Computational Linguistics and Chinese Language Processing
Vol. 11, No. 2, June 2006, pp. 159-182

A Comparative Study of Four Language Identification Systems

Bin Ma* **and Haizhou Li***

Abstract

In this paper, we compare four typical spoken language identification (LID) systems. We introduce a novel acoustic segment modeling approach for the LID system frontend. It is assumed that the overall sound characteristics of all spoken languages can be covered by a universal collection of acoustic segment models (ASMs) without imposing strict phonetic definitions. The ASM models are used to decode spoken utterances into strings of segment units in parallel phone recognition (PPR) and universal phone recognition (UPR) frontends. We also propose a novel approach to LID system backend design, where the statistics of ASMs and their co-occurrences are used to form ASM-derived feature vectors, in a vector space modeling (VSM) approach, as opposed to the traditional language modeling (LM) approach, in order to discriminate between individual spoken languages. Four LID systems are built to evaluate the effects of two different frontends and two different backends. We evaluate the four systems based on the 1996, 2003 and 2005 NIST Language Recognition Evaluation (LRE) tasks. The results show that the proposed ASM-based VSM framework reduces the LID error rate quite significantly when compared with the widely-used parallel PRLM method. Among the four configurations, the PPR-VSM system demonstrates the best performance across all of the tasks.

Keywords: Automatic Language Identification, Acoustic Segment Models, Universal Phone Recognizer, Parallel Phone Recognizers, Vector Space Modeling

1. Introduction

Automatic language identification (LID) is the process of determining the language identity corresponding to a spoken query. It is an important technology in many applications, such as spoken language translation, multilingual speech recognition [Ma *et al.* 2002], and spoken

* Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore, 119613
Phone: (65) 68747866 Fax: (65) 6775 5014
E-mail: {mabin, hli}@i2r.a-star.edu.sg

[Received February 3 2006; Revised April 26, 2006; Accepted May 3, 2006]

document retrieval [Dai *et al.* 2003]. In the past few decades, many statistical approaches to LID have been developed [Kirchhoff *et al.* 2002] [Li and Ma 2005] [Matrouf *et al.* 1998] [Nagarajan and Murthy 2004] [Parandekar and Kirchhoff 2003] [Singer *et al.* 2003] [Torres-Carrasquillo *et al.* 2002] [Yan and Barnard 1995] [Zissman 1996] by exploiting recent advances in the acoustic modeling [Singer *et al.* 2003] [Torres-Carrasquillo *et al.* 2002] of phone units and the language modeling of *n*-grams of these phones [Li and Ma 2005] [Parandekar and Kirchhoff 2003]. Acoustic phone models are used in language-dependent continuous phone recognition to convert speech utterances into sequences of phone symbols in a tokenization process. Then the scores from acoustic models and the scores from language models are combined to obtain a language-specific score for making a final LID decision [Zissman 1996].

Syllable-like units have also been studied [Nagarajan and Murthy 2004]. To further improve the LID performance, other information, such as articulatory and acoustic features [Kirchhoff *et al.* 2002] [Sugiyama 1991], lexical knowledge [Adda-Decker *et al.* 2003] [Ma *et al.* 2002] and prosody [Hazen and Zue 1994], have also been integrated into LID systems. Zissman [1996] experimentally showed that phonetic language models can sometimes be more powerful than MFCC-based Gaussian mixture models (GMMs) [Torres-Carrasquillo *et al.* 2002]. Therefore the fusion of high-level features and good utilization of their statistics are two important research topics for LID.

To make use of high-level features, the LID problem can be taken as consisting of two sub-problems, the tokenization problem and the classification problem. When the tokenization problem is addressed, a fundamental question that arises is whether phone definition is really needed to identify spoken languages. When human beings are constantly exposed to a language without being given any linguistic knowledge, they learn to determine the language's identity by perceiving some of the speech cues in the language. It is also noteworthy that in human perceptual experiments, listeners with multilingual background often perform better than monolingual listeners in identifying unfamiliar languages [Muthusamy *et al.* 1994]. These results motivate us to look for useful speech cues for LID along the same line of a recently proposed automatic speech attribute transcription (ASAT) paradigm for automatic speech recognition [Lee 2004]. When we address the classification problem, we find that the strategies such as feature representation for spoken documents and classifier design principles have direct impacts on LID performance.

In this paper, we adopt the acoustic segment modeling approach to address the tokenization problem. It is assumed that the sound characteristics of all spoken languages can be covered by a set of acoustic units without strict phonetic definitions, which are called acoustic segment models (ASMs) [Lee *et al.* 1998]. They can be used to decode spoken utterances into strings of such units. We also propose a vector space modeling approach (VSM)

to classifier design where the statistics of the units and their co-occurrences corresponding to spoken utterances are used to construct feature vectors.

Hidden Markov modeling (HMM) [Rabiner 1989] is the dominant approach to acoustic modeling. A collection of ASMs is established from the bottom up in an unsupervised manner using HMM, and has been used to construct an acoustic lexicon for isolated word recognition with high accuracy [Lee *et al.* 1998]. In LID research, a large body of prior work in LID has been devoted to the PR-LM framework (the phone-recognition frontend followed by the language model backend) [Zissman 1996] and its variations, where phonetic units are used as acoustic units. This is also referred to as the phonotactic approach. The phonotactic approach has been shown to achieve superior performance in NIST LRE tasks especially when it is fused with acoustic scores [Singer *et al.* 2003]. In this paper, we investigate four LID system configurations cast in a formalism of frontend feature extraction and backend classifier, namely parallel phone recognizer (PPR) and universal phone recognizer (UPR) frontends, and *n*-gram language model (LM) and vector space model (VSM) backends. We show that the ASM-based PPR-VSM system configuration achieves the best performance across 1996, 2003 and 2005 NIST Language Recognition Evaluation tasks.

