# A Vector Space Modeling Approach to Spoken Language Identification

Haizhou Li, *Senior Member, IEEE*, Bin Ma, *Senior Member, IEEE*, and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—We propose a novel approach to automatic spoken language identification (LID) based on vector space modeling (VSM). It is assumed that the overall sound characteristics of all spoken languages can be covered by a universal collection of acoustic units, which can be characterized by the acoustic segment models (ASMs). A spoken utterance is then decoded into a sequence of ASM units. The ASM framework furthers the idea of language-independent phone models for LID by introducing an unsupervised learning procedure to circumvent the need for phonetic transcription. Analogous to representing a text document as a term vector, we convert a spoken utterance into a feature vector with its attributes representing the co-occurrence statistics of the acoustic units. As such, we can build a vector space classifier for LID. The proposed VSM approach leads to a discriminative classifier backend, which is demonstrated to give superior performance over likelihood-based $n$-gram language modeling (LM) backend for long utterances. We evaluated the proposed VSM framework on 1996 and 2003 NIST Language Recognition Evaluation (LRE) databases, achieving an equal error rate (EER) of 2.75% and 4.02% in the 1996 and 2003 LRE 30-s tasks, respectively, which represents one of the best results reported on these popular tasks.**

***Index Terms*—Acoustic segment models (ASMs), artificial neural network (ANN), spoken language identification, support vector machine (SVM), text categorization, vector space model (VSM).**

## I. Introduction

AUTOMATIC spoken language identification (LID) is the process of determining the identity of the language corresponding to a given set of spoken queries. It is a key technology in many applications such as multilingual conversational systems [1], spoken language translation [2], multilingual speech recognition [3], and spoken document retrieval [4]. LID is also a topic of great importance in areas of intelligence and security, where the language identities of recorded messages and archived materials need to be established before any information can be extracted. For voice surveillance over telephone network, LID technology also makes massive, online language routing possible.

A spoken language can be identified using information from multiple sources. In the past few decades, researchers have explored many speech features which include articulatory parameters [5], acoustic features [6], prosody [7], [8], phonotactic [9], [10], and lexical knowledge [11]. Taking advantage of recent advances in continuous speech recognition [12], statistical approaches [13]–[21] have been developed by exploiting techniques in acoustic modeling and $n$-gram language modeling. Recently, investigators at the MIT Lincoln Laboratory have reported promising results using *shifted-delta-cepstral* acoustic features [18] in Gaussian mixture model (GMM), which can be seen as a one-state hidden Markov model (HMM) [22], and is commonly used in text-independent speaker recognition [23]. Another successful approach is to characterize a spoken language using probability distributions of spectral features in the form of linguistically defined units such as phones and syllable-like units [9], [10], [24], where phone models are used to convert speech utterances into sequences of phone symbols, with the resulting acoustic likelihood scores. An interpolated phone-based $n$-gram language model is then constructed for each language, and to derive phonotactic scores. This is also referred to as the phonotactic approach, such as parallel parallel Phone Recognizers followed by Language Models (PRLM) [10] which uses multiple single-language phone recognizers as the frontend and language-dependent language models as the backend. The phonotactic approach has been shown to provide superior performance on National Institute of Standards and Technology (NIST) Language Recognition Evaluation (LRE) tasks especially when fused with acoustic scores [15].

It is generally agreed upon that the fusion of multiple phonotactic features improves performance. For example, the parallel PRLM approach employs parallel phone recognizers to derive multiple phonotactic features. Others have found that phonotactic features from multiresolution analysis, such as phone unigram, bigram, and trigram, complement each other [9], [19], [25]. However, it is not so straightforward to fuse the diverse features extracted from different sources and/or at different levels of resolutions. One of the solutions could be to find a universal set of sound units that represents all spoken languages [9], [19], [21]. In this way, spoken utterances from different languages can be tokenized into sequences of common sound units. We then find a way to represent the multiresolution phonotactic statistics derived from the sound unit sequences. This paper is motivated by the desire to find such a solution.

The fundamental question that arises is whether phones, or other similar linguistically defined units, are really needed to model and identify spoken languages. When human beings are constantly exposed to a language without being given any linguistic knowledge, they learn to determine the language identity by perceiving some of the speech cues in the specific language. For example, an English-speaking listener can often appreciate the syllabic nature of Japanese. It is also noted that in human perceptual experiments, listeners with a multilingual background often perform better than monolingual listeners in identifying unfamiliar languages [26], [27]. These reasons motivate us to explore useful speech attributes in a specific language for LID, along the same lines as a recently proposed automatic speech

Manuscript received August 11, 2005; revised March 13, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Timothy J. Hazen.

H. Li and B. Ma are with the Institute for Infocomm Research, Singapore 119613 (e-mail: hli@i2r.a-star.edu.sg; mabin@i2r.a-star.edu.sg).

C.-H. Lee is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: chl@ece.gatech.edu).

Digital Object Identifier 10.1109/TASL.2006.876860

attribute transcription (ASAT) paradigm for automatic speech recognition [28]. In order to do so, it is desirable to use a universal set of acoustically defined, rather than linguistically defined, units to cover the acoustic characterization of the intended spoken languages.

To apply this concept of universal unit set to language discrimination, we can relate to the fact that the entropy of English can be effectively reduced when high-order statistics of letters are computed. For example, of the 26 English letters plus the space character, letters, such as "n," "s," and "t," occur more frequently in normal text than some other letters, like "x" and "q." By incorporating the first-order statistics, or unigram, of letters, the entropy of English text can easily be reduced from 4.76 to 4.03 bits. When letter bigrams and trigrams are added, the entropy is further reduced to 2.8 and 1.34 bits, respectively [29]. This set of statistics can also be used to decipher encoded letters and discriminate among languages, even if no extra dictionary information is explicitly utilized. The fundamental question that follows is whether the same concept can be applied to spoken language identification. This question has motivated the previously reported phonotactic approaches [9], [10].

If we can tokenize speech with a manageable set of "spoken letters" and develop models to decode them from spoken utterances, then it is clear that the statistics of these spoken letters and their co-occurrences can be used to discriminate one spoken language from another. Although common sounds are shared considerably across spoken languages, the statistics of these sounds, such as phone $n$-gram, can differ considerably from one language to another. In fact, phonotactic scores from phone language models have been shown to be useful in LID [9], [10]. An interesting generalization through acoustic units is to represent any spoken utterance (also referred to as *spoken document* when presented in the form of "spoken letters") with a high-dimensional feature vector, where each element carries sound co-occurrence statistics. This is similar to the feature vector in the latent semantic indexing (LSI) [30] representation of text documents, which is commonly used in information retrieval (IR) systems [31]. Such statistics are considered to be salient features for indexing and retrieving documents. Other useful speech and language features, such as prosodic and syllabic content, can be incorporated into this feature vector as well.

We make the assumption that the sound characteristics of all spoken languages can be covered by a universal set of automatically derived acoustic units with no direct link to phonetic definitions. Their corresponding models, called acoustic segment models (ASMs) [32], can be used to decode spoken utterances into strings of such units. HMMs [22] are often used to model the collection of ASMs, which can be established bottom up in an unsupervised manner. The ASMs have been used to construct an acoustic lexicon for isolated word recognition with high accuracy [32]. In the acoustic lexicon, the ASMs form spoken letters, and serve as a basis from which we derive feature vectors for spoken documents and build language classifiers for automatic language identification. LID as such can be formulated as a text categorization (TC) [33] problem, in which feature extraction and classifier design are two major research components.

The rest of the paper is organized as follows. In Section II, we summarize relevant existing approaches, and introduce the idea of acoustic segment modeling and the vector space modeling (VSM)-based LID framework. We also introduce the working databases. In Section III, we discuss acoustic segment modeling in detail, in relation to augmented phoneme inventory. In Section IV, we establish the notion of acoustic letters and acoustic words, and compute their statistics. In this way, we perform feature extraction from spoken documents. This is sometimes referred to as a *bag-of-sounds* representation [13], analogous to the *bag-of-words* representation [31] of text documents that is often used in information retrieval. In Section V, we illustrate the simplicity of vector-based classifier design with two conventional methods, namely support vector machines (SVM) [34]–[36] and artificial neural networks (ANN) [37]–[39]. In Section VI, we conduct a series of experiments to discuss several practical issues, including acoustic resolution, acoustic coverage, feature selection, and an analytic study of frontend and backend. We report the results in the 1996 and 2003 NIST Language Recognition Evaluation tasks. Finally, in Section VII, we summarize our findings and discuss future work.

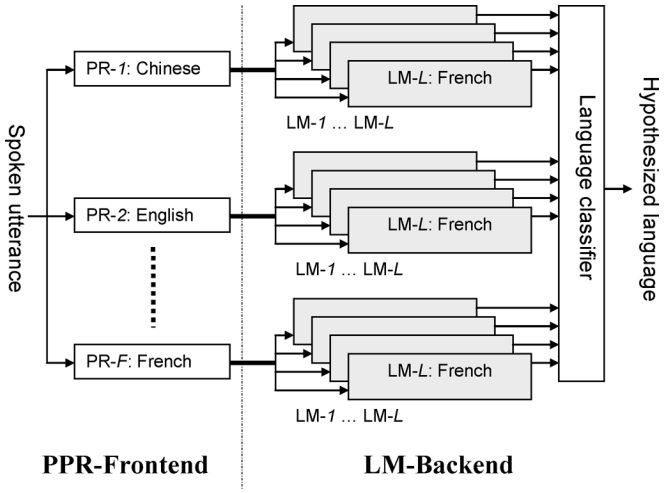



Fig. 1. Block diagram of a PPR-LM LID system.

## II. Problem Formulation

In this section, we first briefly discuss the prior work cast in the formalism of phone recognition (PR) and phone-based language modeling (LM). Then we propose the ASM acoustic modeling and vector space modeling for language classification.

