# Web-Based Automatic Language Identification System

Mauricio M. Olvera, Angel Sánchez, and Larry H. Escobar

***Abstract*—Language Identification (LID) is the automated process of identifying what language is being spoken from a sample of speech by an unknown speaker. In this work we present a web-based LID system using Shifted Delta Cepstral (SDC) features derived from Mel-Frequency Cepstral Coefficients to gather relevant acoustic information from speech signals, and Gaussian Mixture Models (GMM) as a classifier. Speech corpora comprising four languages (English, Spanish, French and German) were made up of recordings from audio media found on the Internet. A web implementation was done using up-to-date web technologies with GNU Octave running on the server side to perform numerical computations. Results showed a system accuracy ranging from 72.5% to up to 80% depending on the duration of speech test segments.**

***Index Terms*—GMM, language identification, MFCC, SDC.**

## I. Introduction

Automatic Language identification (LID) is the task of identifying the language being spoken within a speech utterance [1]. Some of the most important applications of LID systems consist in multilingual spoken dialog systems, which can be found in information kiosks at international airports or touristic places; front-ends for multilingual translation systems in which the input speech can be in one of several languages; index systems for search and classification of audio files stored in language corpora databases. Telephone companies, call-centers and emergency call services also use LID systems to route incoming calls to an operator fluent in the caller's language.

Previous approaches to language identification have proposed the use of pitch contours, formant vectors, acoustic, phonotactic and prosodic features, etc., to represent the characteristic of a language; and for modeling and classification, several techniques as Markov's models, artificial neural networks, vector quantization and other clustering algorithms have been tried. As research in this area of speech signal processing has become widely popular in recent years, novel approaches have been developed in order to increase efficiency and performance of such systems. In this respect, some of the most preferred choices in state-of-the-art LID systems are Parallel Phone Recognition Language Modeling (PPRLM) for phonotactic-based systems and Support Vector Machines (SVM) and Gaussian Mixture Models (GMM) for acoustic-based systems.

Manuscript received on April 25, 2016; revised September 13, 2016. This work was supported by the Digital Signal Processing Laboratory at the Postgraduate Engineering School of the National Autonomous University of Mexico.

The authors are with the Faculty of Engineering, National Autonomous University of Mexico, Coyoacán, Mexico City, 04510, Mexico (e-mail: molveraz@comunidad.unam.mx, angelsanchez@ingenieria.unam.mx, larryesc@gmail.com).

In this work we have developed a web-based GMM-based LID system accessible online at http://odin.fi-b.unam.mx/labdsp/LIDSystem/, using Mel-Frequency Cepstral Coefficients (MFCC), one of the most popular feature extraction techniques in both speaker recognition and language identification systems, which aims to represent all the relevant acoustic and phonetic information of a language, and Shifted Delta Cepstra (SDC) features to improve system accuracy.

## II. Acoustic-Phonetic Approach for LID

Most LID systems are based on acoustic-phonetic information as it is the most suitable approach for representing the characteristics of a language. Acoustic features are characterized by a set of parameters that can be directly acquired from speech and serve as the basic building block for the extraction of phonetic information. As there is a finite set of meaningful sounds appearing in every world language, phonetic inventories differ from one language to another. Even when languages have identical phones, the frequency of occurrence of phones differs across languages [2]. Hence, acoustic-phonetic information turns out to be a satisfactory approach to represent language features.

## III. Speech Corpora

In order to perform the automatic language identification task, we extracted recordings from video and audio media found on the Internet, such as podcasts of news reports, sports and entertainment, as well as audiobooks and conversations from movies and interviews.

Our speech corpora are composed of four languages: English, Spanish, French and German. Each corpus constitutes a collection of audio recordings of 30 speakers of different ages and accents. The duration of the recordings is approximately 60 minutes for every language. As a LID system implementation consists of two phases, namely training and testing; 45 minutes from every speech corpus were used in the training phase to obtain the models for each language and the remaining 15 minutes were used for testing the system.

## IV. Feature Extraction

### A. MFCC Derivation

In order to extract the most relevant acoustic-phonetic information from speech utterances, we have used MFCC, one of the preferred methods for feature extraction [3].

Each step in the derivation of MFCC takes into account computational considerations and perceptual properties of the

doi: 10.18178/ijiee.2016.6.5.643

human auditory system. Fig. 1 shows the process of MFCC derivation.

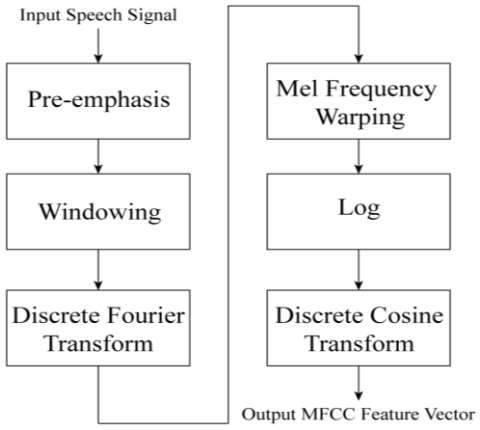


Fig. 1. Block diagram of MFCC derivation.

