segmentation and an inventory size closely approximating a typical phone-set of a language.

We can observe the following from these results:

1. Both PSWR and PPR systems have a high LID accuracy (86% – 92%) on training data for both 45 sec and 10 sec data. The training data performance for 10 sec data is about 1.5% less than for 45 sec data. The PSWR system performs better than the PPR system by about 4% on training data on both 45 sec and 10 sec data.

2. On test data (45 sec), the PSWR system has an LID accuracy of 67.5% in comparison to the PPR system with 70%. On 10 sec data, the corresponding LID accuracies differ by 1% (63.3% for PPR and 62.3% for PSWR). Note that PSWR is offering

**Table I**
LID accuracy of PPR and PSWR systems (6 languages)
Training: 50 utterances / language; Test: 20 utterances / language

**Table I (a)** – Utterance length: 45 sec

| Language | PPR | | PSWR | |
|---|---|---|---|---|
| | Training | Test | Training | Test |
| EN | 95.9 | 100 | 83.7 | 95.0 |
| GE | 92.0 | 65.0 | 88.0 | 60.0 |
| HI | 90.0 | 100 | 88.0 | 95.0 |
| JA | 68.0 | 20.0 | 92.0 | 35.0 |
| MA | 94.0 | 100 | 100 | 90.0 |
| SP | 88.0 | 35.0 | 98.0 | 30.0 |
| **Average** | **87.9** | **70.0** | **91.6** | **67.5** |

**Table I (b)** – Utterance length: 10 sec

| Language | PPR | | PSWR | |
|---|---|---|---|---|
| | Training | Test | Training | Test |
| EN | 96.0 | 93.8 | 84.0 | 90.0 |
| GE | 88.4 | 59.5 | 87.5 | 55.7 |
| HI | 89.9 | 98.8 | 87.5 | 88.8 |
| JA | 66.0 | 13.9 | 89.3 | 29.1 |
| MA | 92.7 | 82.1 | 97.8 | 79.5 |
| SP | 84.9 | 30.3 | 95.5 | 28.9 |
| **Average** | **86.3** | **63.3** | **90.2** | **62.3** |

a comparable performance to PPR with the advantage that the PSWR inventory is obtained from automatic procedures without manually labeled data.

3. The consistently high performance of PSWR system on both the 10 sec data and the 45 sec data shows the reliability of the acoustic score alone in ML based LID as given by (1) and (2); adding an LM model will improve this performance even further.

4. The high performance of PSWR on training data shows its potential to offer excellent LID accuracies in general, even better than a PPR system.

5. While the 3 languages English, Hindi and Mandarin perform consistently well, the

other 3 languages, German, Japanese and Spanish, perform poorly (for both PPR and PSWR systems) thereby lowering the average performance of LID accuracy.

**Table II**
Average LID accuracy of PSWR system on 6 language task
Test data: 20 story-bt sentences; Utterance length: 45 sec
ML Segmentation rate $R$: 2, 5, 10, 20 seg/sec
Sub-word inventory size $L$: 10, 30, 50, 100

| | Mixtures/state | | | | |
|---|---|---|---|---|---|
| $(R, L)$ | 1 | 3 | 6 | 9 | 12 |
| **(2,10)** | 40.8 | 53.3 | 59.2 | **66.7** | 59.2 |
| (2,30) | 50.0 | 60.0 | 64.2 | 62.5 | 56.9 |
| (2,50) | 55.0 | 61.7 | 63.3 | 61.7 | 57.5 |
| (2,100) | 55.8 | 62.5 | 64.2 | 60.0 | 59.2 |
| **(5,10)** | 42.5 | 55.0 | 60.0 | 63.3 | **65.0** |
| (5,30) | 52.5 | 60.8 | 59.2 | 60.0 | 64.2 |
| (5,50) | 55.8 | 64.2 | 64.2 | 60.0 | 64.2 |
| (5,100) | 59.2 | 60.8 | 61.7 | 62.5 | 64.2 |
| (10,10) | 41.7 | 55.0 | 58.3 | 59.2 | 58.3 |
| **(10,30)** | 56.7 | 65.0 | 62.5 | **67.5** | 62.5 |
| (10,50) | 61.7 | 60.8 | 60.0 | 64.2 | 65.8 |
| (10,100) | 59.2 | 64.2 | 63.3 | 62.5 | 62.5 |
| (20,10) | 50.8 | 60.0 | 65.0 | 63.3 | 62.5 |
| (20,30) | 52.5 | 60.8 | 60.6 | 59.2 | 59.2 |
| **(20,50)** | 55.0 | 63.3 | **69.2** | 64.2 | 63.3 |
| (20,100) | 53.3 | 63.3 | 61.7 | 62.5 | 62.5 |

Table II shows the average LID accuracy of PSWR (on 6 languages) for ML segmentation rates, $R = 2, 5, 10$ and 20 seg/sec, and sub-word inventory sizes, $L = 10$, 30, 50 and 100, for 45 sec data on test data of 20 story-bt utterances. For each $(R, L)$, the number of mixtures is varied as 1, 3, 6, 9 and 12. This accounts for the different intra-cluster variability arising from various resolutions of the segment clusters corresponding to various combinations of $R$ and $L$ as described in Sec 4.4.

The following can be noted from Table II:

1. A phone-like segmentation of $R = 10$ seg/sec (with $L = 30$) offers a very good performance of 67.5% (with 9 mixtures/state),

2. Sub-phonemic PSWR with $R = 20$ seg/sec and $L = 50$ (typical size of a phone set in a language) offers the best performance of 69.2% comparable to a PPR system performance of 70.0%.

3. A coarse segmentation followed by coarse clustering ($R = 2, L = 10$) has the best performance of 66.7% (with 9 mixtures/state). Note that, as described in Sec 4.4, this resolution corresponds to a phone-string decoding and can be considered to offer potentially good performance, close to that of phone-like and sub-phonemic decoding.

4. For small $L$, the intra-cluster variability per HMM model is relatively higher (than for large $L$) thereby requiring more number of mixtures per state (6-12) in the HMM model to offer high LID accuracy.

5. The performance difference for 1 mixture/state and the best performances for 6 or 9 mixtures/state can be as high as 26% which shows the importance of choosing the right number of mixtures/state for any typical $(R, L)$ used in the PSWR system.

Encouraged by the above results, we extended our experiment to only 5 sec utterances also. The average LID accuracy for PPR (on 6 languages), on test data of 45 sec, 10 sec and 5 sec data are respectively 70.0%, 63.3% and 59.5%. The corresponding LID accuracies for PSWR system with $R = 10$ and $L = 30$ are 67.5%, 62.3% and 62.2%. Note that while the PPR system performance for 5 sec (59.5%) is 3.8% lower than for 10 sec (63.3%), the PSWR system gives a performance of 62.2% which is only 0.1% less than the 10 sec performance. This indicates that PSWR can also be exploited for more robust performance based on short utterances.

#### 3.9.5.2 Performance of all scores

Fig. 3.12 shows the average LID accuracy for the 6 languages (EN, GE, HI, JA, MA, SP), for both PPR and PSWR systems for the 4 different scores:

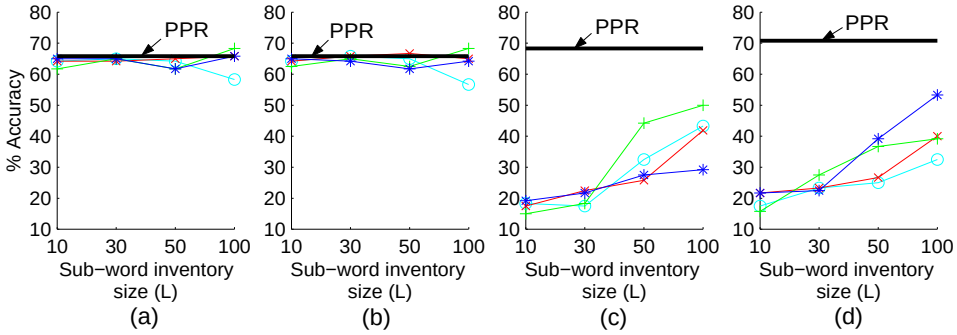



Figure 3.12: Average LID accuracy of PPR and PSWR for different scores: (a) Acoustic score ($\mathbf{P}_A$), (b) Joint acoustic-language score ($\mathbf{P}_{AL}$), (c) Language-model score – Decoupled ($\mathbf{P}_{LD}$) and (d) Language-model score – Joint ($\mathbf{P}_{LJ}$). Test data: 20 utterances / language. Utterance length: 45 sec. Thick dark line: PPR. Legend for PSWR: R=2 (O), R=5 (X), R=10 (+), R=20 (*).

(a) Acoustic score ($\mathbf{P}_A$),

(b) Joint acoustic-language score ($\mathbf{P}_{AL}$),

(c) Language-model score – Decoupled ($\mathbf{P}_{LD}$), and

(d) Language-model score – Joint ($\mathbf{P}_{LJ}$).

These results are for test data of 20 utterances / language each 45 sec long (story-bt utterances). For PSWR, there are two parameters for each of the four scores:

1. Segmentation rate $R$, varying as $R = 2, 5, 10$ and 20 segments/sec, and

2. SWU inventory size $L$, varying as $L = 10, 30, 50$ and 100.

Each plot ((a) to (d)) shows the LID accuracy ($y$-axis) vs $L = 10, 30, 50, 100$ ($x$-axis) for different $R = 2, 5, 10, 20$, as given in the legend in Fig. 3.12. The PPR system does not have any parameters and the LID accuracy is shown as a thick-dark line for each of the four scores. For PSWR, the HMMs of each SWU uses 9 Gaussian mixtures per state, for the various $(R, L)$ studied here.

The following can be observed from this figure:

1. PPR performs equally well (with an LID accuracy of 70%) for all the four scores with slight advantage for $\mathbf{P}_{LD}$ or $\mathbf{P}_{LJ}$.

2. PSWR offers a performance comparable to PPR for both $\mathbf{P}_A$ and $\mathbf{P}_{AL}$ for all the SWU inventory sizes $L$ and segmentation rates $R$.