This paper is organized as follows. In Section 2, we introduce the acoustic segment modeling approach. In Section 3, we discuss LID systems by studying their frontends and backends. In Section 4, we present the experimental results on four front-backend combinations. We draw conclusions in Section 5.

2. Acoustic Segment Modeling

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 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, languages without orthographies, or languages without well-documented phonetic dictionary.

A number of variants have been developed along these lines, which have been referred to as language-independent acoustic phone models. Hazen and Zue [1994] reported using 87 phones from the multilingual OGI-TS corpus. Berkling and Barnard [1994a] explored the possibility of finding and using only those phones that best discriminate between language pairs. Berkling and Barnard [1994b] and Corredor-Ardoy *et al.* [1997] used phone clustering algorithms to find common sets of phones for languages. However, these systems could only operate when a phonetically transcribed database was available. On a separate front, a general effort to circumvent the need for phonetic transcription can be traced back to [Lee *et al.* 1998] on automatic speech recognition, where ASM was constructed in an unsupervised manner.

Some recent studies have applied this concept to LID [Sai Jayram *et al.* 2003]. Motivated by the above efforts, we propose here an ASM method for establishing a universal representation of acoustic units for multiple languages.

2.1 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 (IPA) or Worldbet [Hieronymus 1994]. In practice, this is a challenging endeavor because we need a large collection of labeled speech samples for all languages. Note that these sounds overlap considerably across languages. One possible approximation approach is to use a set of phonemes from several languages to form a superset, called an augmented phoneme inventory (API) here. This idea has been explored in previous works [Berkling and Barnard 1994a] [Berkling and Barnard 1994b] [Corredor-Ardoy *et al.* 1997] [Hazen and Zue 1994]. A good inventory needs to phonetically cover as many targeted languages as possible. This method can be effective when phonemes from all targeted languages form a closed set, as studied by Hazen and Zue [1994]. Human perceptual experiments have also shown a similar effect, where listeners' LID performance improved as their exposure to each language increased [Muthusamy *et al.* 1994].

This API-based tokenization approach was recently explored [Ma *et al.* 2005] by using a set of all 124 phones and 4 noise units from English, Korean, and Mandarin, and by extrapolating them to nine other languages in the NIST LRE tasks. This set of 128 units is referred to as API-I in Table 1, which is a proprietary phone set defined for the IIR-LID[1] 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 [Ma *et al.* 2005] and a text categorization approach to LID [Gao *et al.* 2005], which formed the basis for the vector based feature extraction approach 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 in the OGI-TS[2] multi-language telephone speech 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 counts for each language is given in Table 1.

[1] Language Identification Corpus of the Institute for Infocomm Research

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

Table 1. The languages and phone sets of API-I & -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

2.2 Acoustic Segment Model (ASM)

The above phone-based language characterization approach 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 used languages. Second, a large collection of transcribed speech data is needed to train the acoustic and language phone models for each language. To alleviate these difficulties, a data-driven method that does not rely on exact phonetic transcriptions is preferred. It can be obtained by constructing consistent acoustic segment models (ASMs) [Lee *et al.* 1998] intended to cover the entire sound space of all spoken languages in an unsupervised manner.

As in other types of hidden Markov modeling, the initialization of ASMs is a critical factor for success. Note that the unsupervised, data-driven procedure for obtaining ASMs may result in many unnecessary small segments because of a lack of phonetic or prosodic constraints, (e.g. the number of segments in a word and the duration of an ASM) imposed during segmentation. This problem is especially severe when segmenting a huge collection of speech utterances from a large population of speakers with different language backgrounds. The API approach uses phonetically defined units in the sound inventory. It has the advantage of adopting phonetic constraints in the segmentation process. By using API to bootstrap ASM, our approach effectively incorporates some phonetic knowledge about a few languages in the initialization step to guide the ASM training process as described below:

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. The *J* models had better not to overlap very much according to their acoustic characteristics, and their number should be large enough to provide a reasonable acoustic coverage for all of the target languages.

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 re-train an HMM.

Step 5: Repeat steps 2-4 several times until convergence is achieved.

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 [Rabiner 1989] which adopt iterative optimization of segmentation and maximization. We have found that API-bootstrapped ASMs are more stable than the randomly initialized ASMs. It outperformed API by a big margin in the 1996 NIST LRE task as reported in [Ma *et al.* 2005]. The detailed results will be given in section 4.1.

With an established acoustic inventory obtained using the ASM method, we can tokenize any given speech utterance to obtain a token sequence $\hat{T}$, in a form similar to a text-like document. 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 consistent tokenization than in the underlying lexical characterization of a spoken utterance. The self-organizing ASM modeling approach offers the key property that it does not require the training speech data to be directly or indirectly phonetically transcribed.

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

The main difference between API and ASM lies in the relaxation of phone transcription for segmentation. In API, phone models are trained according to manually transcribed phone labels, while in ASM, segmentation is done in iterations using automatic recognition results. In this way, ASM gains two advantages: (i) it allows us to adjust a set of API phones from a small number of selected languages towards a larger set of targeted languages; (ii) ASMs can be trained on acoustic data similar to that used for the LID task, thus potentially minimizing the mismatch between the test data and the APIs that were trained on a prior set of phonetically transcribed speech data.

3. Frontend and Backend Formulations

In this section, we will first briefly discuss prior works cast in the formalism of phone recognition (PR) and phone-based language modeling (LM). Then, we will propose our phone recognition frontend based on ASM acoustic modeling and our backend of vector space modeling for language classification. Note that the ASMs are no longer the phonemes defined

in Table 1. For easy reference, we will continue to refer to the ASM tokenization process as phone recognition (PR).