The first step in MFCC feature extraction is to pre-emphasize the speech signal, so that the drop in energy across high frequencies caused naturally by the glottal pulse can be boosted. In order to lift the magnitude of higher frequencies with respect to the magnitude of lower frequencies, pre-emphasis is done by using a first-order high-pass Finite Impulse Response (FIR) filter whose difference equation is described as follows:

$$y(n) = x(n) - \alpha x(n-1) \tag{1}$$

where, $\alpha$ can take on values from 0.9 to 1.

The next step is to partition the signal in small frames of N samples to achieve signal stationarity, and then window each individual frame to tone down discontinuities at the edges. Windowing can be performed by different types of window functions.

Then, it is necessary to convert each frame of N samples in time domain to frequency domain to be able to extract spectral information present in the signal by applying Discrete Fourier Transform (DFT). The input to the DFT is the windowed signal $x(n)$ and the output for each of N discrete frequency bands is a complex number X(k). A commonly used algorithm for computing the DFT is the Fast Fourier Transform (FFT) with reduced execution time.

The spectrum obtained from the previous stage is wrapped by a perceptual scale that models the human perception of pitch. The Mel scale is described by (2), and is implemented by creating a filter bank, in which filters are spaced linearly at frequencies below 1 kHz and logarithmically for the rest of frequencies. Usually a set of 20 to 40 filters is performed.

$$f_{mel}=1127\ln\left(1+\frac{f_{Hz}}{700}\right) \tag{2}$$

Then, motivated by the non-linear behavior of the human ear to sounds, we take the log of each of the Mel spectrum values. Finally the last step of MFCC feature construction is to apply the Discrete Cosine Transform (DCT) to decorrelate the components of each frame, where usually the first dozen of DCT coefficients representing the cepstral values are generally kept.

### B. SDC Feature Construction

LID system performance can be greatly increased by the computation of Shifted Delta Cepstral (SDC) features, which are an extension of delta-cepstral coefficients. They capture the variation over many frames of data, a useful property that might explain the effectiveness of the SDC features in discriminating information across languages [4]. SDC coefficients are computed as shown in Fig. 2.

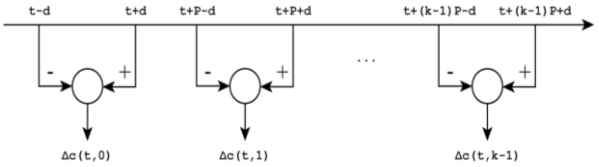


Fig. 2. SDC feature computation (adapted from [5]).

They consist of four parameters known as N-d-P-K. According to [5], $N$ is the number of cepstral coefficients computed at each frame, $d$ represents the time advance and delay for the delta computation, $k$ is the number of blocks whose delta coefficients are concatenated to form the final feature vector, and $P$ is the time shift between consecutive blocks.

At frame $t$, the SDC feature vector is:

$$SDC(t)=\begin{pmatrix}\Delta c(t,0)\\ \Delta c(t,1)\\ \vdots\\ \Delta c(t,k-1)\end{pmatrix} \tag{3}$$

where, $\Delta c(t,i) = c(t+iP+d) - c(t+iP-d)$, for $i$=0,1,2,…,$k$–1.

## V. Language Modeling

The GMM approach attempts to model the probability density function of feature vectors, $x_i$, by the weighted combination of multi-variate Gaussian densities:

$$p(x|\lambda)=\sum_{i=1}^{M} w_i b_i(x) \tag{4}$$

where, $x$ is a dimensional random vector, $b_i(x)$, $i$=1,2,…,$M$, is the component densities, and $w_i$, i=1,2,…,$M$, are the mixture weights.

The multivariate Gaussian function is defined by (5),

$$b_i(x)=\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\} \tag{5}$$

with mean vector $\mu_i$ and covariance matrix $\sum_i$. The mixture weight satisfies the constraint that:

$$\sum_{i=1}^{M} w_i=1 \tag{6}$$

The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weight from all component densities. These parameters can be

represented by the notation:

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, \; i=1,2,\ldots,M \tag{7}$$

The estimation of the GMM parameters from training data is accomplished by either an iterative two-step process known as the Expectation-Maximization (EM) algorithm or a Maximum A Posteriori (MAP) estimation.

In this work we have estimated such parameters using the EM algorithm, whose basic idea is to estimate a new model λ' given an initial language model λ, so that $P(X|\lambda') \geq P(X|\lambda)$. The new model then becomes the initial model for the next iteration and the process is repeated until convergence [6].