A typical parallel PRLM LID system is illustrated in Fig. 1, where a collection of $F$ parallel phone recognizers (PPR frontend) serves as the voice tokenizers. Each recognizer is defined by a single language. Phone recognition is carried out by a Viterbi search using a fully connected null-grammar network of phones without language models. In this paper, we do not refer PPR to a collection of joint acoustic-phonotactic phone recognizers as defined in [10]. Rather, we use PPR as the acronym for the parallel phone recognizers that do not use language models in the phonetic decoding. A PPR frontend converts a spoken utterance into $F$ sequences of token symbols, otherwise known as spoken documents. Then, a set of $L$ $n$-gram phone language models estimates the likelihood phonotactic scores for the spoken documents in order to produce classification decisions. The language models and the classifier are also referred to as the backend.

Generally speaking, a probabilistic language classifier can be formulated as follows. Given a sequence of feature vectors $O$ of length $\tau$, $O = \{o_1, o_2 \ldots, o_\tau\}$, we can express the *a posteriori* probability of language $l$ using Bayes Theorem, as follows:

$$\begin{aligned} P(l|O) &= P(O|l)P(l)/P(O) \\ &= \sum_{\forall T} P\left(O|T, \lambda_f^{\mathrm{AM}}\right) P\left(T|\lambda_{f,l}^{\mathrm{LM}}\right) P(l)/P(O) \end{aligned} \quad (1)$$

where $T$ is a candidate token sequence and $\lambda_f^{\mathrm{AM}}$ is the acoustic model for the $f$th phone recognizer, while $\lambda_{f,l}^{\mathrm{LM}}$ is the $l$th language model for the $f$th phone recognizer. Now we can apply the *maximum a posteriori* decision rule as

$$\hat{l} = \arg\max_{f,l} \sum_{\forall T} P\left(O|T, \lambda_f^{\mathrm{AM}}\right) P\left(T|\lambda_{f,l}^{\mathrm{LM}}\right) P(l)/P(O). \quad (2)$$

The first term on the right-hand side of (2) is the probability of $O$ given $T$ and its acoustic model $\lambda_f^{\mathrm{AM}}$. The second term is the language probability of $T$ given the language model $\lambda_{f,l}^{\mathrm{LM}}$. Finally, the last term is the *prior* probability $P(l)$, which is often assumed to be equal for all languages. The observation probability $P(O)$ is not a function of the language and can be removed from the optimization function.

The exact computation in (2) involves summing over all possible token sequences. In practice, we prefer to approximate the summation by finding the most likely phone sequence $\hat{T}_f$ for the phone recognizer $f$ by using a Viterbi search

$$\hat{T}_f = \arg\max_{T \in B_f} P\left(O|T, \lambda_f^{\mathrm{AM}}\right) \quad (3)$$

where $B_f$ is the set of all possible token sequences from the $f$th phone recognizer. As such, a solution to (2) can be approximated as

$$\hat{l} \approx \arg\max_{f,l} \left[\log P\left(O|\hat{T}_f, \lambda_f^{\mathrm{AM}}\right) + \log P\left(\hat{T}_f|\lambda_{f,l}^{\mathrm{LM}}\right)\right]. \quad (4)$$

Assuming that there are $F$ parallel single-language phone sets, we would like to use this limited set of language-dependent acoustic phone models to approximate the acoustic space for $L$ languages. Typically, we have $F \ll L$. As a result, after a language is decoded by the tokenizer of its competitive language, it needs to be evaluated by a set of $F \times L$ language models to establish their comparability. The system formulated by (3) and (4) is known as parallel PRLM [10]. In this paper, we rename parallel PRLM as PPR-LM in order to identify its PPR frontend and LM backend.

In the prior work, researchers also looked into a language-independent phone recognizer using a set of universal acoustic units, or phones, that is common for all languages. The formulation of (3) and (4) can be simplified as a two-step optimization

$$\hat{T} = \arg\max_{T \in B} \left[\log P(O|T, \lambda^{\mathrm{AM}})\right] \quad (5)$$

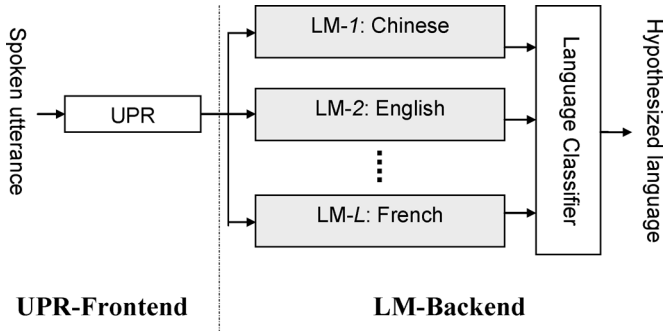



Fig. 2. Block diagram of a UPR-LM LID system.

$$\hat{l} = \arg\max_{l} \left[\log P\left(\hat{T}|\lambda_l^{\mathrm{LM}}\right)\right] \quad (6)$$

where $B$ is the set of all possible token sequences decoded by the set of universal acoustic units. The acoustic probability on the right-hand side of (5) is now the same for all competing languages. Only a language-specific score on the right-hand side of (6) is used for score comparison to select the identified language. As such, the PPR-LM system can be simplified as the UPR-LM system with universal phone recognition (UPR) frontend as in Fig. 2.

A number of UPR-LM systems have been proposed along these lines, such as the automatic language identification (ALI) system [9], the single-language PRLM system [10], and the language-independent phone recognition approach [20], [40]. However, the training of phone sets in these systems relies on the requirement of phonetic transcription for all training utterances.

### *A. Acoustic Segment Modeling for Voice Tokenization*

To obtain the set of universal acoustic units, some prior work proposed pooling all phone units from available training data for all languages [9], [19], [21], [41], e.g., the International Phonetic Alphabet or Worldbet [42]; others used phone clustering techniques to derive a more compact common set [20]. In general, the acoustic units are trained on phonetically transcribed databases. In this paper, we advocate a new way of training the set of universal acoustic units using ASM approach, where acoustic models are trained in a self-organized and unsupervised manner. In doing so there are two obvious advantages. 1) The unsupervised strategy allows the frontend to adapt easily towards new languages without the need of phonetic transcription, as will be discussed in Section III. 2) The universal acoustic units maintain the flexibility to be partitioned into subsets to work for the PPR frontend as in Fig. 1, as will be discussed in Section VI-E.

Torres-Carrasquillo *et al.* [25] reported a frame-based GMM tokenizer that circumvents the need for phonetic transcription. However, it does not really group frames into a token, but "quantizes" each frame vector into a symbol. To capture speech dynamics beyond a frame, the phone model is widely adopted as the voice tokenization unit. The proposed ASM approach in this paper can be seen as an extension to various prior work in pursuit of language-independent acoustic phone models [9], [10], [19]–[21], [40].

### B. VSM for Language Classification

VSM has become a standard tool in IR systems since its introduction decades ago [31]. It uses a vector to represent a text document. One of the advantages of the method is that it allows the discriminative training of classifier over the document vectors. We can derive the distance between documents easily as long as the vector attributes are well defined characteristics of the documents. Each coordinate in the vector reflects the presence of the corresponding attribute.

Inspired by the idea of document vector in text categorization research, we would like to investigate a new concept of the LID classifier using vector space modeling. A spoken language will always contain a set of high-frequency function words, prefixes, and suffixes, which are realized as acoustic unit substrings in the spoken document. Individually, those substrings may be shared across languages. Collectively, the pattern of their co-occurrences discriminates one language from another. Let us take English and Chinese Mandarin as an example. We list the top ten most frequent English and Chinese Mandarin words as follows:

- *the ... of ... to ... a ... and ... in ... that ... for ... one ... is ...* (English)
- *de*的 *... yi*一 *... he*和 *... zai*在 *... shi*是 *... le*了 *... bu*不 *... you*有 *... zhe*这 *... ge*个 *...* (Mandarin).

By coincidence, the most common English word (*the*) and most common Chinese Mandarin word (*de*) share similar pronunciations, resulting in similar voice tokenization statistics. However, we can rely on the co-occurrence information of these same high-frequency words to discriminate one language from another.

Suppose that the sequence of feature vectors $O$ is decoded into a sequence of $\Omega$ acoustic units $\hat{T} = \{t_1, \ldots, t_\pi, \ldots, t_\Omega\}$, each unit is drawn from the universal ASM inventory of $J$ models in a UPR frontend, $t_\pi \in \{w_1, w_2, \ldots w_J\}$. One is able to establish a high-dimensional salient feature vector which is language independent, where all of its elements are expressed as the $n$-gram probability attributes $p(w_n|w_1, \ldots w_{n-1}) = p(t_\pi = w_n|t_{\pi-1} = w_1, \ldots, t_{\pi-n+1} = w_{n-1})$. Its dimensionality is equal to the total number of $n$-gram patterns needed to highlight the overall phonotactic behavior of an utterance as

$$\overline{\lambda} = (p(w_1), \ldots, p(w_2|w_1), \ldots, p(w_3|w_1, w_2), \ldots)\,. \quad (7)$$

The vector space modeling approach evaluates the goodness of fit, or score function, using vector-based distance, such as an inner product

$$P\left(\hat{T}|\lambda_l^{\mathrm{LM}}\right) \propto \overline{\lambda}^T \cdot \omega_l \quad (8)$$

where $\omega_l$ is a language-dependent weight vector of equal dimension to $\overline{\lambda}$, with each component representing the contribution of its individual $n$-gram probability to the overall language score. The spoken document vector in (7) is high dimensional in nature where high order $n$-gram patterns are included. This motivates discriminative feature extraction and selection.

Term weighting [30] is widely used to render the value of the attribute in a document vector by taking into account the frequency of occurrence of each attribute. It is interesting to note that attribute patterns which occur often in a few documents but not as often in others give high indexing power to these documents. On the other hand, patterns which occur very often in all documents possess little indexing power. This desirable property leads to a number of term weighting schemes, such as $tf - idf$ and LSI, that are common for information retrieval [31], natural language call routing [43], and text categorization [44]. We will study methods to extract key features needed for discriminating spoken documents from the statistics of some salient units and their co-occurrences, with details given in Section IV.

Note that the variations [9], [10], [19], [20] of LM backend systems in the prior work used cross-entropy or perplexity based language model scores, that are based on similarity matching in nature, for language classification decision. The VSM can be seen as an attempt to enhance the discrimination power offered by $n$-gram phonotactic information.