3. The advantage of PSWR is the control on the granularity of the acoustic space in terms of variable size of SWUs obtained for different $R$ and $L$. We had expected this to provide a means of determining whether any particular granularity is optimal for discrimination of languages. However, while $\mathbf{P}_A$ and $\mathbf{P}_{AL}$ showed no particular dependence on this granularity, $\mathbf{P}_{LD}$ (or $\mathbf{P}_{LJ}$) seems to require larger SWU inventory size (i.e., fine acoustic resolution) for higher PSWR performance.

4. Given that $\mathbf{P}_{AL}$ integrates both the acoustic and phonotactic information, and represents the complete PPR (and PSWR) system, PSWR performs as well as PPR, making it an efficient alternative to PPR, with the advantage of not requiring manually labeled training data for any of the languages in the task.

5. The acoustic score $\mathbf{P}_A$ alone is also seen to be consistently high for both PPR and PSWR, indicating the possibility of better LID performance with improved acoustic modeling.

6. With regard to LM scores $\mathbf{P}_{LD}$ and $\mathbf{P}_{LJ}$, PSWR is distinctly poorer than PPR. It appears that the SWU inventory is not sufficiently unique across languages. This can also be ascribed to accumulation of errors from the sub-word tokenization stage.

# Chapter 4

# Sub-word recognition and language modeling (SWRLM)

## 4.1 Introduction

SWRLM uses a single front-end sub-word recognizer (SWR) followed by $N$ back-end LMs for an $N$ language LID task. The front-end SWR can be language dependent or language independent. In PSWR, we need sub-word units (SWUs) for all the languages in the task. Where as, since SWRLM uses single front-end, it is enough to have SWUs in only that language which is used for SWR. This sub-word recognizer need not be one among the languages that are in the task but in PSWR the front-ends must be the languages in the LID task.

The SWRLM performance will improve if we use language independent sub-word unit inventory [4], which is obtained from the all languages in the LID task. So, it may span the entire acoustic space, where as language dependent SWR may not span the full acoustic space. When we run multiple SWRLMs in parallel it may span the entire acoustic space. So, it will improve the system performance.

The SWRLM uses a single front-end SWR. During training the sub-word unit inventory is obtained by automatic segmentation, segment clustering followed by segment HMM modeling (Sec. 3.4) [24], [9]. The training utterances is segmented into acoustic segments based on objective criterion for a required number of segments/second, which is a controlling parameter for time resolution. ML segmentation of training data produces a large number of acoustic segments which span the speech segment space. This segment

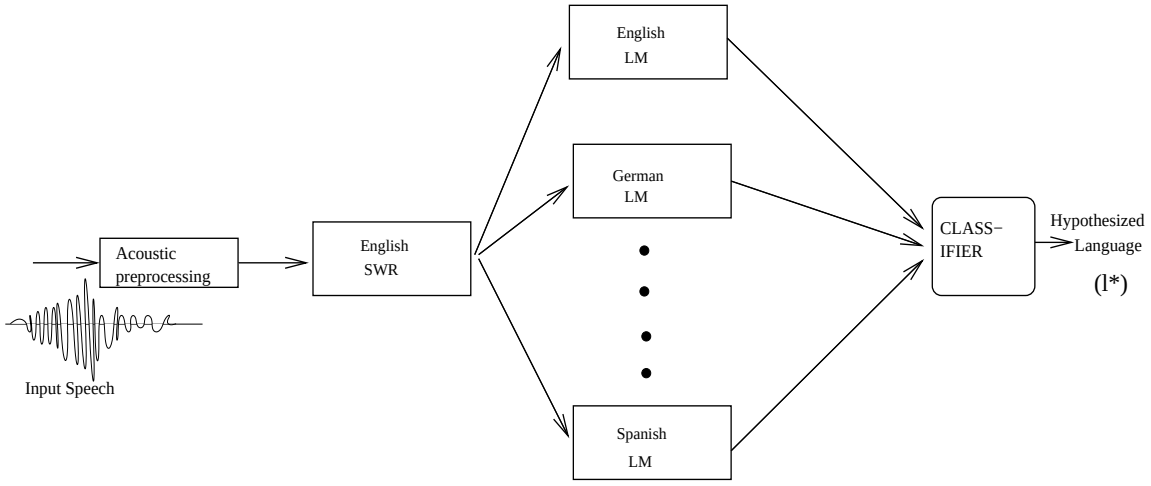



Figure 4.1: Sub-word recognition followed by language modeling (SWRLM) system

corpus is to be partitioned into $L$ clusters using $K$-means segment clustering in which $L$ is a controlling parameter. The SWRLM is studied with various number of segments and clusters and also with two classifiers, viz., Maximum likelihood classifier (MLC) and Gaussian classifier.

## 4.2 SWRLM

Figure 4.1 shows the structure of typical SWRLM. SWRLM uses a single front-end SWR. This single language SWR can be a language dependent SWR or a language independent SWR. We are using language dependent SWR for an $N$ language LID task of languages $\mathcal{L}_1, \ldots, \mathcal{L}_N$. The single language, language dependent front-end can be one of $\mathcal{L}_1, \ldots, \mathcal{L}_N$ or even language outside this set. In this work, the front-end SWR is typically one of $\mathcal{L}_1, \ldots, \mathcal{L}_N$ is represented as $\mathcal{L}_l$.

In SWRLM the front end will have multiple back-ends. The back-end LMs in the SWRLM system performs phonotactic analysis; typically, it operates on the decoded SWU label sequence generated by the front-end sub-word tokenization and evaluates the LM likelihood of the sub-word sequence using the bigram distribution.

### 4.2.1 Sub-word recognizer (SWR)

The SWR in the SWRLM has an inventory of sub-word units (SWUs) $\mathcal{P}_l = \{P_1, P_2, \ldots, P_{L_l}\}$ and corresponding sub-word HMM models $\mathcal{H}_l = \{\lambda_1, \lambda_2, \ldots, \lambda_{L_l}\}$ for the front-end of the language $l$. The SWR front-end of language $\mathcal{L}_l$ has a SWU inventory $\mathcal{H}_l$ for a set of $\mathcal{P}_l$.

The sub-word recognizer in SWRLM uses sub-word inventory $\mathcal{H}_l$ to tokenize an input utterance into a sequence of sub-word unit labels by optimal decoding as in 'connected word recognition' problem, which generates a decoded sub-word label sequence and the associated acoustic likelihood score corresponding to the decoded best path by Viterbi search.

Let the input utterance be a sequence of feature vector $\mathbf{O} = \{O_1, O_2, \ldots, O_T\}$. $\mathbf{O}$ is tokenized by the front-end SWR in to optimal sequence of $K$ phones $\{P_{\overline{1}}, P_{\overline{2}}, \ldots, P_{\overline{k}}, \ldots, P_{\overline{K}}\}$, where $P_{\overline{k}} \in \mathcal{P}_l$. The likelihood associated with this optimal string of SWUs is calculated as $\mathbf{P}_A(l) = P(\mathbf{O}|\mathcal{H}_l)$ (acoustic score), which is given by

$$\mathbf{P}_A(l) = \max_B \sum_{k=1}^{K} \log(p(s_k|\lambda_{\overline{k}})) \tag{4.1}$$

$B = \{b_0, b_1, \ldots, b_K\}$, with $b_0 = 0$ and $b_K = T$, are the segment boundaries for any segmentation of the $T$ frames of $\mathbf{O}$. The $k^{th}$ segment $s_k = \{O_{b_{k-1}+1}, \ldots, O_{b_k}\}$ is associated with the sub-word model $\lambda_{\overline{k}}$ (and the corresponding sub-word $P_{\overline{k}}$) given by

$$\lambda_{\overline{k}} = \arg\max_{\lambda_i \in \mathcal{H}_l} p(s_k|\lambda_i) \tag{4.2}$$

i.e., $\lambda_{\overline{k}}$ is the HMM model which has the maximum likelihood of generating segment $s_k$, from among $\mathcal{H}_l = \{\lambda_1, \lambda_2, \ldots, \lambda_{L_l}\}$; $p(s_k|\lambda_{\overline{k}})$ is the corresponding HMM likelihood and $P_{\overline{k}}$ is the corresponding SWU.

### 4.2.2 Language-model (LM)

In SWRLM the front end will have multiple back-ends for an $N$ language LID task for languages $\mathcal{L}_1, \ldots, \mathcal{L}_N$. SWRLM has $N$ back-end LMs one for each language. The single language front-end SWR feeds the SWU labels sequence of a decoded input utterance to each of $N$ back-end LMs. The back-end LMs in the SWRLM system performs phonotactic

analysis; typically, it operates on the decoded SWU label sequence generated by the front-end sub-word tokenization and evaluates the LM likelihood of the sub-word sequence using the bigram distribution of language Ł$_l$.

Let the front-end SWR tokenize the input utterance into a sequence of $K$ SWU labels $\{P_{\overline{1}}, P_{\overline{2}}, \ldots, P_{\overline{k}}, \ldots, P_{\overline{K}}\}$, as given by (4.1), (4.2). The language likelihood $\mathbf{P}_L(l) = P(\mathbf{O}|\mathcal{B}_l)$ using a bigram model $\mathcal{B}_l$ of language $\mathcal{L}_l$ is then given by,

$$\mathbf{P}_L(l) = \log(p(P_{\overline{1}}|\mathcal{L}_l) + \sum_{k=2}^{K} \log p(P_{\overline{k}}|P_{\overline{k-1}}, \mathcal{L}_l) \tag{4.3}$$

where $P_{\overline{k}}$ and $P_{\overline{k-1}}$ are consecutive symbols observed in the tokenized or decoded SWU stream.