3.1 PPR-LM Configuration

A typical LID system is illustrated in Figure 1, which shows a collection of parallel phone recognizers (PPR frontend) that serve as voice tokenizers, referred to as the frontend. A frontend converts spoken utterances into sequences of token symbols, or spoken documents. It is followed by a set of *n*-gram phone language models (LM) that impose constraints on phone decoding and provide language scores. The LM pool converts an input spoken utterance into a vector of interpolated LM scores. The language models and the classifier are referred to as the backend. The backend classifier models a spoken language using a collection of training samples, in the form of LM score vectors.

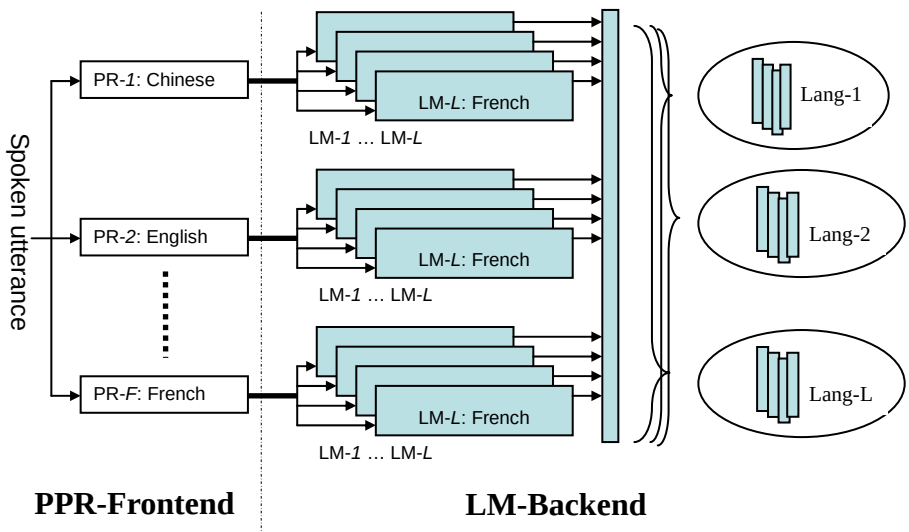


Figure 1. Block diagram of a PPR-LM LID system

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

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

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

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

where the first term on the right hand side of (2) is the probability of *O* given *T* and its acoustic model λ_f^{AM}, the second term is the language probability of *T* given the language model $\lambda_{f,l}^{LM}$, and 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, it can be approximated by finding the most likely phone sequence $\hat{T}_f$, for each phone recognizer *f*, using the Viterbi algorithm:

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

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 follows:

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

We assume that the *F* parallel language-dependent acoustic phone models can be used to approximate the acoustic space of *L* languages. After a spoken utterance is decoded by the *F* recognizers, it needs to be evaluated by a set of $F \times L$ language models to establish comparability. The system formulated by (3) and (4) is known as parallel PRLM, or P-PRLM [Zissman 1996]. In this paper, it will be referred to as PPR-LM to identify its PPR frontend and LM backend.

3.2 UPR-LM Configuration

In prior works, researchers also looked into a language-independent phone recognizer with a set of universal acoustic units, or phones that are common to all languages. The formulations of (3) and (4) can be simplified as a two-step optimization:

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

$$\hat{l} = \arg\max_{l \in A} \left[\log P\left(\hat{T} \mid \lambda_l^{LM}\right) \right], \tag{6}$$

where *B* is the set of all possible token sequences for all languages. 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 a universal phone recognition (UPR) frontend as shown in Figure 2.

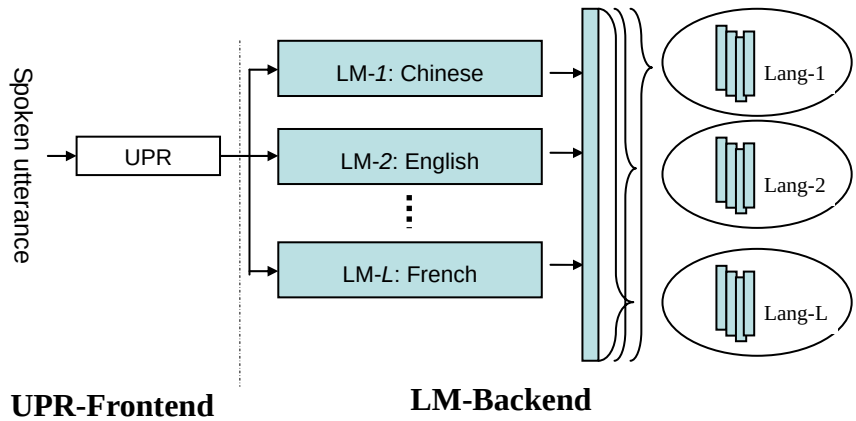


Figure 2. Block diagram of a UPR-LM LID system

A number of UPR-LM systems have been proposed along these lines, such as the ALI system [Hazen and Zue 1994], the single-language PRLM system [Zissman 1996], and the language-independent phone recognition approach [Corredor-Ardoy *et al.* 1997]. However, the training of phone sets in these systems requires phonetic transcription of all training utterances.

In this paper, we propose a new way of training the set of universal acoustic units using the ASM approach described in Section 2.2, where acoustic models are trained in a self-organized and unsupervised manner. This provides two obvious advantages: (1) the unsupervised strategy allows the frontend to adapt easily to new languages without the need for phonetic transcription; (2) the universal acoustic units can be flexibly partitioned into subsets to work for the parallel phone recognition (PPR) frontend as shown in Figure 1.