### C. VSM-Backend Classifier

With the universal ASM acoustic units in place, any spoken utterance can now be tokenized with a set of "key terms" so that their patterns and statistics can now be used to discriminate individual spoken documents. The given collection of spoken documents, also known as document vectors, in the training set from a particular language forms the same "language category," as illustrated in Fig. 3. LID can be considered the process of classifying a spoken document into some predefined language categories. An unknown testing utterance to be identified can be represented as a query vector, so that LID is performed as in the case of text document classification [35]. We can now utilize any classifier learning techniques, such as support vector machines [33] and artificial neural networks [39], developed in the text categorization community to design language classifiers. Two LID systems with the VSM-backends are schematically shown in Fig. 3. For the PPR frontend, the VSM backend forms a large composite document vector by concatenating $F$ vectors resulting from the individual phone recognizers, as shown in Fig. 3(a); for the UPR frontend, it constructs a single document vector from the single phone recognizer, as shown in Fig. 3(b).

To summarize, with the ASM acoustic units, there will be an LID paradigm of two frontend options for the voice tokenization, PPR, or UPR, and two backend options, LM or VSM. For simplicity, we will only present the formulation of UPR frontend followed by VSM backend, or UPR-VSM, in great detail in this paper. This formulation can be easily extended to the four cross combinations between the two frontends and two backends, as will be discussed in the Section VI-E.

A conceptual block diagram of the overall three-stage procedure for training such a UPR-VSM system is illustrated in Fig. 4. In the upper panel of 4(a), the set of universal ASMs is created using the available training utterances from all languages. These ASMs are then used to decode the spoken utterances in each language in the voice tokenization step, and to convert them into a collection of spoken document vectors, which is the feature extraction step, as shown in the middle panel [4(b)]. Finally, this collection of language-labeled spoken document vectors is used to train LID classifiers as shown in the bottom panel [4(c)]. The same procedure in Fig. 4(b) is also used in run-time testing to extract spoken query vectors.

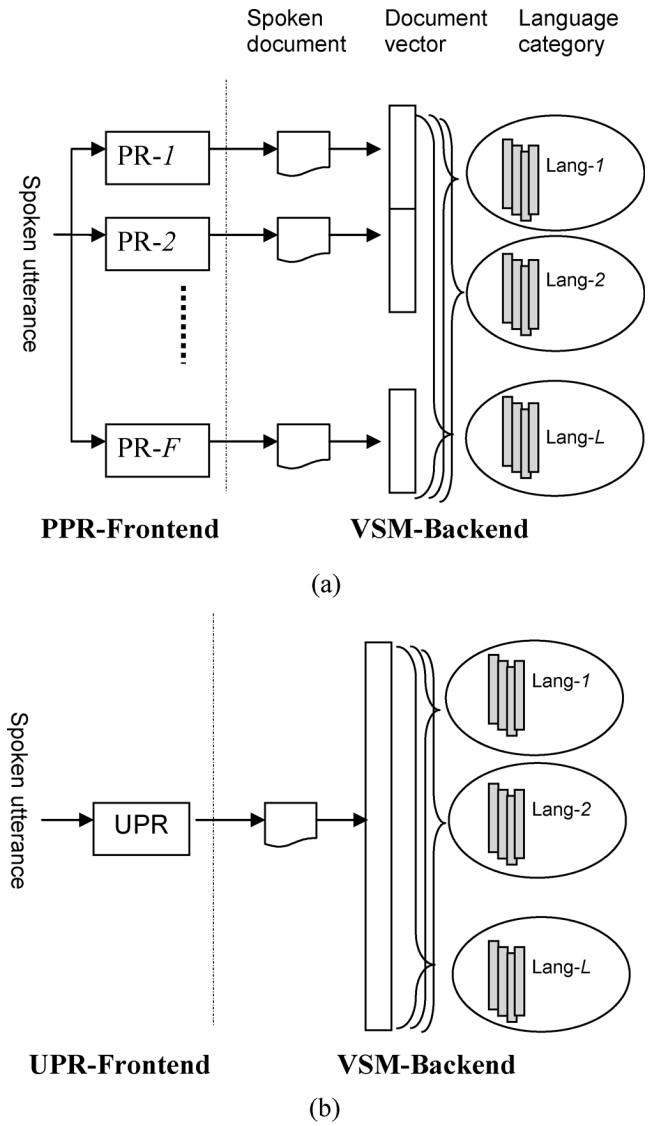



Fig. 3. Block diagram of LID systems with VSM backend. (a) PPR frontend followed by VSM backend. (b) UPR frontend followed by VSM backend.

### D. Databases for LID Experiments

We follow the experiment setup in the NIST LRE tasks.[1] The tasks were intended to establish a baseline of performance capability for language recognition of conversational telephone speech. The evaluation is carried out on recorded telephony speech of 12 languages: Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, and Vietnamese for 1996 and 2003 LRE.

Training sets for building models came from three corpora, namely: 1) the 3-language IIR-LID[2] database [45] with English, Mandarin, and Korean; 2) the six-language OGI-TS (Multi-language Telephone Speech[3]) database with English, German, Hindi, Japanese, Mandarin, and Spanish; and 3) the 12-language LDC *CallFriend*[4] database. The IIR-LID and OGI-TS databases were only used for bootstrapping the acoustic models of an initial set of phones. Both IIR-LID and OGI-TS are telephone speech with phonetic transcriptions. In addition, the *CallFriend* database was used for full-fledged ASM acoustic modeling, backend language modeling, and classifier design. It contains telephone conversations of the same 12 languages that are in the 1996 and 2003 NIST LRE tasks, but without phonetic transcriptions. The three databases are independent of each other.

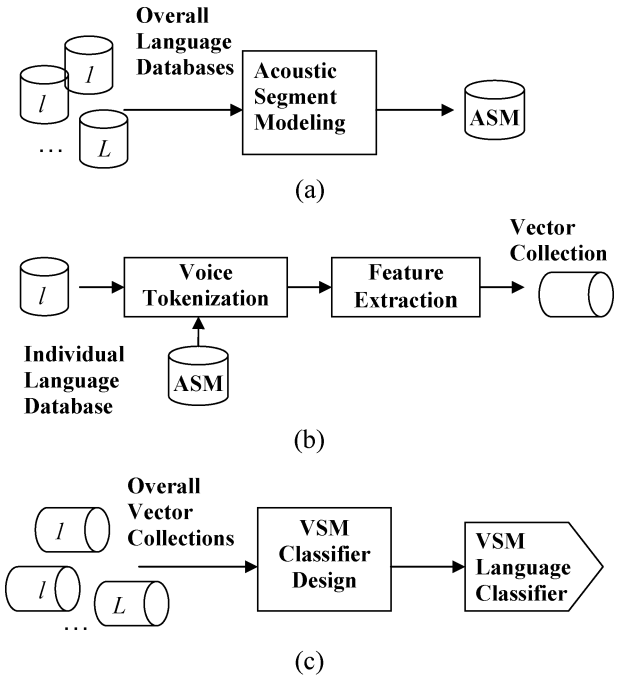



Fig. 4. Overall framework of training UPR-VSM system with ASM acoustic units. (a) Training universal ASMs. (b) Converting utterances into spoken document vectors. (c) Building language classifier.

In the IIR-LID database, each language is given more than 150 h of speech, while in the OGI-TS database, each language is given less than 1 h of speech. In the *CallFriend* database, each of the 12 language databases consists of 40 telephone conversations with each lasting approximately about 30 min, giving a total of about 20 h per language. In language modeling, each conversation in the training set is segmented into overlapping sessions, resulting in about 12 000 sessions for each duration per language. There are three different duration settings, 3, 10, and 30 s. The 1996 LRE evaluation data consist of 1503, 1501 and 1492 sessions for 3, 10, and 30 s, respectively. The 2003 LRE evaluation data consist of 1280 sessions per duration.

## III. Acoustic Segment Modeling

As discussed in Section II, a tokenizer is needed to convert spoken utterances into sequences of fundamental acoustic units specified in an acoustic inventory. We believe that units that are

[1]http://www.nist.gov/speech/tests/index.htm

[2]Institute for Infocomm Research Language Identification corpus.

[3]http://cslu.cse.ogi.edu/corpora/corpCurrent.html

[4]See http://www.ldc.upenn.edu/. The overlap between the *CallFriend* database and the 1996 LRE data has been removed from the training data as suggested in http://www.nist.gov/speech/tests/index.htm for 2003 LRE.

not linked to a particular phonetic definition can be more universal, and therefore conceptually easier to adopt. Such acoustic units are thus highly desirable for universal language characterization, especially for rarely observed languages or languages without orthography and a well-documented phonetic dictionary.

A number of variants have been developed along these lines, which were referred to as language-independent acoustic phone models. Hazen reported using 87 phones from the multilingual OGI-TS corpus [9]. Berkling [19] explored the possibility of finding and using only those phones that best discriminate between language pairs. Berkling [21] and Corredor-Ardoy [20] used phone clustering algorithms to find a common set of phones for languages. However, these systems were constrained to operate only when a phonetically transcribed database is available. On a separate front, a general effort to circumvent the need for phonetic transcription could be traced back to Lee's work [32] in automatic speech recognition, where ASM is constructed in an unsupervised manner. There are also recent reports adopting this concept towards LID [46]. Motivated by the previous two efforts, we will propose an ASM method to establish a universal representation of acoustic units for multiple languages. To the former effort, we own the concept of a language-independent acoustic phone model; to the latter, we credit the unsupervised acoustic modeling technique.

### A. Augmented Phoneme Inventory (API)

Attempts have been made to derive a universal collection of phones to cover all sounds described in an international phonetic inventory, e.g., International Phonetic Alphabet or Worldbet [42]. This is a challenging endeavor in practice. Note that these sounds overlap considerably across languages. One possible approximation is to use a set of phonemes from several languages to form a superset, named augmented phoneme inventory (API) here. This idea has been explored in one way or another in the prior work [9], [19]–[21]. A good inventory needs to phonetically cover as many targeted languages as possible. This method can be effective when phonemes from all the targeted languages form a closed set as studied by Hazen in [9]. Human perceptual experiments have also shown a similar effect where listeners' LID performance improved by increasing their exposure to each language [27].