## 4.3 Classifiers

### 4.3.1 Maximum-likelihood classifier

In SWRLM, the maximum-likelihood classifier decision is used, which hypothesizes that $l^*$ is the language of the unknown utterance (Sec. 3.8), where $l^*$ is given by

$$l^* = \arg \max_{l=1,\ldots,N} \mathbf{P}_L(l) \tag{4.4}$$

### 4.3.2 Gaussian classifier (GC)

For a general $N$-language LID task, SWRLM produces $N$ scores $\{\mathbf{P}(n)\}, n = 1, \ldots, N$ for a given input utterance, where, $\mathbf{P}(n)$ is $\mathbf{P}_L(n)$ corresponding to the $n^{th}$ language path, i.e., output by the SWR and LM of language $\mathcal{L}_n$; any of these will henceforth be referred to as 'recognizer $n$', in general. Treating this as a vector $\mathbf{x} = [x(1), x(2), \ldots, x(N)]^t$, where $x(n) = \mathbf{P}(n)$ for for a $N$-language task for any input utterance. The $N$-languages $\mathcal{L}_1, \ldots, \mathcal{L}_N$, correspond to $N$ clusters in $N$-dimensional space, each cluster $l$ with $P$ training score-vectors $\mathbf{X}_l = \{\mathbf{x}_{l,p}\}_{p=1}^{P}$, where $\mathbf{x}_{l,p} = [x_{l,p}(1), \ldots, x_{l,p}(N)]^t$, $p = 1, \ldots, P$, in the $N$-dimensional score-space. With this, the LID problem can be treated as a conventional $N$-class classification problem in the score-space of $N$-dimensions given the $N$ likely clusters

$\mathbf{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$ of $P$ training vectors each. These $NP$ score-vectors form the supervised training data. Given a test utterance, and its score-vector $\mathbf{y} = [y(1), \ldots, y(N)]^t$, the classification problem is to classify $\mathbf{y}$ into one of $N$ classes $\{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$ given the supervised training data [2]. This is illustrated in Fig. 4.2 for a 2-class LID problem.

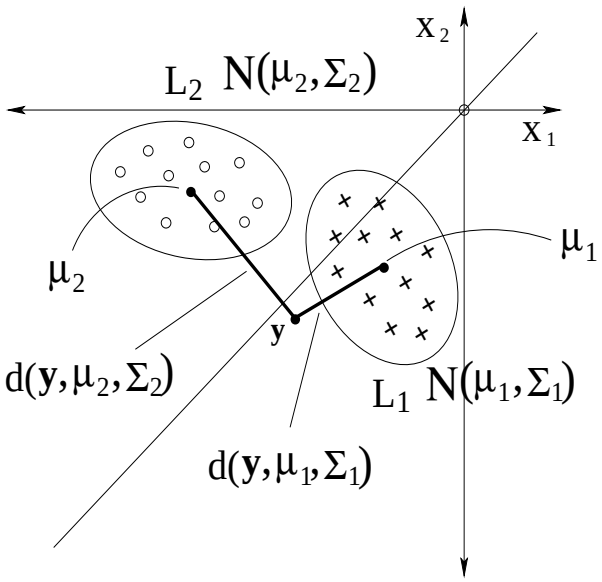



Figure 4.2: Gaussian classifier

We use a multi-variate Gaussian classifier (GC) for solving this [2]. Thus, each class is represented by a Gaussian density $\mathcal{N}(\mu_l, \Sigma_l)$, $\mu_l$ and $\Sigma_l$ being the mean and covariance of class $\mathcal{L}_l$. These are the parameters of the class-conditional densities and are estimated from the training data of $P$ vectors $\{\mathbf{x}_{l,p}\}_{p=1}^{P}$ of class $\mathcal{L}_l$ as,

$$\mu_l = \frac{1}{P} \sum_{p=1}^{P} \mathbf{x}_{l,p} \tag{4.5}$$

$$\Sigma_l = \frac{1}{P} \sum_{p=1}^{P} (\mathbf{x}_{l,p} - \mu_l)(\mathbf{x}_{l,p} - \mu_l)^t \tag{4.6}$$

A test utterance $V$ is classified as language $\mathcal{L}_{l^*}$ based on its score-vector $\mathbf{y} = [y(1), \ldots, y(N)]^t$, if,

$$d(\mathbf{y}, \mu_{l^*}, \Sigma_{l^*}) \leq d(\mathbf{y}, \mu_l, \Sigma_l), l = 1, \ldots, N \tag{4.7}$$

where,

$$d(\mathbf{y}, \mu_l, \Sigma_l) = (\mathbf{y} - \mu_l)^t \Sigma_l^{-1} (\mathbf{y} - \mu_l) \tag{4.8}$$

is the distance measure equivalent of the class conditional probability $p(\mathbf{y} \mid \mathcal{N}(\mu_l, \Sigma_l))$. Fig. 4.2 shows the Gaussian parameters and the distances to score-vector $\mathbf{y}$ from the means of the 2 classes. The distance measure gets simplified to three types: i) Euclidean distance (ED) for $\Sigma_l = I$, the identity matrix, i.e., the covariance information is not used at all, ii) Weighted Euclidean distance (WED) if $\Sigma_l$ is diagonal, i.e., non-diagonal elements are 0, and iii) Mahalanobis distance (MD) when $\Sigma_l$ is full covariance.

The advantage of Gaussian classifier is that it does not suffer from the bias problem as it does not use an absolute discriminant function such as the ML classifier which is fixed relative to the location of the classes. GC will perform well since the parameters of the Gaussian density are estimated on the training vectors, thereby generating the optimal maximum a posteriori (MAP) decision surfaces relative to the location of the clusters for minimum classification error. The Gaussian classifier has one limitation that it is a further parameterization of the score-space distribution. It may be that such a parameterization is insufficient (for example, multi-mixture Gaussian may be required for certain classes) and hence may result in a poorer performance.

## 4.4 Experiments and results

We present here the results of the performance of the SWRLM for all language front-ends with different segment rates ($R$ = 2, 5, 10 and 20) and sub-word unit inventory sizes ($L$ = 10, 30 and 50). We also present results of two different classifiers, namely Maximum likelihood classifier (MLC) and Gaussian classifier (GC).

### 4.4.1 Database

We study the performance of SWRLM system using OGI-TS database [17] for experiments. We evaluated the SWRLM system using the six languages - English (EN), German (GE),

Hindi (HI), Japanese (JA), Mandarin (MA) and Spanish (SP).

SWRLM system is trained on 622 'story-bt' (story-before-the-tone) utterances from all six languages spoken by 622 different speakers. The system is tested using 20 'story-bt' utterances per language outside from the training data; the training and test utterances are each 45 seconds long.

### 4.4.2 Parameters of SWRLM system

The ML segmentation technique (Sec. 3.4.1.1), can segment the input speech into a pre-specified number of segments [24], [9]. If $m$ is number of segments, we can specify this as $m = Rt$, where $R$ is the number of segments per second and $t$ is the duration of the input utterance in seconds. $R$ is used as a parameter to control the segment rate of the ML segmentation. $R$ takes values as 2, 5, 10 and 20 seg/sec. $R = 2$ and 5 correspond to a coarse segmentation and can generate broad-phonetic segments and phone strings. $R = 10$ gives phone-like segmentation as the phone rate in normal speech is about 10 phones/sec; $R = 20$ results in a fine segmentation and produces sub-phonetic segments.

The sub-word inventory size $L$, controls the resolution of the acoustic space. $L$ is varied as 10, 30 and 50. Small $L$ such as 10, corresponds to a coarse clustering and generates HMM models which typically model broad-phonetic categories. Values of $L = 30$ and 50 generate phone-like units in the inventory; their clusters match the phonetic units as languages typically have phone set sizes in this range. Note that smaller $R$ results in larger segments which requires a longer $L$ to capture the full variability of the acoustic space.

### 4.4.3 Model building

Speech data is parameterized every 20ms with a frame shift of 20ms. Each frame of speech is first pre-emphasized by $(1 - 0.95z^{-1})$ and then windowed by a Hamming window. The pre-emphasized and windowed frame is then used for parameter estimation.

The training of a SWRLM system consists of generating an inventory of sub-word HMM models of any desired size $L$. The training procedure is implemented using HTK

[28].

ML segmentation and segment clustering are done using 12-dimensional MFCC parameters with a lifter [9]. Subsequent to this, HMM models are computed using a 26-dimensional parameter vector consisting of 12 MFCC, 12 delta-MFCC, energy and delta energy.

We used 3 state, left to right HMM models and 6 Gaussian mixtures per state for each $(R, L)$ from a range of $R = 2$, 5, 10, 20 seg/sec and $L = 10$, 30 and 50. Only diagonal covariances are used for all mixture components.

### 4.4.4 Results

Figure (4.3) and (4.4) shows the percentage LID accuracy of SWRLM system for six languages (English, German, Hindi, Japanese, Mandarin and Spanish) using MLC and GC on LM scores respectively. These results are for 622 training utterances (English: 130, German: 81, Hindi: 179, Japanese: 65, Mandarin: 78 and Spanish: 89) in total, each of 45 seconds long 'story-bt' utterances and test data is of 20 utterances / language.

For SWRLM, there are two parameters for each language,

- Segment rate $R$, varying as $R = 2$, 5, 10 and 20 segments/seconds.
- SWU inventory $L$, varying as $L = 10$, 30 and 50.

Each plot in Figure (4.3) and (4.4) shows the LID accuracy ($y$-axis) vs each language English, German, Hindi, Japanese, Mandarin and Spanish ($x$-axis) for different segment rates $R = 2$, 5, 10 and 20 segments/seconds for a fixed SWU inventory ($L$).

We can observe the following from the plots:

1. SWRLM performance is consistent across the languages as the segment rate increases on both training and test data.
2. As the SWU inventory size increases LID performance improves by 50% (from 42.28% to 92.12%) on training data and by 28% (from 33.33% to 61.67%) on test data with MLC.