3.3 Vector Space Modeling for Language Classification

Vector space modeling (VSM) has become a standard tool in Information Retrieval (IR) systems since its introduction decades ago [Salton 1971]. It uses a vector to represent a text document. One of the advantages of the method is that it allows the discriminative training of classifiers 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 vectors 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 spoken documents. Individually, these substrings may be shared across languages. Collectively, the pattern of their co-occurrences discriminates one language from another.

Suppose that the sequence of feature vectors *O* is decoded into a sequence of Ω acoustic units $\hat{T} = \{t_1, ..., t_\pi, ..., t_\Omega\}$, where each unit is drawn from the universal ASM inventory of *J* models in a UPR frontend, $t_\pi \in \{w_1, w_2, ...w_J\}$. 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 $p(w_n \mid w_1, ...w_{n-1}) = \quad p(t_\pi = w_n \mid t_{\pi-1} = w_1, ..., t_{\pi-n+1} = w_{n-1})$. Its dimension is equal to the total number of *n*-gram patterns needed to highlight the overall behavior of an utterance:

$$\bar{\lambda} = \left(p(w_1), ..., p(w_2 \mid w_1), ..., p(w_3 \mid w_1, w_2), ... \right). \tag{7}$$

The vector $\bar{\lambda}$ is also called a *bag-of-sounds* (BOS) vector [Li and Ma 2005], which represents a spoken utterance in a document vector in a same way as in text-based document vector representation [Gao *et al.* 2005] [Salton 1971]. The vector space modeling approach evaluates the goodness of fit, or score function, using a vector-based distance, such as an inner product:

$$P\left(\hat{T} \left| \lambda_l^{LM} \right.\right) \propto \bar{\lambda}^T \cdot \omega_l \,, \tag{8}$$

where ω_l is a language-dependent weight vector with dimension equal to $\bar{\lambda}$, 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 as high order *n*-gram patterns are included. This makes it suitable for discriminative feature extraction and selection.

For the PPR frontend, the sequence of feature vectors *O* is decoded into *F* independent sequences of acoustic units. A BOS vector $\bar{\lambda}_f$ can be derived from each sequence in the same way as in (7) for each phone recognizer. A grand BOS vector is, therefore, constructed by concatenating the *F* vectors $\bar{\lambda}_f$ to represent the input spoken utterance. With multiple tokenizers, we hope that the grand BOS vector will describe the input spoken utterance in a greater detail.

Term weighting [Bellegarda 2000] 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 often occur in a few documents but not as often in others provide high indexing power for these documents. On the other hand, patterns which occur very often in all documents possess little indexing power. This desirable property has led to the development of a number of term weighting schemes, such as *tf-idf*, that are

commonly used in information retrieval [Salton 1971], natural language call routing [Kuo and Lee 2003], and text categorization [Gao *et al.* 2004]. We adopt the standard *tf-idf* term weighting scheme in this paper.

Note that the variations [Berkling and Barnard 1994a] [Corredor-Ardoy *et al.* 1997] [Hazen and Zue 1994] [Zissman 1996] of LM backend systems proposed in prior works used cross-entropy or perplexity based language model scores, which are based on similarity matching, for language classification decision-making. The VSM can be seen as an attempt to enhance the discrimination power offered by *n*-gram phonotactic information.

3.4 VSM-Backend

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 be used to discriminate between individual spoken documents. The given collection of spoken documents in the training set from a particular language forms the same language category. LID can be considered the process of classifying a spoken document into some pre-defined language categories. An unknown testing utterance to be identified can be represented as a query vector, and LID can then be performed as in text document classification [Joachims 2002]. We can then utilize any classifier learning technique, such as support vector machine [Sebastiani 2002] or artificial neural network [Haykin 1994], developed by the text categorization community to design language classifiers. An LID system with the VSM-backend is shown in Figure 3 for the PPR frontend and in Figure 4 for the UPR frontend. The VSM-backend takes as inputs *n*-gram statistics in the form of document vectors. The backend structure remains the same for both the UPR and PPR frontends, so long as we can represent the voice tokenizations from the PPR/UPR frontend in document vectors. With the document vectors from the training database, the backend groups training document vectors into language classes.