This API-based tokenization was recently explored [41] by using a set of all 124 phones and four noise units from English, Korean, and Mandarin, and extrapolating them to the other nine languages in the NIST LRE tasks. This set of 128 units is referred to as API-I, which is a proprietary phone set defined for the IIR-LID database. Many preliminary LID experiments were conducted using the IIR-LID database and the API-I phone set. For example, we have explored an API-based approach to universal language characterization [41], and a text categorization approach to LID [45], which formed the basis for the vector based feature extraction to be discussed in the next section. To expand the acoustic and phonetic coverage, we further used another larger set of APIs with 258 phones, from the six languages defined by the OGI-TS database. These six languages all appear in the NIST LRE tasks. This set will be referred to as API-II. A detailed breakdown of how the two phone sets were formed with phone set counts for each language is listed in Table I.

TABLE I
LANGUAGES AND PHONE SETS OF API-I AND -II

| API-I | Count | API-II | Count |
|---|---|---|---|
| English | 44 | English | 48 |
| Mandarin | 43 | Mandarin | 39 |
| Korean | 37 | German | 52 |
| General | 4 | Hindi | 51 |
| | | Japanese | 32 |
| | | Spanish | 36 |
| Total | 128 | Total | 258 |

### B. Acoustic Segment Model (ASM)

The above phone-based language characterization suffers from two major shortcomings. First, a combined phone set from a limited set of multiple languages cannot easily be extended to cover new and rarely seen languages. Second, a collection of transcribed speech data is needed to train the acoustic models for each language. A mismatch may exist between the so trained acoustic models and the intended test environment. To alleviate these difficulties, a data-driven method that does not rely on phonetically transcribed data is preferred. This can be accomplished by constructing consistent ASMs [32] intended to cover the entire sound space of all spoken languages in an unsupervised manner.

Just like in any other hidden Markov modeling, the initialization of ASM is a critical factor to the success of ASM. Note that the unsupervised, data-driven procedure to obtain ASMs may result in unnecessarily many small segments because of a lack of phonetic or prosodic constraints (e.g., number of segments in a word and duration of an ASM) imposed during segmentation. This is especially severe in the case of segmenting a huge collection of speech utterances given by a large population of speakers from different language backgrounds. The API approach uses phonetically defined units in the sound inventory. It has the advantage of having phonetic constraints in the segmentation process. By using API to bootstrap ASM, we effectively incorporate some phonetic knowledge about a few languages in the initialization to guide the ASM training process as follows.

Step 1) Carefully select a few languages, typically with large amounts of labeled data, and train language-specific phone models. Choose a set of $J$ models for bootstrapping.
Step 2) Use these $J$ models to decode all training utterances in the training corpora. Assume the recognized sequences are "true" labels.
Step 3) Force-align and segment all utterances in the training corpora, using the available set of labels and HMMs.
Step 4) Group all segments corresponding to a specific label into a class. Use these segments to retrain an HMM.
Step 5) Repeat Steps 2–4 several times until convergence.

In this procedure, we jointly optimize the $J$ models as well as the segmentation of all utterances. This is equivalent to the commonly adopted segmental ML and $k$-means HMM training algorithm [22] through iterative optimization of segmentation and maximization. We found that API-bootstrapped ASM was more stable than the randomly initialized ASM, but there are

also other ways of initializing ASM, and adding language-independent phonetic or prosodic constraints. The API-bootstrapped ASM outperforms API by a big margin in 1996 NIST LRE task as reported in [41].

With an established acoustic inventory obtained from API or ASM methods, we are now able to tokenize any given speech utterance into a token sequence $\hat{T}$, in a form similar to a text-like document, as shown in (5). Note that ASMs are trained in a self-organized manner. We may not be able to establish a phonetic lexicon using ASMs and translate an ASM sequence into words. However, as far as LID is concerned, we are more interested in a consistent tokenization than the underlying lexical characterization of a spoken utterance. The self-organizing ASM modeling approach offers the key property that we no longer require the training speech data to be directly or indirectly phonetically transcribed.

Comparing the API and ASM methods, we note that the API method comes with more linguistic/phonetic grounding while the ASM method is more acoustically oriented. Instead of using a bottom-up approach to derive purely acoustically oriented units in an unsupervised manner, we use API to bootstrap the units.

The main difference between API and ASM is the relaxation of phone transcription for segmentation. In API, we train the phone models according to manually transcribed phone labels while in ASM, the segmentation is done in iterations using automatic recognition results. In this way, ASM gives us two advantages. 1) It allows us to adjust a set of API phones from a small number of selected languages towards a larger set of targeted languages. 2) The ASM can be trained on the similar acoustic data as is used for the LID task, thus potentially minimizing mismatch between the test data and the API that is trained on a prior set of phonetically transcribed speech.

## IV. Extraction of Salient Language Features

After all the spoken utterances are tokenized, we are now ready to represent a spoken document or a spoken query by a vector whose dimensionality is equal to the size of the total number of useful features, including the statistics of the units and their co-occurrences. For example, in the case of a moderate set of size $J = 128$ ASM units, the total dimension will reach a total of $M = J + J \times J = 16\,512$ features with $J$ unigrams and $J \times J$ bigrams. It is precisely with the usage of such high-dimensional vectors that we expect the discrimination capability of the feature vector, like the one in (7), to improve LID performance even when only phonotactic features are used.

Since high-performance language-dependent acoustic and language models of phones are not required for tokenization, we no longer need a large amount of training speech samples from each spoken language. Instead, we only need a reasonably sized language-dependent training set as spoken documents to obtain a collection of spoken document vectors to train a language classifier for each language. When trigrams with 128 ASM units are incorporated, the vector dimensionality increases to over two millions, which is beyond the ability of current technology to handle in practice. We will study the balance between the acoustic resolution, i.e., the number of units $J$ needed to model the universal sound space for all languages, and the language resolution, i.e., the dimensionality of the language feature vector $M$ needed to provide an adequate discriminative power for language identification. Other salient features relevant to human perception of language cues can also be investigated and incorporated in future work.

### A. Bag-of-Sounds Characterization of Spoken Documents

Many studies on vector-based document representation are available in the information retrieval and text categorization literature [31], [43]–[45], [47]. In this paper, we focus on the language feature characterization that is used to discriminate between languages. Intuitively, sounds are heavily shared across different spoken languages due to the common human speech production mechanism. The acoustic unit as proposed in Section III allows us to move away from the conventional lexical descriptions of spoken languages. To account for the sequential, acoustic-phonotactic constraints, we introduce the concept of the acoustic word (AW). An AW is typically smaller than a lexical word, and is composed of acoustic letters, such as the set of acoustic segment units, in the form of $n$-gram. By exploiting the statistics of AWs, we can improve the discriminative power of the feature vectors by incorporating $n$-grams of different orders.

The sequential, acoustic-phonotactic constraints can be typically described by the ASM $n$-grams, which represent short-term statistics, such as lexical constraints. Suppose that we have a token sequence, $t_1, t_2, t_3, t_4$. We derive the unigram statistics from the token sequence itself. We derive the bigram statistics from $t_1(t_2)\ t_2(t_3)\ t_3(t_4)\ t_4(\#)$ where the acoustic vocabulary is expanded over the token's right context. Similarly, we derive the trigram statistics from the $t_1(\#, t_2)\ t_2(t_1, t_3)\ t_3(t_2, t_4)\ t_4(t_3, \#)$ to account for left and right contexts. The # sign is a place holder for free context. In the interest of manageability, we use up to only token trigrams. In this way, for an acoustic vocabulary of $J$ tokens, we have potentially $J \times J$ bigram and $J \times J \times J$ trigram AWs to form a vocabulary of $M = J + J \times J + J \times J \times J$ AWs.

The aforementioned *bag-of-sounds* concept [13] is analogous to the *bag-of-words* paradigm originally formulated in the context of information retrieval and text categorization. In human languages, some words invariably occur more frequently than others. One of the most common ways to express this notion is known as Zipf's Law [48], [49] which states that there is always a set of words which dominates most of the other words of a language in terms of their frequency of use. This is valid for spoken words as well.

A *bag-of-sounds* vector captures document-level statistics of co-occurrences of AWs. We describe a spoken document as a count vector, $c = \{c_1, c_2, \ldots, c_M\}^T$, of AWs, where each element represents the statistics of an AW and takes the AW vocabulary size $M$ as its dimension size. It is possible to explore the relations and higher-order statistics among the diverse AWs through vector space techniques. Applying Zipf's Law, some AWs can be seen as being more informative and discriminative than others. Like in information retrieval, we can compile a list of stop AWs which do not render much discriminative information across spoken documents. By increasing the number of stop AWs, we effectively reduce the vector dimension and computation cost.

### B. Latent Semantic Indexing

Given an acoustic word inventory $W = \{W_1, W_2, \ldots, W_M\}$ with $M$ acoustic word terms, the content of the overall training set of $D$ documents can be represented by a term-document matrix $H = (d_1, d_2, \cdots, d_D)$, where each component is a quantity describing the statistic of an acoustic term that occurs in a document. For the $j$th document vector $d_j$, an LSI representation can be computed as in [30], with the $i$th component defined as

$$d_{ji} = (1 - \varepsilon_i) \cdot c_{ji}/n_j \tag{9}$$

where $c_{ji}$ is the number of times the term $W_i$ occurred in the $j$th document, $n_j$ is the total number of all the acoustic terms that appeared in the $j$th document, and $\varepsilon_i$ is a normalized entropy for $W_i$ in the training set that is further defined as

$$\varepsilon_i = -\frac{1}{\log D} \sum_{j=1}^{D} \frac{c_{ji}}{t_i} \log \frac{c_{ji}}{t_i} \tag{10}$$

with $t_i = \sum_j c_{ji}$ denoting the total count of $W_i$ in the training set.

## V. Vector-Based Language Classifier Design

After representing a spoken document as a vector of statistics of AWs, LID becomes a vector-based classification problem. Many classifier designs exist in the machine learning literature for high-dimension vector classification [33]. Studies also reveal that dimensionality reduction is effective in improving expressiveness of input vectors. In this study, we will use two conventional techniques, namely SVM and ANN, to train classifiers. Other discriminative classifier learning, such as maximal figure-of-merit (MFoM) learning [44], was studied elsewhere [45].