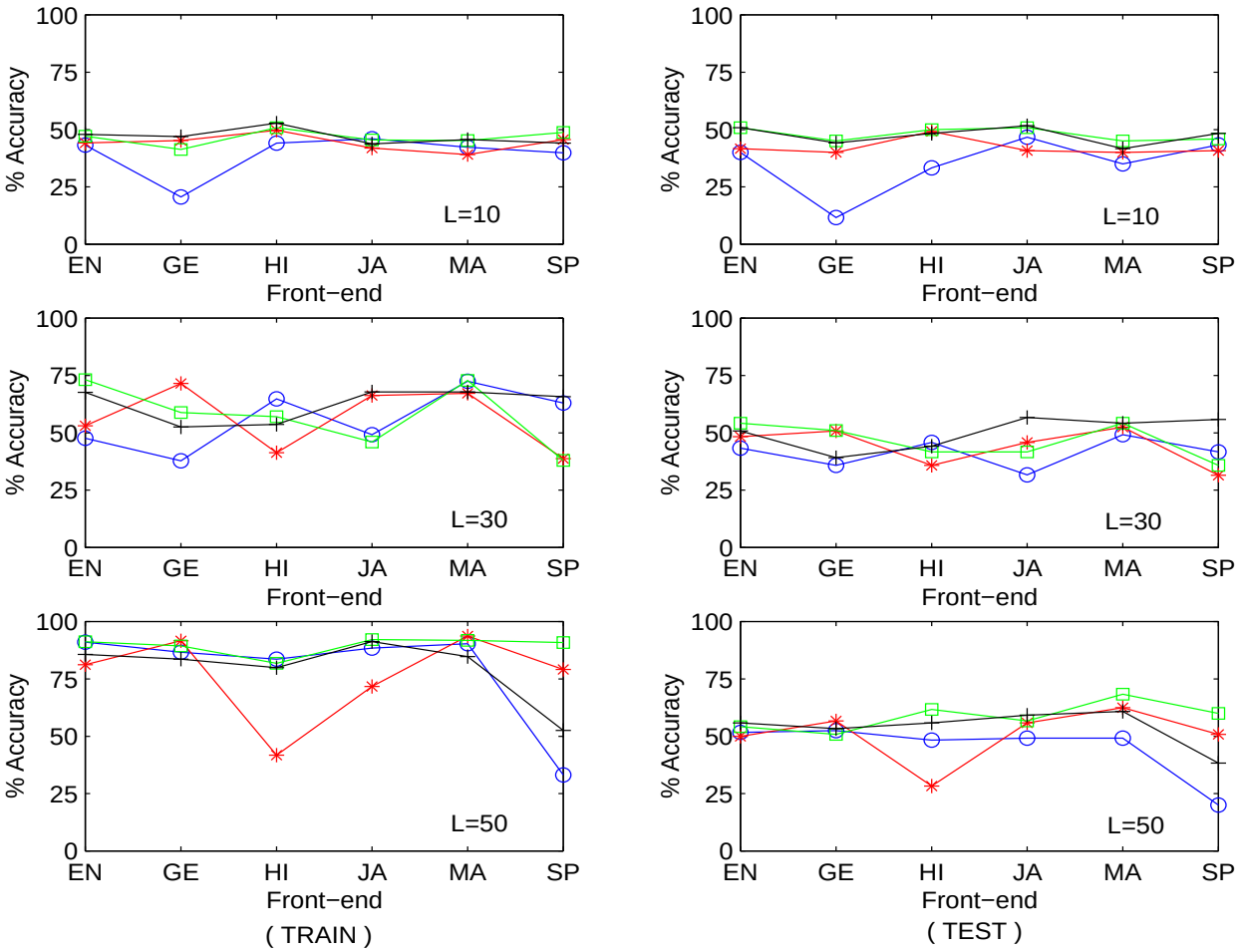



Figure 4.3: Percentage LID accuracy of SWRLM with LM score using MLC. Train data: 622 utterances from all languages, Test data: 20 utterances / language. Utterance length: 45 sec. Legend: R=2 (O), R=5 (*), R=10 (□), R=20 (+).

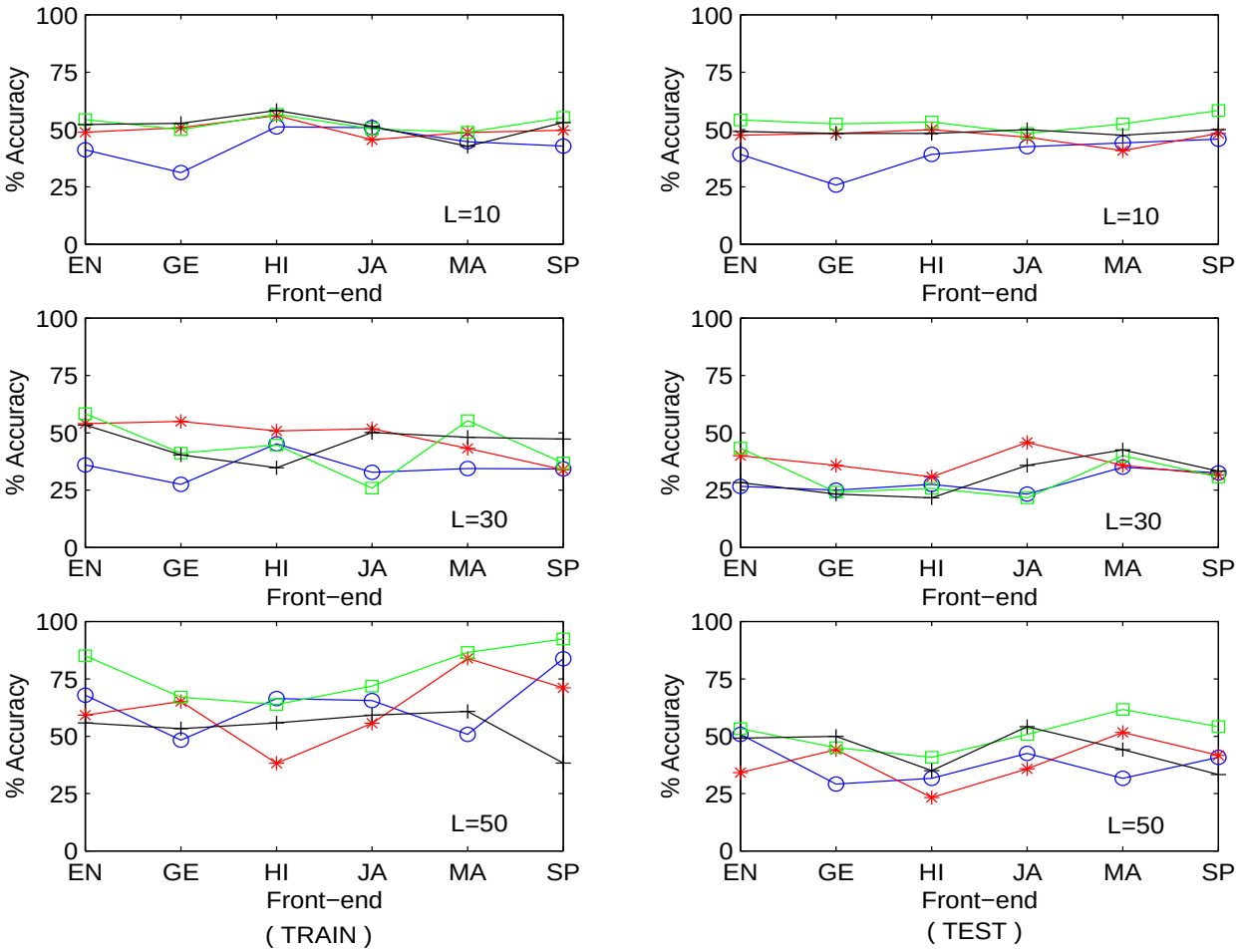



Figure 4.4: Percentage LID accuracy of SWRLM with LM score using GC. Train data: 622 utterances from all languages, Test data: 20 utterances / language. Utterance length: 45 sec. Legend: R=2 (O), R=5 (*), R=10 (□), R=20 (+).

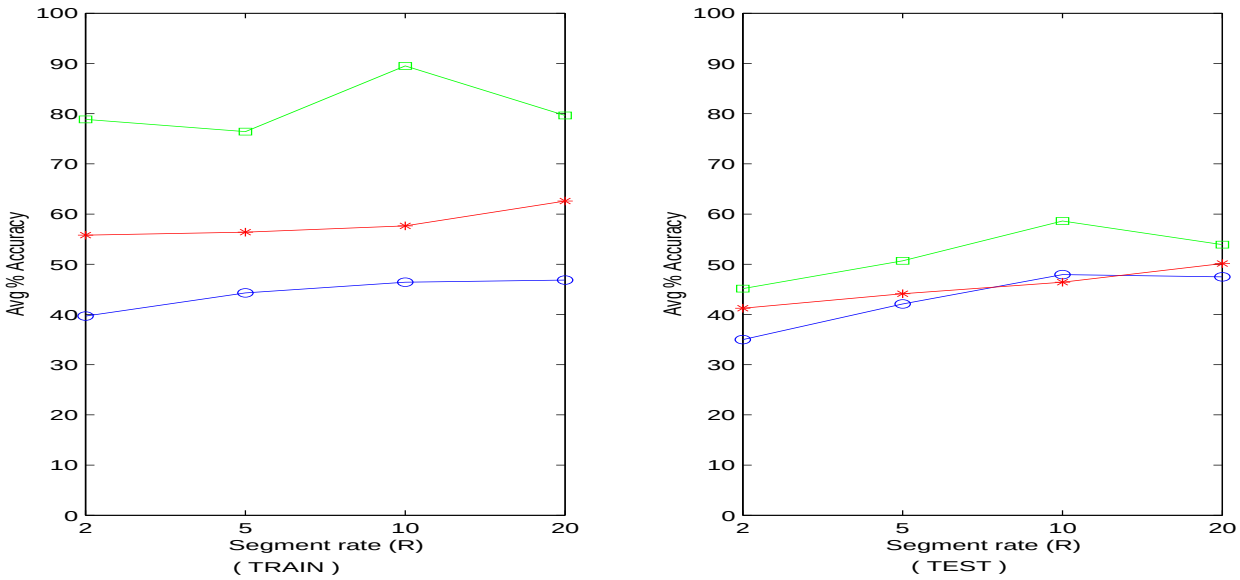



Figure 4.5: Average LID accuracy across languages of SWRLM with LM score using MLC. Legend: L=10 (O), L=30 (*), L=50 (□)