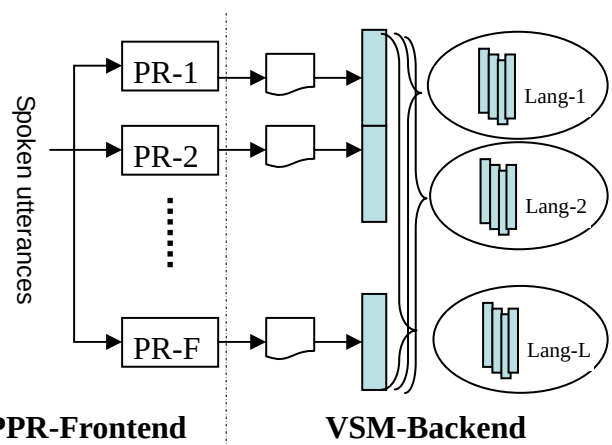


Figure 3. Block diagram of a PPR-VSM LID system

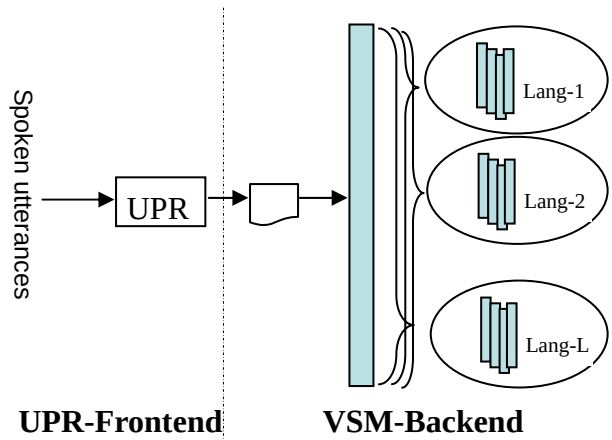


Figure 4. Block diagram of a UPR-VSM LID system

3.5 Classifiers in VSM-Backend

There are many ways to reduce the dimension of the document vectors and to enhance the discriminative ability, such as by applying latent semantic indexing (LSI). In this paper, we propose to use a set of output scores from an array of support vector machines (SVMs) as the dimension-reduced vector for the final classifier. For each of L target languages, we have a number of high dimensional training vectors as shown in (7). An SVM is a 2-way classifier used to partition the high dimensional vector space. We construct an SVM between each of the language pairs. As a result, we obtain $L \times (L-1)/2$ pair-wise SVM classifiers for the L target languages. For each input utterance, an output score is generated from each of the pair-wise SVM classifiers, resulting in a vector of $L \times (L-1)/2$ dimensions that represent $L \times (L-1)/2$ pair-wise language discriminative scores, called a *discriminative vector*. The linear kernel is adopted for the SVMs in the SVMlight V6.01 tool[3] implementation. In this way, each language category can be represented by a Gaussian mixture model (GMM) which is trained on the *discriminative vectors* of the training utterances. The GMM classifiers are built as part of the VSM-backend for decision-making. At run-time, the VSM-backend identifies the language of a spoken document in language recognition/detection trials and verifies the language identity of a spoken document in language verification trials.

To summarize, we have discussed an LID paradigm of two frontend options for voice tokenization, PPR or UPR, and two backend options, LM or VSM. The PPR-LM and UPR-LM configurations were well studied in the previous works. However, a systematic comparison among the PPR-LM, UPR-LM, PPR-VSM and UPR-VSM configurations has not

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

been made. Thus, we conducted a comparative study over the four combinations of frontends and backends based on ASM acoustic units.

4. Experiments

We followed the experiment setup in the NIST Language Recognition Evaluation (LRE) tasks[4]. The tasks were intended to establish a baseline of performance capability for language recognition of conversational telephone speech. The evaluation was carried out on recorded telephony speech in 12 languages, Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, and Vietnamese, for the 1996, 2003 NIST LRE tasks, and in 7 languages, English, Hindi, Japanese, Korean, Mandarin, Spanish, and Tamil for the 2005 NIST LRE task.

In this paper, training sets for building models came from two corpora, namely: (i) the 6-language OGI-TS database with English, German, Hindi, Japanese, Mandarin, and Spanish; and (ii) the 12-language LDC *CallFriend*[5] database. The OGI-TS database was only used to bootstrap the acoustic models of an initial set of phones. It consists of 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 in the same 12 languages that are in the 1996 and 2003 NIST LRE tasks, but without phonetic transcriptions. The two databases are independent of each other.

In the OGI-TS database, there is less than 1 hour of speech in each language. In the *CallFriend* database, each of the 12 language databases consists of 40 telephone conversations with each lasting approximately 30 minutes, giving a total of about 20 hours per language. In language modeling, each conversation in the training set is segmented into overlapping sessions, resulting in about 12,000 sessions for each of three durations per language. These three durations are 3 seconds, 10 seconds, and 30 seconds. The 1996 NIST LRE evaluation data consists of 1,503, 1,501, and 1,492 sessions for 3 seconds, 10 seconds, and 30 seconds respectively. The 2003 NIST LRE evaluation data consist of 1,200 sessions per duration. The 2005 NIST LRE evaluation data consist of 3,662 sessions per duration.

4.1 Frontend Acoustic Modeling

Our early research on API and ASM [Ma *et al.* 2005] showed the following:

(1) The ASM frontend outperformed the API frontend when followed by the VSM backend;

---

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

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

In the language identification task on the 12 languages in the 1996 NIST LRE evaluation data (30 seconds only), 128 API units were trained with the API-I phone set by using the IIR-LID database, and 128 ASM units were further obtained based on the bootstrapping of APIs using the *CallFriend* database. With the UPR-VSM setup using the BOS vectors containing both unigram and bi-gram, an error rate of 13.9% was achieved with ASMs, while the error rate with APIs was 19.2%.

(2) Higher ASM coverage, with a larger ASM inventory and higher order n-gram (trigram), improved the LID performance;

Under the same experiment setups as in (1), we investigated the effects of the acoustic coverage by clustering the 128 ASM units into 64 and 32 ASMs according to acoustic similarity. Table 2 compares the acoustic and linguistic coverage achieved using 32, 64, and 128 AMS units, and by using unigram, bi-gram, and trigram. It shows that these reduced-sized ASM units greatly impaired the discrimination power of the ASM systems. We needed a reasonable number of ASM units that was large enough in order to cover the sound variation in all of the languages.

Table 2. Comparison of acoustic and linguistic coverage

Error Rate (%)	32-ASM	64-ASM	128-ASM
Unigrams	40.1	26.7	22.3
Bigrams	32.6	18.6	13.9
Trigrams	27.9	NA	NA

(3) Note that the initialization of acoustic model has a strong impact on the resulting models in HMM training. Apparently, API phone models provide good initialization for ASM models.

In the following experiments, we used phonetically labeled OGI-TS corpus to train API-II phones, as shown in Table 1.

For each utterance, 39-dimensional features consisting of 12 MFCCs and normalized energy, plus their first and second order time derivatives were extracted for each frame. Utterance based cepstral mean subtraction was applied to the features to remove channel distortion. A two-step modeling approach was adopted. First, the language dependent phonemes in API-II were trained language by language based on the phonetic training database. Each phoneme was modeled with an HMM of 3 states. The resulting 258 API-II phonemes were then used to bootstrap 258 ASM models. The 258 ASM models were further trained based on the 12 language *CallFriend* database in an unsupervised manner as described in Section 2.2. The average segment lengths of the 258 ASM models based on the *CallFriend* database ranged from 33 ms to 150 ms.