The SVM is optimized on a structural risk minimization principle [34]. Because of its distribution-free property, it is suitable for designing vector-based classifiers. An alternative is to reduce the dimensionality of the document vectors and then feed the vectors into a multilayer perceptron (MLP) neural network classifier [39]. Using error back-propagation [37], a single neural network can be trained to perform LID. We now describe the two classifiers in more detail.

### A. SVM Classifier

SVM is a classifier of natural choice because the feature vectors are high dimensional, sparse in nature, and do not follow a specific distribution. In this study we consider a two-class SVM, i.e., an input document vector $d$, is labeled as $x_+$ or $x_-$ depending on whether the input belongs to the desired language category. The SVM trains a classifier of the form $f(d) = a^T\psi(d) + b$, described by a weight vector $a$, an offset $b$, and a kernel function $\psi(.)$. Learning is posed as an optimization problem with the goal of maximizing the margin, i.e., the distance between the separating hyperplane, $a^T\psi(d) + b = 0$, and the nearest training vectors, or if $f(d) > 0$, then $d \in x_+$, and if $f(d) \leq 0$, then $d \in x_-$. An extension of this formulation also allows for a wider margin at the cost of misclassifying some of the training examples. We used the $\mathrm{SVM}^{\mathrm{light}}$ V6.01 program[5] to train the SVM models. This program allows us to explore both linear and nonlinear SVM kernels. In this study, we work with a linear kernel SVM, where $\psi(d) = d$, and $f(d)$ can be seen as a realization of (8). Other forms of kernels can also be used.

Note that SVM is a two-class classifier. For $L$ classes, we build $L \times (L-1)/2$ two-class classifiers. A spoken document of unknown class goes through $L \times (L-1)/2$ two-class classification trials. For language identification, the class that gains most of the winning votes represents the identified language. We report the error rate (ER%) as the number of misclassifications over the number of total trials.

For language verification, we formulate the decision function using Bayes decision theory. Using the SVM outputs to form a vector $d'$ of $L \times (L-1)/2$ dimensions to represent the input document vector $d$, not only do we effectively reduce the dimension from a large $M$ to a small $L \times (L-1)/2$, but we also represent the spoken document in a discriminative space of language pairs. Two Gaussian mixture models (GMMs) are built for each language using the vectors $d'$, $\lambda^{x+}$ for the desired language and $\lambda^{x-}$ for all its competing languages. By applying Bayes rule, we decide $\lambda^{x+}$ if $P(\lambda^{x+}|\hat{T}) > P(\lambda^{x-}|\hat{T})$ or $P(\hat{T}|\lambda^{x+})P(\lambda^{x+}) > P(\hat{T}|\lambda^{x-})P(\lambda^{x-})$. Assuming no prior knowledge about $\hat{T}$, that is, $P(\lambda^{x+}) = P(\lambda^{x-})$, we decide $\lambda^{x+}$ as true if $P(\hat{T}|\lambda^{x+}) > P(\hat{T}|\lambda^{x-})$, and false otherwise. We will report language verification results in terms of equal error rate (EER%).

### B. Artificial Neural Network Classifier

Artificial neural networks are capable of learning complex mappings between inputs and outputs, and are particularly useful when the underlying statistics of the considered task are not well understood. We will explore the use of singular vector decomposition (SVD) for dimensionality reduction of document vectors in the experiments. An MLP typically has three layers of nodes. The input layer takes feature vectors as inputs. Each node in the output layer corresponds to a language. Therefore, we have as many nodes in the first layer as the dimensionality in the SVD-reduced space. The nonlinear functions at the hidden and output nodes are sigmoid. For an $L$-class LID task, we have $L$ nodes in the output layer. The number of nodes in the hidden layer is chosen in an empirical manner to yield a good compromise between generalization and regression of the training data. During training, the output node target corresponding to the language of the input vector is set to one, and the other output node targets are set to zero. The training can be done by an error back-propagation algorithm. We use NETLAB [50] for all ANN experiments. During classification, the $l$th output of the MLP is an estimate of $P(\lambda_l^{\mathrm{LM}}|\hat{T})$, which can be used for the identification and verification decision. Since we do not assume prior knowledge about $\hat{T}$, by applying Bayes rule, (6) can now be realized as follows:

$$\hat{l} = \arg\max_l \left[\log P\left(\hat{T}|\lambda_l^{\mathrm{LM}}\right)\right] = \arg\max_l \left[\log P\left(\lambda_l^{\mathrm{LM}}|\hat{T}\right)\right]. \tag{11}$$

[5] http://svmlight.joachims.org/.

TABLE II
ASM ACOUSTIC RESOLUTION (ER% ON 30-s/1996LRE)

| | 8-mix | 16-mix | 32-mix |
|---|---|---|---|
| ER (%) | 16.8 | 15.9 | 13.9 |

## VI. EXPERIMENTAL RESULTS AND DISCUSSION

We used the three corpora, IIR-LID, OGI-TS, and *CallFriend*, for training, and the 1996 and 2003 (12-language) NIST LRE tasks for testing. In the interest of brevity, in Section VI-A–VI-D, we only report the results in error rate (ER%) for the primary subset of 30-s segments in the 1996 LRE task. To establish system performance comparison, in Section VI-E and VI-F, we produce a comprehensive benchmarking on the 1996 and 2003 LRE tasks using the EER% measure.

Three key topics are addressed by conducting a systematic series of experiments. For the topic of training ASMs for tokenization, we are interested in two issues, namely acoustic resolution in terms of the required model detail for each ASM; and acoustic coverage in terms of the number of units in the acoustic inventory needed to model all languages with good accuracy. For the topic of vector-based feature extraction, we studied the complexity and discriminative power of the ASM-derived feature vectors, including feature selection and vector dimensionality reduction. The LID performance as a function of the training set sizes will also be studied. Finally, to appreciate the contributions of different frontends and backends, we construct and test four combinative systems using PPR, UPR frontends, and LM, VSM backends. We also compare the overall performance of the proposed ASM approach with the state-ofthe-art systems in the recent literature.

### A. Effect of ASM Acoustic Resolution

When modeling an acoustic space with three-state HMMs, the model size is dictated by two major settings, the number of models and the number of components in each state. We will discuss the number of components here and the number of models in Section VI-B.

The number of components in an HMM state can be seen as the acoustic resolution that is required to characterize the ASM acoustic space. Now we take a set of 9000 AWs derived from the resulting 128-ASM to be discussed in Section VI-C for a study. Using the 128-ASM units described in Section III-B, we built three sets of models with state mixture Gaussian densities of 8, 16, and 32 mixture components each. The LID error rates are listed in Table II. As expected, increasing acoustic resolution does translate into better LID performance. We chose 32-mix state for rest of the experiments.

### B. Effect of ASM Acoustic Coverage

The number of ASM units reflects the acoustic coverage in terms of characterizing the sound space of all intended spoken languages. We further investigate the effect of the number of ASM units on LID performance. We started with API-II, a 258 phone set from six languages. After being trained in a supervised manner on the OGI-TS database, the API-II phone set was used to bootstrap a set of ASMs, referred to as 258-ASM. We also trained an API-I set of 128 phones from three languages on the phonetically transcribed IIR-LID database, then used the API-I models to bootstrap a new set of ASMs, referred to as 128-ASM. The 258- and 128-ASM were further trained in an unsupervised manner on the *CallFriend* database, as described in Section III-B. To further reduce the number of ASM units, we clustered the 128-ASM models into 64-ASM and 32-ASM according to some acoustic similarity [20].

TABLE III
ACOUSTIC AND PHONOTACTIC COVERAGE (ER% ON 30-s/1996LRE)

| | 32-ASM | 64-ASM | 128-ASM | 259-ASM |
|---|---|---|---|---|
| Unigram | 40.1 | 26.7 | 22.3 | 20.9 |
| Bigram | 32.6 | 18.6 | 13.9 | 13.4 |
| Trigram | 27.9 | NA | NA | NA |

The error rates for all four systems are listed in Table III. The results in the row labeled "Bigram" used feature vectors of dimension 1056 ($J = 32$), 4160 ($J = 64$), 16 512 ($J = 128$), and 66 822 ($J = 258$) for "32-ASM," "64-ASM," "128-ASM," and "258-ASM," respectively. By reducing the acoustic vocabulary size from 258 to 32, we see a big dimension reduction from 66 822 to 1056. It is not surprising to see that the error rate increased drastically from 13.4% to 32.6%, while the ASM coverage is reduced by as much as 87.6% from 258- to 32-ASM. This is consistent with our intuition that a reasonable number of ASM units are needed to cover the sound variations in all spoken languages. It also shows that these reduced-dimension feature vectors greatly impaired the discriminative power of VSM-based LID systems.

To look into this property more closely, we listed results obtained with unigrams, bigrams, and trigrams. Comparing the rows labeled "Unigram" and "Bigram," we clearly see that low-dimension language feature vectors alone (dimensions 32, 64, 128, and 258 for the four unigram systems, respectively) are not enough to discriminate the set of 12 spoken languages. Using trigram feature vector for "32-ASM," with a vector dimension of $M = J + J \times J + J \times J \times J = 33\,824$, almost doubles the dimension of the bigram feature vectors for "128-ASM" ($M = 16\,512$). The error rate only improves slightly from 32.6% to 27.9%. At this point, we do not have the corresponding "Trigram" results for the other ASM configurations due to the extremely high dimensionality of these systems. However, we do expect some additional improvement.

### C. Term Selection and Feature Reduction

In our data-driven approach to deriving AWs from acoustic units and their $n$-grams, we observed that to the imprecision of speech recognition, not all the resulting AWs are valid and useful. Hence, it is desirable to remove those noise words to reduce the feature space. One simple solution is to discard words that have very low frequency and words that occur in too few documents to be statistically significant. This method is referred to as count-trimming (CT).

A second way is to use mutual information (MI) to indicate how significantly an AW's presence contributes to the semantic classification of the spoken documents, and then remove those AW items that show little correlation with the classification results [51].