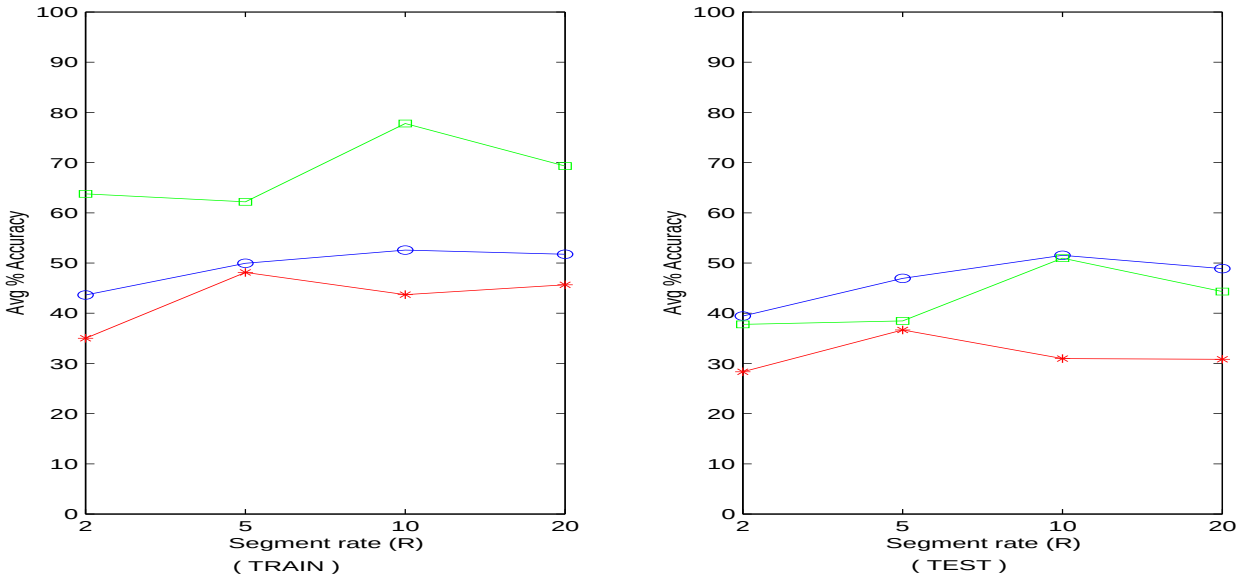



Figure 4.6: Average LID accuracy across languages of SWRLM with LM score using GC. Legend: L=10 (O), L=30 (*), L=50 (□)

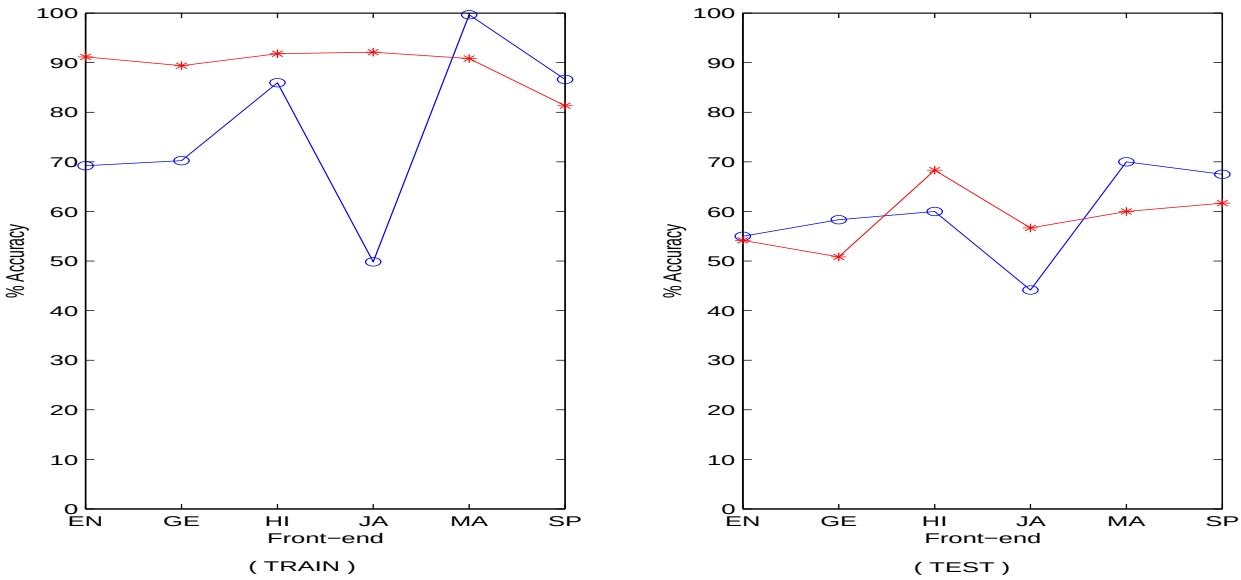



Figure 4.7: Percentage LID accuracy of SWRLM and PRLM on LM scores using MLC. Legend: PRLM (O), SWRLM (*)

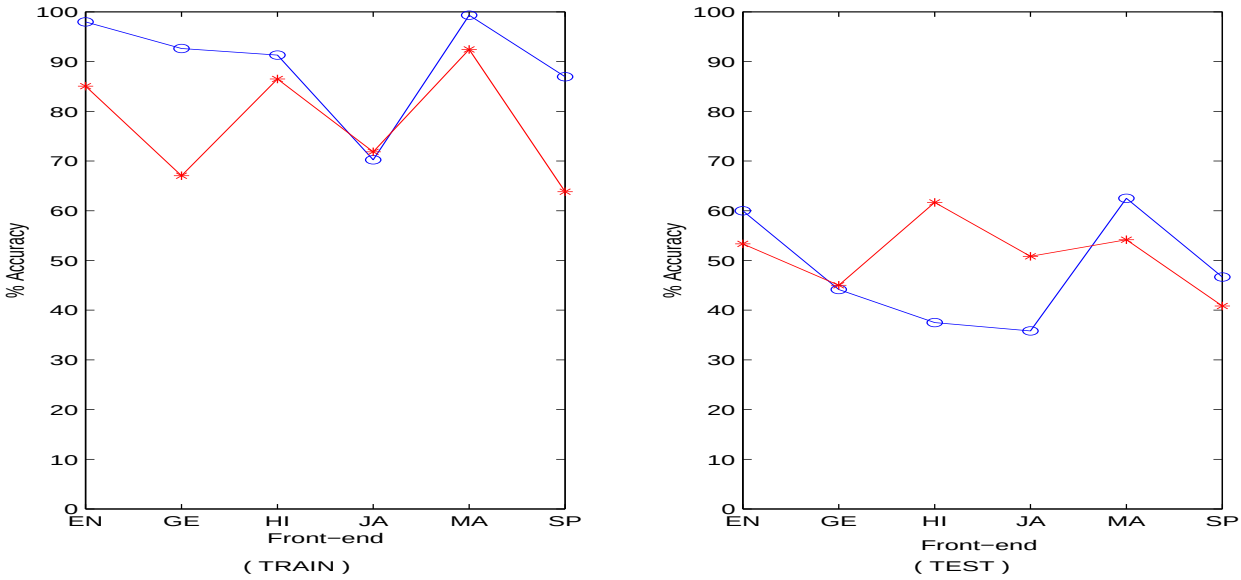



Figure 4.8: Percentage LID accuracy of SWRLM and PRLM on LM scores using GC. Legend: PRLM (O), SWRLM (*)

3. The best system performance is observed for $R = 10$ and $L = 50$ on both training and test data, 92.12% on train data and 68.33% on test data. This is the immediate reflection of the fact that $R = 10$ matches the normal speech rate of 10 phone / sec and $L = 50$ generates phone like units in the SWU inventory; their clusters match the phonetic units as languages have phone set sizes around this value.

4. Gaussian classifier performance is better than MLC for all $R$ and for SWU inventory of $L = 10$ across the languages.

5. MLC performance is almost better than Gaussian classifier for all $R$ and sub-word unit inventory size of $L = 30$ and 50 across the languages.

From Figure (4.5) and (4.6) it is clear that,

1. As segment rate increase from 2 to 10 LID performance increases from 39.12% to 89.52% on training data with SWU inventory sizes $L = 10$ and 50 respectively using MLC for classification.

2. On test data its performance increases form 35% to 58.61% with MLC as sub-word unit inventory size increased from $L = 10$ to $L = 50$ and segment rate $R = 2$ to $R = 10$.

3. With Gaussian classifier the LID performance increases from 35% to 77.79% on training data and 28.33% to 50.97% on test data as sub-word unit inventory size increased from $L = 10$ to $L = 50$ and segment rate $R = 2$ to $R = 10$.

Figure (4.7) shows the LID performance of PRLM system and SWRLM for $R = 10$ and $L = 50$. It can be observed that SWRLM on training data is satisfactorily better than PRLM as its performance across languages is consistent and its best performance is also comparable to PRLM system.

On test data PRLM and SWRLM systems performances are comparable.

From Figure (4.7) and (4.8) it can be noted that with Gaussian classifier SWRLM is comparable with PRLM across languages.

In focus of overall performance, we can conclude that the SWRLM is comparable to PRLM system, indicating its LID efficiency and the potential of the SWR front-end in SWRLM to replace the conventional PR in PRLM.

# Chapter 5

# Parallel sub-word recognition and language modeling (P-SWRLM)

## 5.1 Introduction

As we know that the sounds in the languages to be identified do not always occur in the one language used to train the front-end sub-word recognizer. Thus, it seems natural to look for a way to incorporate phones from more than one language into a SWRLM-like system. Hazen[6] has proposed to train a front-end recognizer from more than one language.

Alternatively, another approach is simply to run multiple SWRLM systems in parallel with the single language SWR each trained in different languages [33]. Therfore, P-SWRLM uses multiple front-end SWRs with each SWR followed by $N$ number of back-end LMs for an $N$ language LID task. In SWRLM, we need SWUs from only one language that is used to train the SWR, where as in P-SWRLM as it has multiple SWRs each SWR is having it's own language dependent SWU inventory. The language that is used to build the SWU inventory for a particular front-end SWR may not be a language in the LID task. As P-SWRLM has multiple front-end SWRs it will span more acoustic space as compared to SWRLM and the system performance will improve with addition of SWRLMs of different languages.