4.2 Backend Classifier

First, the 15-language/dialect[6] training data in the *CallFriend* database was tokenized to obtain a collection of text-like phone sequences from each of the 6 tokenizers. We computed PPR-LM scores based on the resulting phone sequences. We trained up to 3-gram phone LMs for each PPR-LM tokenizer-target language pair, resulting in $15 \times 6 = 90$ LMs. For each input utterance, 90 interpolated scores were derived to form a vector. In this way, the training utterances could be represented by a collection of 90-dimension score vectors. Similarly, for UPR-LM, we trained up to 3-gram phone LMs for each of the target languages, resulting in 15 LMs. The training utterances were then represented by a collection of 15-dimension score vectors. Both PPR-LM and UPR-LM shared the same LM backend design, which adopted the framework of PR-LM. The low dimension score vectors could be modeled by the Gaussian Mixture Model (GMM) [Torres-Carrasquillo *et al.* 2002].

Next, we will discuss the VSM backend classifier [Li and Ma 2005]. The VSM backend first converted the text-like tokenization sequences into BOS vectors as discussed in Section 3.3. Then the BOS vectors were further processed by the support vector machines to derive $L \times (L-1)/2$ dimensional discriminative vectors. For a frontend of 6 languages, English, Mandarin, Japanese, Hindi, Spanish and German, there were 258 phonemes in total. In the case of UPR, we derived a BOS vector containing both mono-phones and bi-phones with 66,822 (= 258^2 + 258) elements. In the case of PPR, we derived a BOS vector with 11,708 (= $48^2 + 39^2 + 52^2 + 51^2 + 32^2 + 36^2 + 48 + 39 + 52 + 51 + 32 + 36$) elements. The BOS vectors were then reduced to a discriminative vector of $105 = 15 \times 14/2$ dimensions for an evaluation task involving 15 target languages. In this study, both LM score vectors and BOS discriminative vectors were modeled by the GMM classifier.

The main difference between the LM and the VSM backend classifier lies in the representation of the document vector. In LM backend, the document vector is characterized by interpolated LM scores, while in VSM backend, the document vector is derived from outputs of support vector machines, which introduce discriminative ability between language pairs. If we see the LM backend as a likelihood-based classifier, then the VSM backend is a discrimination-motivated classifier.

4.3 Four LID Systems

We have discussed two different frontends, PPR and UPR, and two different backends, LM and VSM. To gain insight into the behavior of each of the frontends and backends, it is desirable to investigate the performance of each of the four combined systems as shown in

[6] In the 12-language *CallFriend* database, English, Mandarin, and Spanish have two dialects, respectively.

Figure 5, 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 with two different settings. First, the 258 ASMs were arranged in a 6-language PPR frontend. They were redistributed according to their API-II definitions into 6 languages. Second, they were lumped together in a single UPR frontend. The training of the 258-ASM was discussed in Section 2.2. We used the GMM classifier in the LM backend and VSM backend, in which we trained 512-mixture GMMs to model the desired language and to model all its competing languages, and reported the equal error rates (EER%) between false-alarm and miss-detect.

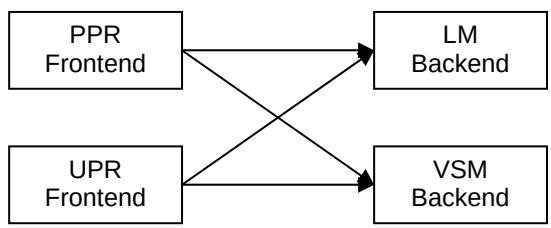


Figure 5. Block diagram of four combinations of frontends and backends

The UPR-VSM system follows the block diagram of the language-independent acoustic phone recognition approach [Ma *et al.* 2005]. PPR-LM was implemented as in [Zissman 1996]. The LM backend uses trigrams to derive phonotactic scores. The results for the 1996, 2003 and 2005 NIST LRE tasks are shown in Tables 3, 4, and 5, respectively. In Table 6, we also report the execution times for the 2003 NIST LRE task obtained in terms of the real-time-factor (xRT) with an Intel Xeon 2.80 GHz CPU.

Before discussing results, we will examine the effects of the combined frontends and backends. In the combined systems, there are two unique frontend settings, PPR and UPR. PPR converts an input spoken utterance into 6 spoken documents using the parallel frontend, while UPR converts an input into a single document. However, there are four unique LM and VSM backend settings. The LM in PPR-LM and that in the UPR-LM are different; the former has 15×6 n-gram language models, while the latter only has 15 language models. In other words, the former LM classifier is more complex, with a larger number of parameters, than the latter. The VSM in PPR-VSM and the VSM in UPR-VSM have different levels of complexity as well. The former VSM processes vectors with 11,708 dimensions, while the latter processes those with 66,822 dimensions, as discussed in Section 4.2. The vectors in PPR-VSM and UPR-VSM are shown in Figure 6.

Although the dimensionality of V-PPR is lower than that of V-UPR, V-PPR is 6 times as dense as V-UPR, resulting in more complex support vector machine partitions (SVM) [Vapnik 1995]. In other words, the VSM classifier in the PPR-VSM is more complex than that in UPR-VSM. In terms of the 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.