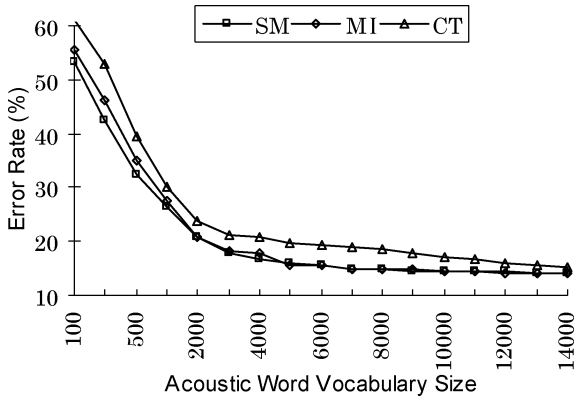



Fig. 5. Selection of AWs evaluated on 30-s/1996LRE.

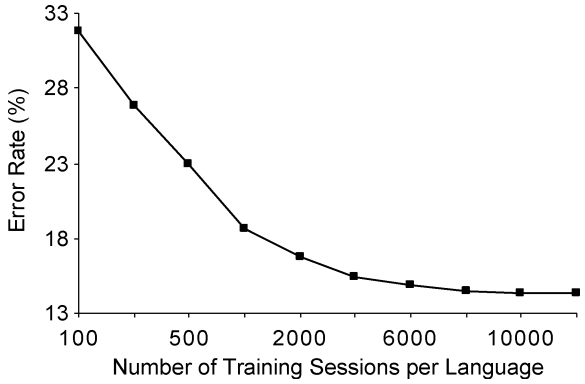



Fig. 6. Training set size evaluated on 30-s/1996LRE.

A third criterion is to consider the weight vector used in the linear discriminant function when training SVMs. It was found that features with higher weight are usually more influential in determining the width of the separation margin (SM) [34]. Therefore, these AWs can be included in the vocabulary.

Now we proceed to study the effects of the aforementioned three feature selection methods with SVM-based classifiers by adopting the 128-ASM set as the acoustic tokens, and including unigrams and bigrams resulting in a vocabulary of $M = 16\,512$ AWs. By removing less influential AWs, we gradually reduced the dimensionality and examined the LID performance results as plotted in Fig. 5. As expected, the simple CT technique did not work as well as the other two discriminative selection methods, with SM working slightly better than MI. Feature selection was rather effective, and we managed to maintain almost the same error rate by reducing vocabulary size from 16 512 to 9000, which amounts to a 45.5% dimensionality reduction.

Another technique that significantly reduces the feature dimension, and is commonly adopted in the IR community, is the use of SVD to decompose the $M \times D$ term-document matrix $H$ obtained in (9) into the product of three matrices

$$H = USV^T \tag{12}$$

where $U$ is a $M \times R$ left singular matrix with rows $u_m$, where $1 \leq m \leq M$, $S$ is a $R \times R$ diagonal matrix of singular values, with $s_1 \geq s_2 \geq \ldots \geq s_R > 0$; and $V$ is a $D \times R$ right singular matrix with rows $v_d$, where $1 \leq d \leq D$. Both the left and the right singular matrices, $U$ and $V$, are column-orthonormal.

If we retain only the top $Q$ singular values in matrix $S$ and zero out the other $(R - Q)$ components, the LSI feature dimension could be effectively reduced to $Q$, which is often much smaller than $R$. By doing so, the three matrices are much smaller in size than those in (12), and this greatly reduces the computational requirements. We can therefore compare spoken documents in this new $Q$ dimensional space, referred to as $Q$-space in the rest of this paper. Any document or query represented by a vector $d$ in the original $M$-dimensional space can now be transformed into a $Q$-dimensional vector. These reduced vectors are then used to train all languages classifiers and perform LID. Since we will explore high-dimension salient feature vectors, this dimension reduction technique is indeed very useful.

Given a test spoken document $\tilde{d}$ which is not part of the training set, we would like to classify it into one of the $L$ languages. Using the left singular matrix $U$, we can construct a document vector $\tilde{v}$ in the $Q$-space, referred to as a *pseudo document vector* [30]

$$\tilde{d} \rightarrow \tilde{v} = \tilde{d}^T U S^{-1}. \tag{13}$$

In a preliminary study, we used the set of 128-ASM models discussed earlier to tokenize all training spoken documents, then set $Q = 90$ for the reduced feature dimension. By training a single ANN with the number of hidden nodes set to be 100, using NETLAB [50], we obtained an LID error rate of 15.4%, which is close to the best error rates obtained in Fig. 5 but with only one hundredth of the features.

### D. Performance and Training Set Sizes

It is well known that the size of the training set often greatly affects the performance of a pattern classification system. We proceed with the 9000 AWs derived from the feature reduction described in Section VI-C to study the effects of training set size. We used the same set of 128-ASM models in Section VI-A. In Fig. 6, we plot the LID error rate as a function of the number of training sessions for SVM-based classifiers. The full corpus includes 12 000 spoken documents, or sessions, for each language. The subset was randomly selected from the full corpus with an equal amount from each language. We observe that when subset size grows beyond 8000 sessions per language, the performance begins to saturate.

In Table IV, we compare the error rates using different training corpus sizes. We also report the number of resulting support vectors in the SVM classifiers. When the number of sessions per language was 1000, there were 2000 training vectors for each language pair. The two-class SVM classifiers derived 1048 support vectors on average. Note that the number of support vectors increased as training corpus grew, but at a much slower rate than that of the training corpus size. When the training corpus size grew by 12 times, the number of support vectors only doubled. This explains the fact that, beyond 8000 sessions, increasing training corpus size does not translate into accuracy improvements.

In text categorization, we use the SVM classifier to identify the topic of a text document, where topics can be defined at different levels of semantic granularity and characterized by the

TABLE IV
TRAINING SET SIZE EVALUATED ON 30-s/1996LRE (ER%)

| *# sessions/language* | 1,000 | 2,000 | 6,000 | 12,000 |
|---|---|---|---|---|
| *SVM classifier (# support vectors)* | 18.2 (1048) | 16.2 (1457) | 14.4 (1951) | 13.9 (2142) |

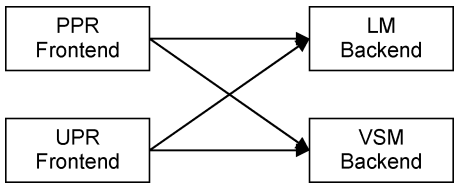



Fig. 7. Block diagram of four combinations of frontend and backend.

co-occurrences of lexical words in a document. Similarly, we now use SVM classifier to identify the category of a spoken document. The top-level semantic category in LID is the language identity itself. Note that $n$-gram acoustic words (bigram and trigram) are smaller units than lexical words. They represent phonotactic constraints of a language instead of the lexical constraints of a topic. In a given spoken language, the real topics of spoken documents are considered variations to the vector representation, analogous to the intratopic variations in text categorization. The number of support vectors in Table IV indicates how well the SVM classifier partitions the language vector space. The slow increment in number of support vectors indicates that the intralanguage variations are captured well as the training set size grows towards 12 000. Nonetheless the intralanguage variations still present a challenge to LID in the same way as intratopic variations do to text categorization.

### *E. Frontend and Backend*

We have discussed two different frontends, PPR and UPR, and two different backends, LM and VSM. To gain further insight into the behavior of each of the frontends and backends, it is desirable to investigate the performance of each of the four combinative systems as given in Fig. 7, namely PPR-LM, PPR-VSM, UPR-LM, and UPR-VSM, where the PPR/UPR frontends are built on a set of universal ASMs.

Without loss of generality, we deployed the same 258-ASM in two different settings. First, the 258 ASMs were arranged in a six-language PPR frontend. They were redistributed according to their API-II definitions into six languages. Second, they were lumped together in a single UPR frontend. The training of 258-ASM was discussed in Section III-B. We used SVM classifier in the VSM backend, in which we trained 512-mixture GMMs for both $\lambda^{x+}$ and $\lambda^{x-}$ of each language, and report the equal error rates.

The UPR-LM system follows the block diagram of language-independent acoustic phone recognition approach [20]. The PPR-LM is implemented as in [10]. The LM backend uses trigram to derive phonotactic scores. The results on the 1996 and 2003 LRE tasks are reported in Tables V and VI, respectively. In Table VII, we also report the execution time of

TABLE V
EER% COMPARISON OF FOUR SYSTEMS ON 1996LRE

| System | 30-second | 10-second | 3-second |
|---|---|---|---|
| PPR-VSM | 2.75 | 8.23 | 21.16 |
| PPR-LM | 2.92 | 8.39 | 18.61 |
| UPR-VSM | 4.87 | 11.18 | 22.38 |
| UPR-LM | 6.78 | 15.90 | 27.20 |

TABLE VI
EER% COMPARISON OF FOUR SYSTEMS ON 2003LRE

| System | 30-second | 10-second | 3-second |
|---|---|---|---|
| PPR-VSM | 4.02 | 10.97 | 21.66 |
| PPR-LM | 4.62 | 11.30 | 21.18 |
| UPR-VSM | 6.81 | 13.75 | 24.44 |
| UPR-LM | 10.81 | 19.95 | 30.48 |

TABLE VII
EXECUTION TIME COMPARISON ON 2003LRE (REAL-TIME-FACTOR OF 30-s TRIALS)

| System | Frontend | Backend | Total |
|---|---|---|---|
| PPR-VSM | 0.7xRT | 0.01xRT | 0.71xRT |
| PPR-LM | 0.7xRT | 0.03xRT | 0.73xRT |
| UPR-VSM | 0.3xRT | 0.001xRT | 0.301xRT |
| UPR-LM | 0.3xRT | 0.02xRT | 0.32xRT |

2003 LRE task in terms of real-time-factor (xRT) on an Intel Xeon 2.80-GHz CPU.

Before looking into the results, let us study the combinative effect of the frontends and backends. In the combinative systems, there are two unique frontend settings, PPR and UPR. The PPR converts an input spoken utterance into six spoken documents using the parallel frontend, while the UPR converts an input into a single document. However, there are four unique LM and VSM backend settings. The LM in the PPR-LM and that in the UPR-LM are different; the former has $6 \times 12$ $n$-gram language models while the latter only has 12 language models. In other words, the former LM classifier is more complex, with a larger number of parameters, than the latter. The VSM in the PPR-VSM and the VSM in the UPR-VSM present different complexity as well. The former VSM processes vectors of 11 708 $(= 48^2+39^2+52^2+51^2+32^2+36^2+48+39+52+51+32+36)$ dimensions while the latter processes those of 66 822 $(= 258^2 + 258)$ dimensions, as formulated in Section IV. The vectors in PPR-VSM and UPR-VSM can be illustrated as in Fig. 8.