The P-SWRLM is studied with different channels, say M channels, can be $\mathcal{L}_1, \ldots, \mathcal{L}_M$, where all the $M$ languages are drawn from the $N$-language LID of languages $\mathcal{L}_1, \ldots, \mathcal{L}_N$ or the front-end can be from any language not part of of the LID task. Moreover, in this work we studied 6-language LID task ($N = 6$) and also use upto 6-channels in the P-SWRLM

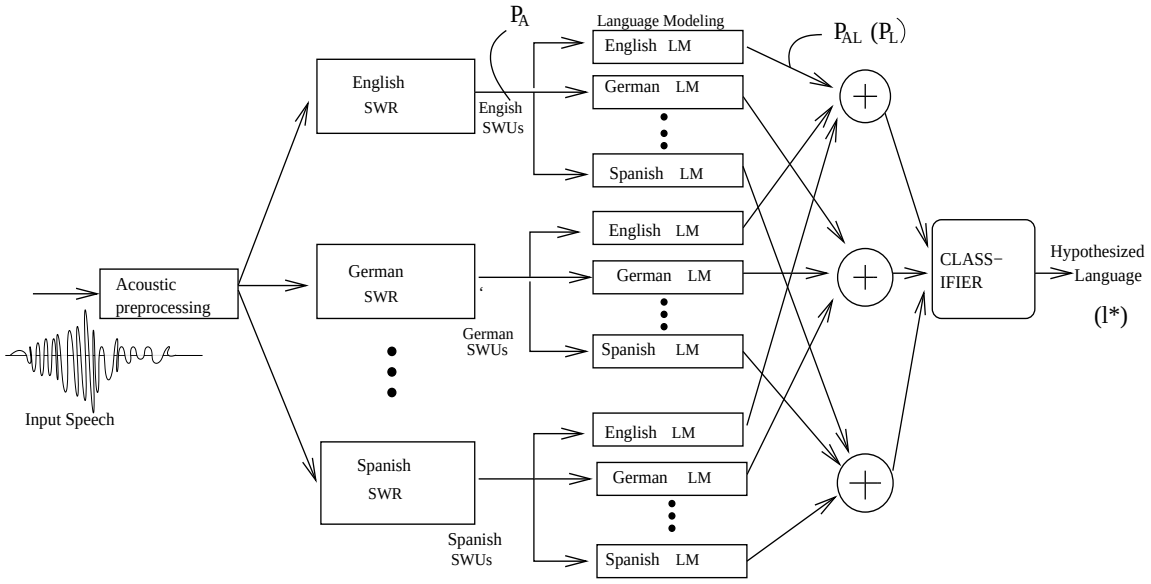



Figure 5.1: Parallel-SWRLM (P-SWRLM) system

system. Hence the front-end languages referred as $L_m$, $m = 1, \ldots, M$ for a $M$-channel P-SWRLM (with $M$ ¡ $N$ in this case). In P-SWRLM, we studied parallelism across languages, segment rates and acoustic space.

## 5.2 P-SWRLM

Figure 5.1 shows a typical P-SWRLM. Essentially it has multiple SWRLMs in parallel. Each front-end is followed by $N$ back-end language models (LMs) for an $N$ language LID task. In P-SWRLM each front-end SWR will have a language dependent sub-word unit (SWU) inventory. The important point to note here is that the SWU inventory is obtained without the need for manually labeled training data. Each front-end is followed by $N$ back-end language model (LM) which performs phonotactic analysis on the SWU label sequences obtained by the front-end tokenization of each channel.

A SWR of language $L_m$ in the P-SWRLM system has an inventory of sub-word units (SWUs) $\mathcal{P}_m = \{P_1, P_2, \ldots, P_L\}$ and corresponding sub-word HMM models $\mathcal{H}_m = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$ for the front-end of the language $L_m$. In the P-SWRLM system, the front-end training phase involves the design of the SWU inventory $\mathcal{H}_m$ (Sec. 3.4.1) for a set of

SWUs $\mathcal{P}_m$ for each of the language $L_m$.

### 5.2.1 Sub-word recognizer (SWR)

The SWR of language $L_m$ uses sub-word inventory $\mathcal{H}_m$ to tokenize an input utterances into a sequence of sub-word unit labels by optimal decoding as in 'connected word recognition' problem. A P-SWRLM having $M$ front-end SWRs will generate $M$ decoded sub-word label sequences and the associated acoustic likelihood scores corresponding to the best decoded path by Viterbi search.

Let the input utterance be a sequence of feature vector $\mathbf{O} = \{O_1, O_2, \ldots, O_T\}$. $\mathbf{O}$ is tokenized by the front-end SWR of language $L_m$ (i.e., channel $M$) into an optimal sequence of $K$ SWUs $\{P_{\bar{1}}, P_{\bar{2}}, \ldots, P_{\bar{k}}, \ldots, P_{\bar{K}}\}$, where $P_{\bar{k}} \in \mathcal{P}_m$ and $m = 1, \ldots, M$. The likelihood associated with this optimal string of SWU label sequence is calculated as $\mathbf{P}_A(m) = P(\mathbf{O}|\mathcal{H}_m)$ (acoustic score), which is given by

$$\mathbf{P}_A(m) = \max_{B,K} \sum_{k=1}^{K} \log(p(s_k|\lambda_{\bar{k}})) \tag{5.1}$$

$B = (b_0, b_1, \ldots, b_K)$, with $b_0 = 0$ and $b_K = T$, are the segment boundaries for any segmentation of the $T$ frames of $\mathbf{O}$. The $k^{th}$ segment $s_k = (O_{b_{k-1}+1}, \ldots, O_{b_k})$ is associated with the SWU model $\lambda_{\bar{k}}$, where, $\lambda_{\bar{k}}$ is the HMM model which has the maximum likelihood of generating segment $s_k$, from among $\mathcal{H}_m = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$; $p(s_k|\lambda_{\bar{k}})$ is the corresponding HMM likelihood and $P_{\bar{k}}$ is the corresponding SWU.

### 5.2.2 Language-model (LM)

In P-SWRLM each front-end will have $N$ back-ends. The back-end LMs in the P-SWRLM system performs phonotactic analysis; typically, it operates on the decoded SWU label sequence generated by the SWR.

For channel $m$, there are $N$ back-end LMs referred by $LM_{m,1}, \ldots, LM_{m,l}, \ldots, LM_{M,N}$ i.e., $LM_{m,l}$, $l = 1, \ldots, N$. The $LM_{m,l}$ is associated with a bigram distribution $\mathcal{B}_{m,l}$ given by $\mathcal{B}_{m,l} = \{p(i,j)\}$, $i, j = 1, \ldots, L$, where $\{p(i,j)\} = \{p(P_j|P_i)\}$ and $P_i, P_j \in \mathcal{P}_m$ are the SWU labels occurring in succession in the SWU label sequence $\{P_{\bar{1}}, \ldots, (P_{\bar{k}-1} = P_j), (P_{\bar{k}} =$

$P_i), \ldots, P_{\overline{K}}\}$ tokenized by the channel $m$ SWR for input utterances of language $L_l$.

$$\mathbf{P}_L(m,l) = P(\mathbf{O}|\mathcal{B}_{m,l}) = \sum_{k=2}^{K} \log p(P_{\overline{k}}|P_{\overline{k-1}}, \mathcal{L}_l) \tag{5.2}$$

### 5.2.3 Classifiers

#### 5.2.3.1 Maximum-likelihood classifier

The P-SWRLM system with $M$-channels, each with $N$ back-end LMs (for an $N$-language LID task) has $MN$ LM scores $P_L m, l$, $m = 1, \ldots, M$ and $l = 1, \ldots, N$. The scores $P_L(m,l)$, $m = 1, \ldots, M$ for any $l$ can be treated as multiple (i.e., $M$) evidences of an input utterance $\mathbf{O}$ belong to language $\mathcal{L}_l$, i.e., $P_m(\mathbf{O}/L_l) = P_L(m,l)$, $m = 1, \ldots, M$. Thus, a single LM score $P_L(l) = P(\mathbf{O}/L_l)$ can be obtained by simply adding the $M$ likelihoods $P_L(m,l)$, $m = 1, \ldots, M$ given by

$$\mathbf{P}_L(l) = \sum_{m=1}^{M} P_L(m,l) \tag{5.3}$$

There are $L$ LM scores $P_L(l)$, $l = 1, \ldots, L$ giving the language model likelihood $P(\mathbf{O}/L_l)$, which is the likelihood of an input utterance $\mathbf{O}$ to be from any language $L_l$. At this point, the $L$ LM scores are similar to that of a SWRLM system for $N$ language LID task (Sec. 3.8.