Table 3. EER% comparison of 4 systems on 1996 NIST LRE

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 4. EER% comparison of 4 systems on 2003 NIST LRE (without Russian)

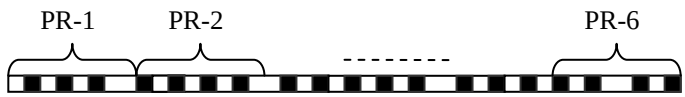
System	30-second	10-second	3-second
PPR-VSM	3.62	10.36	21.25
PPR-LM	4.54	11.31	20.37
UPR-VSM	6.33	13.35	24.30
UPR-LM	10.24	19.23	30.28

Table 5. EER% comparison of 4 systems on 2005 NIST LRE (all 7-language trials, without German)

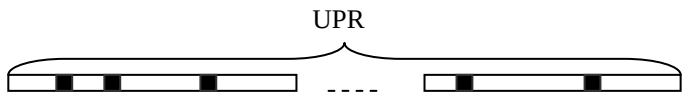
System	30-second	10-second	3-second
PPR-VSM	5.78	12.48	24.23
PPR-LM	6.76	12.48	22.48
UPR-VSM	9.10	16.80	26.52
UPR-LM	13.71	22.40	30.89

Table 6. Execution time comparison on 2003 NIST LRE (Real-Time-Factor of 30-sec 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



6(a) A 11,708 dimensional vector from 6 PPRs (V-PPR)



6(b) A 66,822 dimensional vector from the UPR (V-UPR)

Figure 6. Two different spoken document vectors in PPR-VSM and UPR-VSM

Summarizing the results obtained in the three NIST LRE tasks, we have the following findings:

(i) The VSM backend demonstrates a clear advantage over the LM backend for the 30-second and 10-second trials. This can be easily explained by the fact that VSM models are designed to capture phonotactics over the context of the whole spoken document. As a result, VSM favors longer utterances which provide richer long span phonotactic information.

(ii) The system performance highly correlates with the complexity of the system architectures. This can be seen in Tables 3, 4, and 5, which show that PPR-VSM achieved the best result with an EER of 2.75%, 3.62%, and 5.78% in the 30-second 1996, 2003 and 2005 NIST LRE tasks, respectively, followed by PPR-LM, UPR-VSM, and UPR-LM. Note that we can increase the system complexity by using more PPRs. We expect that more PPRs will improve the PPR-VSM system performance further.

(iii) Although PPR-LM outperformed UPR-VSM in general, the UPR frontend was superior in computational efficiency during run-time operation over the PPR frontend. In Table 6, we find that the systems with the UPR frontend ran almost 60% faster than those with the PPR frontend.

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

4.4 Overall Performance Comparison

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

For the sake of brevity, we only compare results obtained in the 30-second tests, which represent the primary condition of interest in the NIST LRE tasks. Systems 1, 2, and 3 in Table 7 were trained and tested on the same databases. Therefore, the results can be directly compared. They are extracted from Tables 3 and 4. We also cite two results from recent reports [Gauvain *et al.* 2004] [Singer *et al.* 2003] as references. Table 7 shows that the performance of PPR-VSM system is among the best in the 1996 and 2003 NIST LRE tasks.

Ma *et al.* [2005] reported that the API-bootstrapped ASM outperformed API phone models in the LID task. This paper extends our previous work through comprehensive benchmarking, which produced further findings and validated the effectiveness of the proposed VSM solution. The systems reported in this paper contributed to the ensemble classifier that participated in the 2005 NIST LRE representing IIR site.

The proposed VSM-based language classifier compares phonotactic statistics from spoken documents. We have not explored the use of acoustic scores resulting from the tokenization process. It was reported that combining information of acoustic scores along with phonotactic statistics produced good results [Corredor-Ardoy *et al.* 1997] [Singer *et al.* 2003] [Torres-Carrasquillo *et al.* 2002]. Furthermore, fusion of phonotactic statistics at different levels of resolutions also improved overall performance [Lim *et al.* 2005]. We have good reason to expect that fusion among our 4 combinative systems, or between our systems and other existing methods, including GMM tokenizer [Torres-Carrasquillo *et al.* 2002], will lead to further improvements.

Table 7. EER% Benchmark on 30-second 1996/2003 NIST LRE

	System	1996 LRE	2003 LRE
1	PPR-VSM	2.75	3.62
2	PPR-LM	2.92	4.54
3	UPR-VSM	4.87	6.33
4	Phone Lattice [Gauvain *et al.* 2004]	3.20	4.00
5	Parallel PRLM [Singer *et al.* 2003]	5.60	6.60

5. Conclusion

We have studied the effects of frontends and backends in the LID system. In the following, we summarize our findings. (1) A vector space modeling (VSM) backend consistently outperformed the LM backend in the combination tests; (2) The PPR-VSM system

configuration demonstrated superior performance across all of the primary tasks (30-second tests); (3) The UPR frontend was effective in run-time operation.

In this study, we formulated both LM backend and VSM backend classifiers as a vector classification problem. The traditional LM backend applies similarity based approach to the vector representation of spoken documents. The VSM backend represents spoken documents using discriminative vectors derived from the outputs of support vector machines. We achieved EERs of 2.75% and 3.62% in the 30-second 1996 and 2003 NIST LRE tasks respectively with the PPR-VSM system. These are some of the best reported results for a single LID classifier. The VSM backend was also successfully implemented in IIR's submission to 2005 NIST LRE. The good results can be credited to the enhanced discriminatory ability of the VSM backend.

Exploring the *bag-of-sounds* spoken document vectors using the bigram statistics of ASM acoustic units, 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, 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, multilingual spoken document categorization by topic. We are also interested in exploring other 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 in an acoustic word vocabulary, in order to increase the indexing power of these words for their corresponding languages.

References

Adda-Decker, M., F. Antoine, P.B. Mareuil, I. Vasilescu, L. Lamel, J. Vaissiere, E. Geoffrois, and J.-S. Lienard, "Phonetic Knowledge, Phonotactics and Perceptual Validation for Automatic Language Identification," In *Proceedings of the 15th International Congress of Phonetic Sciences*, 2003, pp. 747-750.