Although the dimensionality of V-PPR is lower than that of V-UPR, V-PPR is approximately six times as dense as V-UPR, resulting in more complex support vector machines partitions (SVM) [34]. In other words, the VSM classifier in the PPR-VSM is more complex than that in UPR-VSM. In

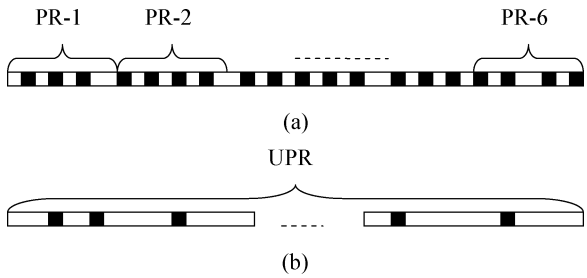



Fig. 8. Two different spoken document vectors in PPR-VSM and UPR-VSM. (a) A 11 708-dimensional vector from six PPRs (V-PPR). (b) A 66 822-dimensional vector from the UPR (V-UPR).

terms of overall classifier backend complexity, we rank the four systems from high to low as follows: PPR-VSM, PPR-LM or UPR-VSM, and UPR-LM.

Summarizing the results in the two LRE tasks, we have the following findings.

1) VSM backend demonstrates a clear advantage over LM backend for the 30-s trials, while LM works better for the 3-s trials in general. This can be easily explained by the fact that VSM models are designed to capture higher order phonotactics. As a result, VSM favors longer utterances which provide richer long span phonotactic information.
2) The system performance highly correlates with the complexity of system architectures. This can be found consistently in Tables V and VI, where the PPR-VSM presents the best result with an EER of 2.75%, 4.02% in the 30-s 1996 and 2003 LRE tasks respectively, followed by PPR-LM, UPR-VSM, and UPR-LM. Note that we can increase the system complexity by including more phone recognizers in PPR. We expect more phone recognizers to further improve the PPR-VSM system performance.
3) Although PPR-LM outperforms UPR-VSM in general, UPR frontend shows an advantage of computational efficiency in run-time operation over PPR frontend. In Table VII, we find that systems with UPR frontend run almost 60% faster than those with PPR frontend.

As a general remark, ASM-based acoustic modeling not only offers an effective unsupervised training procedure, hence low development cost, but also offers an efficient run-time operation as in the case of UPR frontend. More importantly, it delivers outstanding system performance. VSM is the choice of backend when longer utterances are available, with PPR-VSM delivering the best result in the comprehensive benchmarking for the 30-s test condition.

### *F. Overall Performance Comparison*

LID technology has gone through many years of evolution. Many results have been published in the literature on the 1996 and 2003 NIST LRE tasks, which provides good benchmarks for new technology development. Here, we summarize some recently reported results.

For brevity, we only compare the results in the 30-s tests, which represent the primary condition of interest in the LRE tasks. In Table VIII, Systems 1–3 are trained and tested on the same databases. Therefore, the results can be directly compared.

TABLE VIII
EER% BENCHMARK ON 30-s 1996/2003LRE

| | System | 1996 LRE | 2003 LRE |
|---|---|---|---|
| 1 | PPR-VSM | 2.75 | 4.02 |
| 2 | PPR-LM | 2.92 | 4.62 |
| 3 | UPR-VSM | 4.87 | 6.81 |
| 4 | Phone Lattice [17] | 3.20 | 4.00 |
| 5 | NN Fusion [17] | NA | 2.70 |
| 6 | Parallel PRLM [15] | 5.60 | 6.60[6] |

We extract the results from Tables V and VI to compare with the recently reported results (Systems 4–6) [15], [17]. Table VIII shows that the performance of PPR-VSM system represents one of the best reported results on the 1996 and 2003 LRE tasks.

Ma [41] reported that the API-bootstrapped ASM outperforms API phone models in LID task. This paper extends our previous work through a comprehensive benchmarking which further ascertains our finding and validates the effectiveness of the proposed VSM solution. The systems reported in Tables V and VI contributed to the ensemble classifier that participated in the 2005 NIST LRE representing the Institute of Infocomm Research (IIR) site [52].

The proposed VSM-based language classifier only compares phonotactic statistics from spoken documents. We have not explored the use of the acoustic scores resulting from the tokenization process. It was reported that there is a clear win in combining information about the acoustic scores along with the phonotactic statistics [15], [18], [20]. Furthermore, fusion of phonotactic statistics at different levels of resolutions also improves overall performance [53]. We have good reason to expect that fusion among our four combinative systems, or between our systems and other existing methods, which include acoustic score classifier [18] and GMM tokenizer [25], will bring us further improvements.

## VII. CONCLUSION

We have proposed a vector space modeling approach to spoken language identification. The main contributions of this paper are in two areas: 1) an ASM technique for PPR/UPR frontend design and 2) a VSM strategy towards discriminative classifier backend design. Next, let us summarize our findings.

Based on a universal set of ASMs, all the utterances can be tokenized into sequences of these acoustic units in the form of text documents, with the acoustic unit serving as acoustic alphabets. These letters can then be grouped into acoustic words based on their co-occurrences, similar to defining words based on letters. We can now adopt a vector based representation of spoken document, similar to the term vector representation of text documents. Techniques in information retrieval, such as latent semantic indexing, can then be readily used to perform feature extraction for document indexing and retrieval. By grouping spoken documents of the same spoken language into a language category, LID can now be formulated as a text categorization problem. Thus, we can take advantage of many methods in feature extraction, feature reduction, and classifier design methods

to improve the capability and performance of language identification and verification. Using a conventional SVM classifier design, we achieved an EER of 2.75% and 4.02% in 30-s 1996 and 2003 LRE tasks, respectively from the PPR-VSM system, which represents one of the best reported results as a single LID classifier. This can be credited to the enhanced discriminatory ability of the VSM backend. We also have demonstrated the effectiveness of UPR frontend in system development and at run-time operation.

By exploring the *bag-of-sounds* spoken document vectors using bigram statistics of "spoken letters," we found that one of the advantages of the VSM method is that it can represent a document with heterogeneous attributes (a mix of unigram, bigram, trigram, etc). Inspired by the feature reduction results, we believe that the *bag-of-sounds* vector can be extended to accommodate trigram statistics and acoustic features as well.

We have successfully treated LID as a text categorization application with the topic category being the language identity itself. The VSM method can be extended to other spoken document classification tasks as well, for example, in multilingual spoken document categorization by topic. We are also interested in exploring language-specific features, such as syllabic and tonal properties. It is quite straightforward to incorporate specific salient features and examine their benefits. Furthermore, some high-frequency, language-specific words can also be converted into acoustic words and included into the acoustic word vocabulary, in order to increase the indexing power of these words for their corresponding languages.

Classifier design is another key topic for performance improvement. We have only used conventional techniques to evaluate the proposed ASM approach so far. It is clear that other discriminative classifier designs, such as maximal figure-of-merit learning, has shown improvement in accuracy and robustness for learning topic classifiers with very few training samples. We believe this will become a useful tool for training language classifiers in cases where only a small number of samples are available, or for some rarely observed languages.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Sheng Gao of the Institute of Infocomm Research (IIR) for his discussions. They would also like to thank Dr. A. Martin of NIST for providing the 1996 and 2003 LRE schemes, and Y. Chen and Y.-L. Chow of IIR for their efforts in improving this paper. The ANN results discussed at the end of Section VI-C and the results in Section VI-E were graciously made available by J. Li of the Georgia Institute of Technology and R. Tong of IIR, respectively.

## REFERENCES

[1] V. W. Zue and J. R. Glass, "Conversational interfaces: advances and challenges," *Proc. IEEE*, vol. 88, no. 8, pp. 1166–1180, Aug. 2000.

[2] A. Waibel, P. Geutner, L. M. Tomokiyo, T. Schultz, and M. Woszczyna, "Multilinguality in speech and spoken language systems," *Proc. IEEE*, vol. 88, no. 8, pp. 1181–1190, Aug. 2000.

[3] B. Ma, C. Guan, H. Li, and C.-H. Lee, "Multilingual speech recognition with language identification," in *Proc. ICSLP*, 2002, pp. 505–508.

[4] P. Dai, U. Iurgel, and G. Rigoll, "A novel feature combination approach for spoken document classification with support vector machines," in *Proc. Multimedia Information Retrieval Workshop*, 2003, pp. 1–5.

[5] K. Kirchhoff, S. Parandekar, and J. Bilmes, "Mixed memory Markov models for automatic language identification," in *Proc. ICASSP*, 2002, pp. 761–764.

[6] M. Sugiyama, "Automatic language recognition using acoustic features," in *Proc. ICASSP*, 1991, pp. 813–816.

[7] A. G. Adami and H. Hermansky, "Segmentation of speech for speaker and language recognition," in *Proc. Eurospeech*, 2003, pp. 841–844.

[8] M. Adda-Decker, F. Antoine, P. Boula de Mareuil, E Geoffrois, and J.-S. Liénard, "Phonetic knowledge, phonotactics and perceptual validation for automatic language identification," in *Proc. ICPhS*, 2003, pp. 747–750.

[9] T. J. Hazen, "Automatic language identification using a segment-based approach," M.S. thesis, Mass. Inst. Technol., Cambridge, MA, 1993.

[10] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 1, pp. 31–44, Jan. 1996.

[11] D. Matrouf, M. Adda-Decker, L. F. Lamel, and J.-L. Gauvain, "Language identification incorporating lexical information," in *Proc. ICSLP*, 1998, pp. 181–184.