#### 5.2.3.2 Gaussian classifier

The P-SWRLM system shown in Figure 5.2, the back-end LMs of a SWR generates a score vector $\mathbf{x_m} = [x_m(1), \ldots, x_m(N)]^t$, where $x_m(l) = P_L(m,l)$, $m = 1, \ldots, M$ for a $N$-language task for any input utterance. Instead of combining $N$-language's $MN$ LM scores to yield $N$ scores of corresponding language, we will treat it as a $MN$ dimensional score vector as $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\} = \{x_1(1), \ldots, x_1(N), x_2(1), \ldots, x_2(N), \ldots, x_M(1), \ldots, x_M(N)\}$. Then The $N$-languages $\mathcal{L}_1, \ldots, \mathcal{L}_N$ correspond to $N$ clusters in $MN$-dimensional space. With this, the LID problem can be treated as a conventional $N$-class classification problem in the score-space of $MN$-dimensions given the $N$ likely clusters $\mathbf{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$. Given

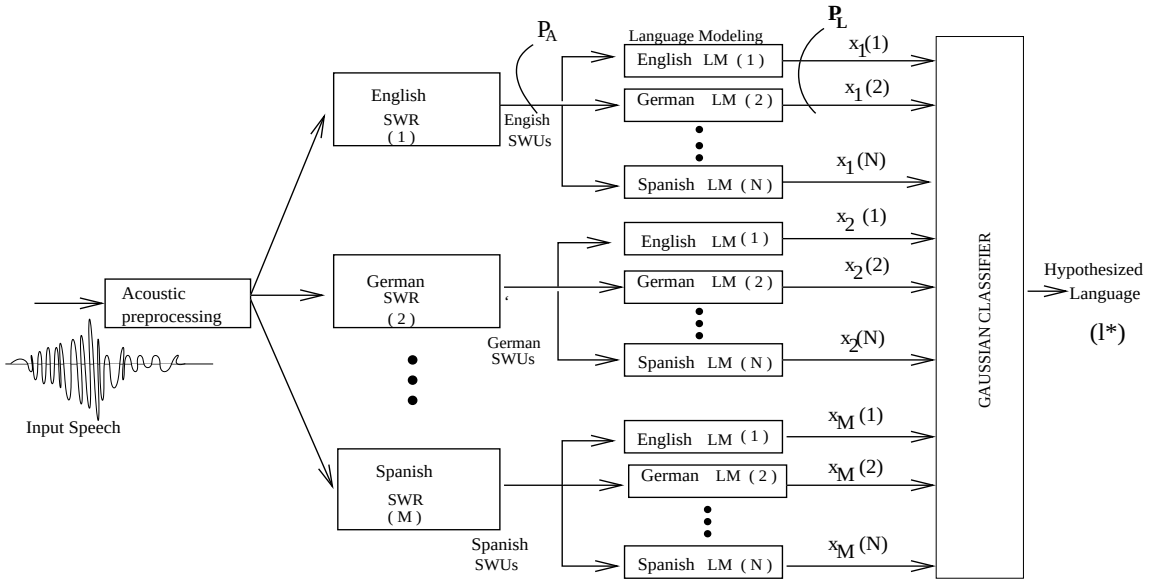



Figure 5.2: Parallel-SWRLM (P-SWRLM) system with Gaussian classifier

a test utterance will generate a score vector $\mathbf{y}$ of $MN$ dimensions then the classification problem is to classify $\mathbf{y}$ into one of $N$ classes $\{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$ (Sec. 4.3.2).

## 5.3 Experiments and results

We present here the results of the performance of the P-SWRLM for all all channels from 1 to 6 and for each channel we studied all possible combinations of the front-end languages. To study the P-SWRLM we chosen segment rate $R = 10$ and SWU inventory $L = 50$ as with this segmentation rate and SWU inventory size SWRLM is giving best performance and also $R = 10$ matches the normal speech rate of 10 phone / sec and $L = 50$ generates phone like units in the SWU inventory; their clusters match the phonemic units as languages have phone set sizes around this value. The P-SWRLM is studied with different classifiers like Maximum likelihood classifier (MLC) and Gaussian classifier.

### 5.3.1 Database and parameters of P-SWRLM

We study the performance of P-SWRLM system using OGI-TS database [17]. We have used the same languages and same training and test utterances as used to evaluate SWRLM

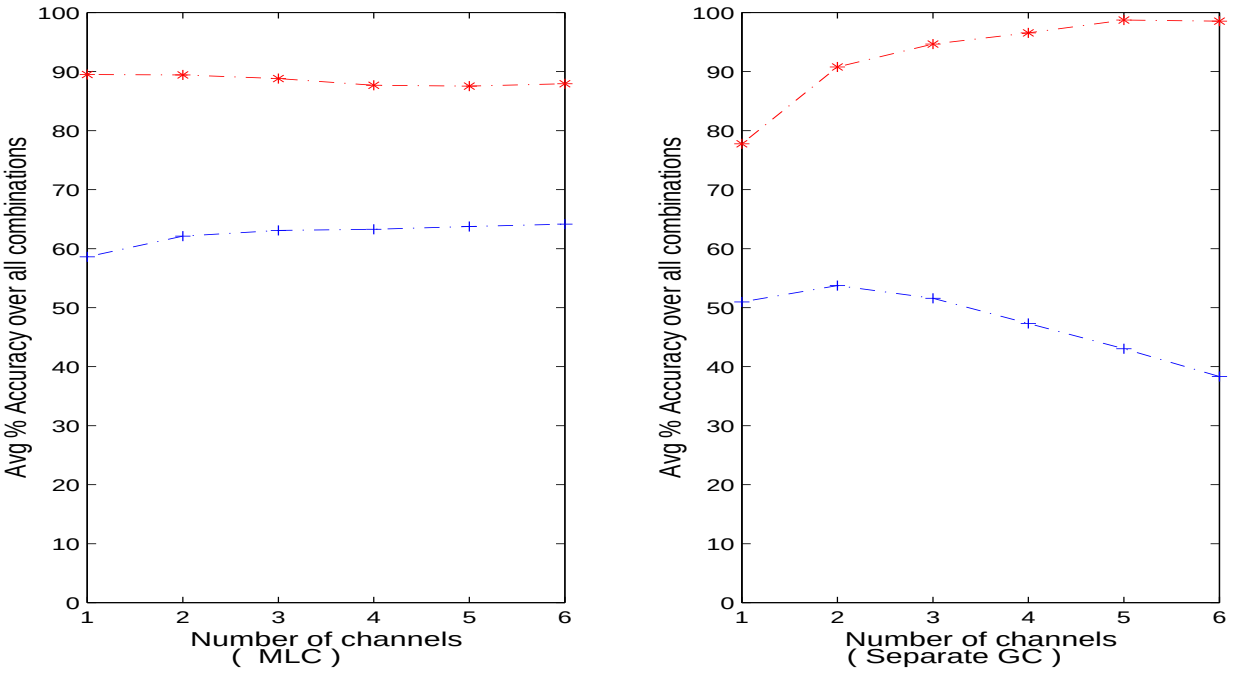



Figure 5.3: Average percentage LID accuracy of P-SWRLM with LM score using MLC and Separate Gaussian classifier over all possible combinations across number of channels. segmentation rate $R$=10 seg /sec, Code book size $L$=50 Train data: 622 utterances from all languages, Test data: 20 utterances / language. Utterance length: 45 sec. Legend: Train (*), Test (+).

system performance (Sec. 4.4.1).

We build a sub-word unit inventory with segmentation rate $R = 10$ and SWU inventory $L = 50$ as it is giving best results with SWRLM frame work. We used 3 state, left to right HMM models and 6 Gaussian mixtures per state for various $R$ and $L$ used in the design of the front-end SWRs (Sec. 4.4.2, 4.4.3).

### 5.3.2 Results

Figure (5.3) shows the average LID accuracy over all combinations of P-SWRLM system for all channels from 1 to 6 using MLC and Gaussian classifier on LM scores. These results are 622 training utterances (English: 130, German: 81, Hindi: 179, Japanese: 65, Mandarin: 78 and Spanish: 89) in total each of 45 seconds long 'story-bt' utterances and a test data of 20 utterances / language.

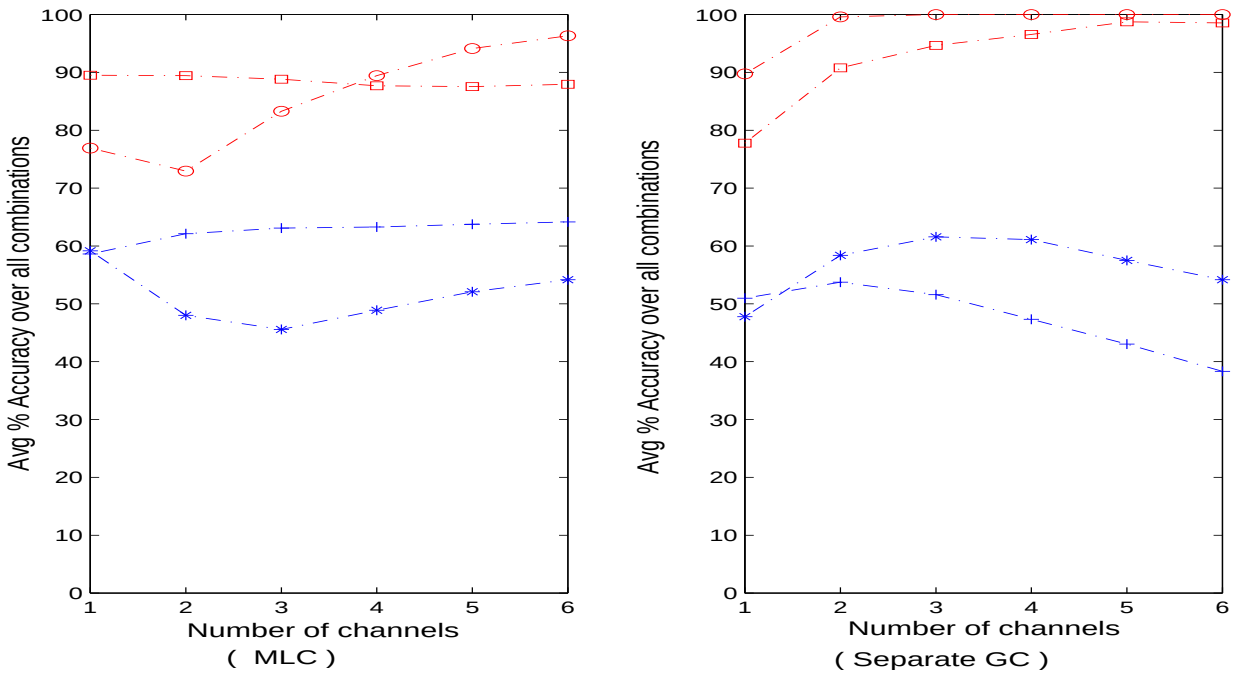



Figure 5.4: Average percentage LID accuracy of P-SWRLM and P-PRLM with LM score using MLC and Separate Gaussian classifier over all possible combinations across number of channels. segmentation rate $R$=10 seg /sec, Code book size $L$=50 Train data: 622 utterances from all languages, Test data: 20 utterances / language. Utterance length: 45 sec. Legend: Train: P-SWRLM (□), P-PRLM (o); Test: P-SWRLM (+), P-PRLM (*).

We can observe the following from the figures:

1. The P-SWRLM system performance increases with the increase in number of channels. In case of Gaussian classifier, the performance is increases from 77.78% to 98.55%) on training data where as with MLC, the system performance is almost the same even though the number of channels increases from 1 to 6. With one channel, its performance for MLC is 89.52% and with six channel it is 87.94%.