Bellegarda, J.R., "Exploiting Latent Semantic Information in Statistical Language Modeling," In *Proceedings of IEEE*, 88(8), 2000, pp. 1279-1296.

Berkling, K.M., and E. Barnard, "Analysis of phoneme-based features for language identification," *International Conference on Acoustics, Speech & Signal Processing*, 1994a, vol. 1, pp. 289-292.

Berkling, K.M., and E. Barnard, "Language identification of six languages based on a common set of broad phonemes," *International Conference on Spoken Language Processing*, 1994b, pp. 1891-1894.

Corredor-Ardoy, C., J.L. Gauvain, M. Adda-Decker, and L. Lamel, "Language identification with language-independent acoustic models," *5th European Conference on Speech Communication and Technology*, 1997, vol. 1, pp. 55-58.

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

Gao, S., B. Ma, H. Li, and C.-H. Lee, "A text-categorization approach to spoken language identification," 9th *European Conference on Speech Communication and Technology (Interspeech)*, 2005, pp. 2837-2840.

Gao, S., W. Wu, C.-H. Lee, and T.-S. Chua, "A MFoM learning approach to robust multiclass multi-label text categorization," *International Conference on Machine Learning*, 2004, pp. 329-336.

Gauvain, J.L., A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," *International Conference on Spoken Language Processing*, 2004.

Haykin, S., *Neural Networks: A comprehensive foundation*, McMillan, 1994.

Hazen, T.J., and V. W. Zue, "Recent Improvements in An Approach to Segment-Based Automatic Language Identification," *International Conference on Spoken Language Processing*, 1994, pp. 1883 -1886.

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

Joachims, T., *Learning to classify text using support vector machines*, Kluwer Academic Publishers, 2002.

Kirchhoff, K., S. Parandekar, and J. Bilmes, "Mixed Memory Markov Models for Automatic Language Identification," *International Conference on Acoustics, Speech & Signal Processing*, 2002, vol. 1, pp. 761-764.

Kuo, H.K.J., and C.-H. Lee, "Discriminative training of natural language call routers," *IEEE Trans. on Speech and Audio Proces*sing, 11(1), 2003, pp. 24-35.

Lee, C.-H., "From Knowledge-Ignorant to Knowledge-Rich Modeling: A New Speech Research Paradigm for Next Generation Automatic Speech Recognition," *International Conference on Spoken Language Processing*, 2004, pp.109-112.

Lee, C.-H., F. K. Soong, and B.-H. Juang, "A Segment Model Based Approach to Speech Recognition," *International Conference on Acoustics, Speech & Signal Processing*, 1998, pp. 501-504.

Li, H., and B. Ma, "A Phonotactic Language Model for Spoken Language Identification," *43rd Meeting of the Association for Computational Linguistics*, 2005, pp. 515-522.

Lim, B.P., H. Li, and B. Ma, "Using local and global phonotactic features in Chinese dialect identification," *International Conference on Acoustics, Speech & Signal Processing*, 2005, vol. 1, pp. 577-580.

Ma, B., C. Guan, H. Li, and C.-H. Lee, "Multilingual Speech Recognition with Language Identification," *International Conference on Spoken Language Processing*, 2002, pp. 505-508.

Ma, B., H. Li, and C.-H. Lee, "An Acoustic Segment Modeling Approach to Automatic Language Identification," 9th *European Conference on Speech Communication and Technology (Interspeech)*, 2005, pp. 2829-2832.

Matrouf, D., M. Adda-Decker, L.F. Lamel, and J.-L. Gauvain, "Language Identification Incorporating Lexical Information," *International Conference on Spoken Language Processing*, 1998.

Muthusamy, Y.K., N. Jain, and R. A. Cole, "Perceptual Benchmarks for Automatic Language Identification," *International Conference on Acoustics, Speech & Signal Processing*, 1994, vol. 1, pp. 333-336.

Nagarajan, T., and H.A. Murthy, "Language Identification Using Parallel Syllable-Like Unit Recognition," *International Conference on Acoustics, Speech & Signal Processing*, 2004, vol. 1, pp. 401-404.

Parandekar, S., and K. Kirchhoff, "Multi-Stream Language Identification Using Data-Driven Dependency Selection," *International Conference on Acoustics, Speech & Signal Processing*, 2003, vol. 1, pp. 28-31.

Rabiner, L.R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, 77(2), 1989, pp. 257-286.

Sai Jayram, A.K.V., V. Ramasubramanian, and T. V. Sreenivas, "Language identification using parallel sub-word recognition," *International Conference on Acoustics, Speech & Signal Processing*, 2003, vol. 1, pp. 32-35.

Salton, G., *The SMART retrievl system*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

Sebastiani, F., "Machine learning in automated text categorization," *ACM Computing Surveys*, 34(1), 2002, pp. 1-47.

Singer, E., P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, and D.A. Reynolds, "Acoustic, Phonetic and Discriminative Approaches to Automatic Language Recognition," 8th *European Conference on Speech Communication and Technology*, 2003, pp. 1345-1348.

Sugiyama, M., "Automatic Language Recognition Using Acoustic Features," *International Conference on Acoustics, Speech & Signal Processing*, 1991, vol. 2, pp. 813-816.

Torres-Carrasquillo, P.A., D.A. Reynolds and J. R. Deller, Jr, "Language Identification Using Gaussian Mixture Model Tokenization," *International Conference on Acoustics, Speech & Signal Processing*, 2002, vol. 1, pp. 757-760.

Vapnik, V., *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

Yan, Y., and E. Barnard, "An Approach to Automatic Language Identification Based on Language Dependent Phone Recognition," *International Conference on Acoustics, Speech & Signal Processing*, 1995, vol. 5, pp. 3511-3514.

Zissman, M.A., "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Trans. Speech and Audio Proc.*, 4(1), 1996, pp. 31-44.