[12] J. L. Gauvain and L. Lamel, "Large-vocabulary continuous speech recognition: advances and applications," *Proc. IEEE*, vol. 88, no. 8, pp. 1181–1200, Aug. 2000.

[13] H. Li and B. Ma, "A phonotactic language model for spoken language identification," in *Proc. ACL*, 2005, pp. 515–522.

[14] S. Parandekar and K. Kirchhoff, "Multi-stream language identification using data-driven dependency selection," in *Proc. ICASSP*, 2003, pp. 28–31.

[15] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, "Acoustic, phonetic and discriminative approaches to automatic language recognition," in *Proc. Eurospeech*, 2003, pp. 1345–1348.

[16] Y. Yan and E. Barnard, "An approach to automatic language identification based on language dependent phone recognition," in *Proc. ICASSP*, 1995, pp. 3511–3514.

[17] J. L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Proc. ICSLP*, 2004, pp. 1215–1218.

[18] P. A. Torres-Carassquilo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller, Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP*, 2002, pp. 89–92.

[19] K. M. Berkling and E. Barnard, "Analysis of phoneme-based features for language identification," in *Proc. ICASSP*, 1994, pp. 289–292.

[20] C. Corredor-Ardoy, J. L. Gauvain, M. Adda-Decker, and L. Lamel, "Language identification with language-independent acoustic models," in *Proc. Eurospeech*, 1997, pp. 55–58.

[21] K. M. Berkling and E. Barnard, "Language identification of six languages based on a common set of broad phonemes," in *Proc. ICSLP*, 1994, pp. 1891–1894.

[22] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[23] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[24] T. Nagarajan and H. A. Murthy, "Language identification using parallel syllable-like unit recognition," in *Proc. ICASSP*, 2004, pp. 401–404.

[25] P. A. Torres-Carrasquillo, D. A. Reynolds, and R. J. Deller, Jr., "Language identification using Gaussian mixture model Tokenization," in *Proc. ICASSP*, 2002, pp. 757–760.

[26] J. Allen, "How do humans process and recognize speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 567–577, Jul. 1994.

[27] Y. K. Muthusamy, N. Jain, and R. A. Cole, "Perceptual benchmarks for automatic language identification," in *Proc. ICASSP*, 1994, pp. 333–336.

[28] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recognition," in *Proc. ICSLP*, 2004, pp. 109–112.

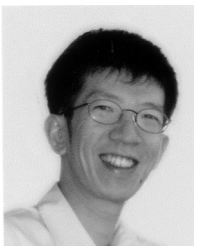
[29] C. E. Shannon, "Prediction and entropy of printed English," *Bell Syst. Tech. J.*, vol. 30, pp. 50–64, 1951.

[30] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proc. IEEE*, vol. 88, no. 8, pp. 1279–1296, Aug. 2000.

[31] G. Salton, *The SMART Retrievl System*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[32] C.-H. Lee, F. K. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proc. ICASSP*, 1988, pp. 501–541.

[33] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.
[34] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[35] T. Joachims, *Learning to Classify Text Using Support Vector Machines*. Norwell, MA: Kluwer, 2002.
[36] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, "Language recognition with support vector machines," in *Proc. ODYS*, 2004, pp. 41–44.
[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagation errors," *Nature*, vol. 323, pp. 533–536, Oct 1986.
[38] R. Lippmann, "An introduction to computing with neural nets," *IEEE Acoust., Speech, Signal Process. Mag.*, vol. 4, no. 2, pp. 4–22, Apr. 1987.
[39] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: MacMillan, 1994.
[40] T. J. Hazen and V. W. Zue, "Recent improvements in an approach to segment-based automatic language identification," in *Proc. ICSLP*, 1994.
[41] B. Ma, H. Li, and C.-H. Lee, "An acoustic segment modeling approach to automatic language identification," in *Proc. Interspeech*, 2005, pp. 2819–2832.
[42] J. L. Hieronymus, ASCII Phonetic Symbols for the World's Languages: Worldbet AT&T Bell Labs, 1994, Tech. Rep..
[43] H. K. J. Kuo and C.-H. Lee, "Discriminative training of natural language call routers," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 1, pp. 24–35, Jan. 2003.
[44] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua, "A MFoM learning approach to robust multiclass multi-label text categorization," in *Proc. ICML*, pp. 42–49.
[45] S. Gao, B. Ma, H. Li, and C.-H. Lee, "A text-categorization approach to spoken language identification," in *Proc. Interspeech*, 2005, pp. 2837–2840.
[46] A. K. V. Sai Jayram, V. Ramasubramanian, and T. V. Sreenivas, "Language identification using parallel sub-word recognition," in *Proc. ICASSP*, 2003, pp. 32–35.
[47] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Comput. Ling.*, vol. 25, no. 3, pp. 361–388, 1999.
[48] G. K. Zipf, *Human Behavior and the Principal of Least Effort, An Introduction to Human Ecology*. Reading, Mass: Addison-Wesley, 1949.
[49] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 28, pp. 11–20, 1972.
[50] I. T. Nabney, *NETLAB: Algorithms for Pattern Recognition*. New York: Springer-Verlag, 2001.
[51] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling, "Feature selection using linear classifier weights: interaction with classification with classification models," in *Proc. SIGIR*, 2004, pp. 234–241.
[52] R. Tong, B. Ma, D. Zhu, H. Li, and E. S. Chng, "Integrating acoustic, prosodic and phonotactic features for spoken language identification," in *Proc. ICASSP*, 2006, pp. 205–208.
[53] B. P. Lim, H. Li, and B. Ma, "Using local and global phonotactic features in Chinese dialect identification," in *Proc. ICASSP*, 2005, pp. 577–580.



**Haizhou Li** (M'91–SM'01) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and electronic engineering from the South China University of Technology (SCUT), Guangzhou, in 1984, 1987, and 1990, respectively.

He was a Research Assistant from 1988 to 1990 at the University of Hong Kong, Hong Kong, China. In 1990, he joined SCUT as an Associate Professor where he became a Full Professor in 1994. From 1994 to 1995, he was a Visiting Professor at CRIN, France. In 1995, he became the Manager of the ASR group at the Apple-ISS Research Centre, where he led the research of Apple's Chinese Dictation Kit for Macintosh. In 1999, he was appointed as the Research Director of Lernout and Hauspie Asia Pacific, where he oversaw the creation of the world's first multimodal speech, pen, and keyboard input solution for Chinese computing. From 2001 to 2003, he was the Vice President of InfoTalk Corporation, Ltd. Since 2003, he has been with the Institute for Infocomm Research (I2R), where he is now the Head of the Speech and Dialogue Processing Laboratory. He is also an Adjunct Associate Professor of the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include automatic speech recognition, speaker recognition, spoken language recognition, and natural language processing.

Dr. Li was a recipient of the National Infocomm Award 2001 and the TEC Innovator's Award 2004 in Singapore. He is now the Vice President of the COLIPS and a member of ACL.

**Bin Ma** (SM'00–SM'06) received the B.Sc degree in computer science from Shandong University, Shandong, China, in 1990, the M.Sc degree in pattern recognition and artificial intelligence from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, in 1993, and the Ph.D degree in computer engineering from the University of Hong Kong, Hong Kong, China, in 2000.

He was a Research Assistant from 1993 to 1996 at the National Laboratory of Pattern Recognition, IACAS, and was involved in research on the Chinese Dictation Machine supported by the National High Technology Plan. In 2000, he joined Lernout and Hauspie Asia Pacific as a Researcher focusing on the speech recognition of multiple Asia languages. From 2001 to 2004, he worked for InfoTalk Corporation, Ltd. and became the Senior Technical Manager engaging in mix-lingual telephony speech recognition system for the Asia–Pacific market, and in embedded speech recognition system on PDAs and hand-phones. Since 2004, he has been a Scientist with Institute for Infocomm Research, Singapore. His current research interests include robust speech recognition, speaker and language recognition, spoken document retrieval, natural language processing, and machine learning.

**Chin-Hui Lee** (F'97) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973, the M.S. degree in engineering and applied science from Yale University, New Haven, CT, in 1977, and the Ph.D. degree in electrical engineering with a minor in statistics from University of Washington, Seattle, in 1981.

After graduation, he joined Verbex Corporation, Bedford, MA, and was involved in research on connected word recognition. In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, CA, where he engaged in research and product development in speech coding, speech synthesis, speech recognition, and signal processing for development of the DSC-2000 Voice Server. Between 1986 and 2001, he was with Bell Laboratories, Murray Hill, NJ, where he became a Distinguished Member of Technical Staff and Director of the Dialogue Systems Research Department. His research interests include multimedia communication, multimedia signal and information processing, speech and speaker recognition, speech and language modeling, spoken dialogue processing, machine learning, biometric authentication, information retrieval, and bioinformatics. His research scope is reflected in a best seller, entitled "Automatic Speech and Speaker Recognition: Advanced Topics," (Kluwer, 1996). From August 2001 to August 2002 he was a Visiting Professor at the School of Computing, National University of Singapore. In September 2002, he joined the faculty of the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. He has published more than 250 papers and 25 patents on the subject of automatic speech and speaker recognition.

Prof. Lee has participated actively in professional societies. He is a member of the IEEE Signal Processing Society, Communication Society, Computer Society, and the European Speech Communication Association. He is also a lifetime member of the Computational Linguistics Society in Taiwan. From 1991 to 1995, he was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. During the same period, he served as a member of the ARPA Spoken Language Coordination Committee. From 1995 to 1998, he was a member of the Speech Processing Technical Committee of the IEEE Signal Processing Society (SPS) and chaired the Speech TC from 1997 to 1998. In 1996, he helped promote the SPS Multimedia Signal Processing Technical Committee, of which he is a founding member. He received the SPS Senior Award in 1994 and the SPS Best Paper Award in 1997 and 1999, respectively. In 1997, he was awarded the prestigious Bell Labs President's Gold Award for his contributions to the Lucent Speech Processing Solutions product. More recently, he was named one of the six Distinguished Lecturers for the year 2000 by the IEEE Signal Processing Society.