2. On test data, the P-SWRLM system performance increases with the number of channels with MLC but the Gaussian classifier performance decreases as the number of channels increases from 1 to 6.

3. The best system performance with MLC is 89.92% on training data with one channel and 64.16% on test data with six channels.

4. The best system performance with Gaussian classifier is 98.55% on training data with 6 channels and 53.74% on test data with two-channel P-SWRLM.

From above results, we can in general we can conclude that,

1. For the case of MLC, when a poor channel is combined with good channel then it may decrease the LID performance whereas it will be unaffected with Gaussian classifier.

2. However the Gaussian classifier performance is poor ( on test data) with increase in number of channels, most likely, due to increase in dimensionality of the score vector resulting in poor generality to test data (even though training data performance improves significantly with increase in number of channels). Which shows the potential of Gaussian classifier to perform better than MLC, even LID performance reaches 98.5% for 6-channel.

Fig. (5.4) shows the average LID accuracy for each of the 6-channels, for both P-PRLM and P-SWRLM with MLC and Gaussian classifier.

The following can be observed from this figure:

1. P-SWRLM performance on training data is 12.6%, 16.48% and 5.54% more than P-PRLM system with MLC for 1-channel, 2-channel and 3-channel respectively.

2. For Gaussian classifier, P-PRLM performs better than P-SWRLM.

3. P-PRLM performance on training data is 1.73%, 6.6% and 8.38% more than P-SWRLM with MLC for 4-channel, 5-channel and 6-channel case.

4. On test data, with MLC the P-SWRLM is significantly better than P-PRLM for all the different number of channels. Moreover, P-SWRLM also shows an improvement in performance with addition of channels using MLC.

5. The best performance observed is 92.12% on training data with 1-channel and 64.4% on test data with P-SWRLM for 6-channel case with MLC.

Table 5.1: **Table II**
Parallelism across $R$: Language Mandarin and $L$=50

| Classifier | 4-channel | | 1-channel | |
|---|---|---|---|---|
| | Training | Test | Training | Test |
| MLC | 88.90 | 58.83 | 91.80 | 68.33 |
| GC | 94.05 | 46.66 | 86.49 | 61.66 |

Table 5.2: **Table II**
Parallelism across SWU inventory: Language Mandarin and $R$=10

| Classifier | 3-channel | | 1-channel | |
|---|---|---|---|---|
| | Training | Test | Training | Test |
| MLC | 79.26 | 64.16 | 91.80 | 68.33 |
| GC | 90.99 | 45.83 | 86.49 | 61.66 |

We also studied the parallelism across segment rate $R$ with $L = 50$ and fixing the front-end language as mandarin, results are shown in **Table I** and parallelism across $L$ with $R = 10$, results are shown in **Table II**.

From these tables it is clear that,

1. On training data performance of LID is good for 4-channel, parallelism across $R$ as compared to 3-channel, parallelism across $L$ with both the classifiers, MLC and GC.

2. On test data Gaussian classifier is comparable for both parallelisms, where as MLC is performing better.

3. The performance of 1-channel itself is better than parallelism across $R$ or $L$ on test data.

# Chapter 6

# Future work

- Language independent SWR has to be studied for SWRLM as they may span entire acoustic space as its SWU inventory is obtained from more than one language. Where as the language dependent SWR of SWRLM may not span full acoustic space.
- Future work can focus on alternative segmentation methods which will give better performance with lower segmentation rate.
- We used two types of classifiers viz., Maximum likelihood classifier (MLC) and Gaussian classifier (GC). Alternatively Neural network based classifier can be studied.
- As with phone recognition systems mono/poly phone approach gives as good a performance as any other phone recognition approaches. In a similar way mono/poly sub-words can also be studied.
- Lexicon based LID with sub-word units is an interesting and important issue for further research.
- We prepared Indian language database (ILDB). This database needs more supervision and has to be scrutinized after the corrections are made, LID on Indian languages can be performed. Since this database has no phonetic transcription, sub-word based approach has to be used for further research on Indian languages.

# Chapter 7

# Conclusions

We studied PSWR, SWRLM and P-SWRLM systems for LID as an efficient alternative to phone based system viz., PPR, PRLM and P-PRLM. While all the three phone based systems requires manually labeled training data, sub-word based systems do not require manual labels in any of the languages. The sub-word based system performance is comparable to its equivalent phone based systems, indicating its LID efficiency and the potential of the SWR front-end to replace phone recognizer (PR) systems in these LID systems.

# References


[1] M. Bacchiani and M. Ostendorf. Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, 29:99–114, 1999.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification.* John Wiley & Sons, Inc., 2nd edition, 2001.

[3] T. J. Hazen. Automatic language identification using a segment-based approach. Master's thesis, Massachusetts Institute of Technology, Sep 1993.

[4] T. J. Hazen and V. W. Zue. Automatic language identification using a segment-based approach. In *Proc. Eurospeech*, volume 2, pages 1303–1306, Sep 1993.

[5] T. J. Hazen and V. W. Zue. Recent improvements in an approach to segment-based automatic language identification. In *Proc. of International Conference on Spoken Language Processing*, volume 4, pages 1883–1886, Sep 1994.

[6] T. J. Hazen and V. W. Zue. Segment-based automatic language identification. *Journal of Acoustic Society of America*, 101(4):2323–2331, Apr 1997.

[7] Vishweswar G. Hiremath. Automatic language identification using phone recognition. Master's thesis, Dept. ECE, National Institute of Technology, Suratkal, Feb. 2003. M. Tech. Project carried out in Dept. of ECE, Indian Institute of Science, Bangalore.

[8] A. K. V. Sai Jayram, V. Ramasubramanian, and T. V. Sreenivas. Automatic

language identification using acoustic sub-word units. In *Proc. of International Conference on Spoken Language Processing*, Denver, Colorado, Sep 2002.

[9] A. K. V. Sai Jayram, V. Ramasubramanian, and T. V. Sreenivas. Robust parameters for automatic segmentation of speech. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages I–513–I–516, Orlando, Florida, May 2002.

[10] A. K. V. Sai Jayram, V. Ramasubramanian, and T. V. Sreenivas. Language identification using parallel sub-word recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, Apr 2003.

[11] C. H. Lee, B. H. Juang, F. K. Soong, and L. R. Rabiner. Word recognition using whole word and subword models. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 683–686, 1989.

[12] C. H. Lee, F. K. Soong, and B. H. Juang. A segment model based approach to speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 501–504, New York, April 1988.

[13] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantization design. *IEEE Transactions on Communications*, COM-28:84–95, Jan 1980.

[14] M. A. Lund, K. Ma, and H. Gish. Statistical language identification based on untranscribed training data. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 793–796, Atlanta, Georgia, May 1996.

[15] Y. K. Muthusamy. *Segmental approach to automatic language identification.* PhD thesis, Oregon Graduate Institute of Science & Technology, 1993.

[16] Y. K. Muthusamy, E. Barnard, and R. A. Cole. Reviewing automatic language identification. *IEEE Signal Processing Magazine*, 11(4):33–41, Oct 1994.

[17] Y. K. Muthusamy, R. A. Cole, and B. T. Oshika. The OGI multi-language telephone speech corpus. In *Proc. of International Conference on Spoken Language Processing*, pages 895–898, 1992.

[18] K. K. Paliwal. Lexicon-building methods for an acoustic sub-word based speech recognizer. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990.

[19] Hanne Paul Dalsgaard, Oven Andersen and Bojan. Language-identification using language-dependent phonemes and language-independent speech units. In *Proc. of International Conference on Spoken Language Processing*, pages 1808–1811, 1996.

[20] Oven Andersen Paul Dalsgaard and Barry. On the use of data-driven clustering technique for identification of poly- and mono-phonemes for four european languages. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages I–121–I–123, 1994.

[21] L. R. Rabiner and B.-H. Juang. *Fundamentals of speech recognition.* Prentice Hall, 1993.

[22] Larry Gillick Sergio Mendoza and Yoshiko Ito. Automatic language identification using large vocabulary continuous speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 785–788, 1996.

[23] T. Svendsen, K. K. Paliwal, E. Harbog, and P. O. Husey. An improved sub-word based speech recognizer. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 108–111, 1989.

[24] T. Svendsen and F. K. Soong. On the automatic segmentation of speech signals. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 77–80, Dallas, 1987.

[25] Ann E. Thyme-Gobbel and Sandra E. Hutchins. On using prosodic cues in automatic language identification. In *Proc. of International Conference on Spoken Language Processing*, pages 1768–1771, 1996.

[26] R. C. F. Tucker, M. J. Carey, and E. S. Paris. Automatic language identification using sub-word models. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 301–304, Apr 1994.

[27] A. K. V. Sai Jayram V. Ramasubramanian and T. V. Sreenivas. Language identification using parallel phone recognition. In *Workshop on Spoken Language Processing*, pages 109–116, Mumbai, India, Jan 2003. Tata Institute of Fundamental Research.

[28] P. C. Woodland and S. J. Young. The HTK tied-state continuous speech recognizer. In *Proc. Eurospeech*, volume 3, pages 2207–2210, Sep 1993.

[29] Y. Yan. *Development of an approach to language identification based on language-dependent phone recognition.* PhD thesis, Oregon Graduate Institute of Science & Technology, Oct 1995.

[30] Y. Yan and E. Barnard. An approach to automatic language identification based on language-dependent phone recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3511–3514, May 1995.

[31] Y. Yan and E. Barnard. An approach to language identification with enhanced language model. In *Proc. Eurospeech*, pages 1351–1354, Madrid, Sep. 1995.

[32] Y. Yan, E. Barnard, and R. A. Cole. Development of an approach to automatic language identification based on phone recognition. *Computer Speech and Language*, 10(1):37–54, 1996.

[33] M. A. Zissman. Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing*, 4(1):31–44, Jan 1996.

[34] M. A. Zissman and K. M. Berkling. Automatic language identification. *Speech Communication*, 35(1-2):115–124, Aug 2001.