In the first step of the EM algorithm, called expectation or E-step, the expected value of the log-likelihood function is computed based on observed data and initial model parameters. In the second step, called maximization or M-step, the parameters that maximize the expected log-likelihood found on the E-step are computed to be used to determine the distribution of the latent variables in the next iteration [7].

Convergence is guaranteed by the increase of the likelihood at each EM iteration where the following re-estimation formulas are used:

Mixture weights:

$$\bar{w}_i = \frac{1}{T}\sum_{t=1}^{T} Pr(i|x_t, \lambda) \tag{8}$$

Means:

$$\bar{\mu}_i = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda)\, x_t}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)} \tag{9}$$

Variances:

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)} - \bar{\mu}_i^2 \tag{10}$$

The a posteriori probability for component is given by (11).

$$Pr(i|x_t, \lambda) = \frac{w_i b_i(x)}{\sum_{k=1}^{M} w_k b_k(x)} \tag{11}$$

## VI. Language Recognition

During recognition, an unknown speech utterance, *X*, comprising of observations $x_1, x_2, x_3, \ldots, x_T$, is classified by computing the average log-likelihood that each language model produced. This is given by the following equation:

$$p(X|\lambda) = \frac{1}{T}\sum_{t=1}^{T} \log p(x_t|\lambda) \tag{12}$$

where, λ is the model for the corresponding language. The maximum-likelihood classifier hypothesis, *H*, can be calculated as in (13).

$$H = \text{argmax}\; p(x|\lambda_l) \tag{13}$$

where, $l = 1,2,\ldots L$, for *L* number of languages.

## VII. Experimental Setup

Fig. 3 shows our LID system block diagram which consists of two phases, namely training and testing. In the training phase, feature extraction, classification and language modeling take place, while during testing feature matching is performed.

For our implementation, speech corpora used as training data consist of speech utterances sampled at 8 kHz with 16 bits/sample resolution. For the pre-emphasis stage we have set the parameter α to 0.95 as it is most commonly used in Automatic Speech Recognition Systems (ARS). We have blocked the signal into frames of 25 ms with an overlap of 50%, and then windowed it using a Hamming window, whose function is as follows:

$$w(n) = \begin{cases} 0.54\text{-}0.46\cos\left(\frac{2\pi n}{L}\right), & 0 \leq n < L\text{-}1 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

For spectral information extraction, we have used the FFT algorithm for computing the DFT, so we applied a 256-point FFT to compute the magnitude spectrum of the windowed signal. Our Mel filter bank was built out of 23 triangular bandpass filters with pass range from 35 Hz to 4 kHz. Finally, after applying DCT to decorrelate frame components, we have kept only 12 of the 23 cepstral values.

We extracted SDC coefficients from MFCC features applying a 7-1-3-7 SDC scheme, which led to 84-dimensional feature vectors.

Then, for language modeling, we have trained a GMM with 2048 components for each language. Each GMM model was trained by means of the EM algorithm using 11 iterative steps.

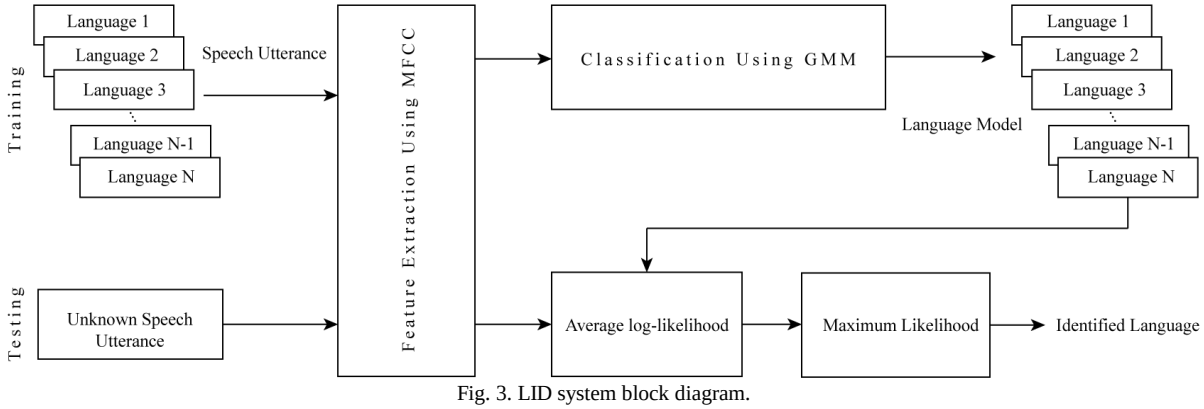


Fig. 3. LID system block diagram.

## VIII. Experimental Results

In order to test the accuracy of our LID System we have used 15 minutes of audio from each speech corpus. Test audios were divided into excerpts of 3, 10, 30 and 60 seconds. System accuracy was done by (15),

$$\%\text{Accuracy} = (\text{Correct} / \text{Total}) \cdot 100 \quad (15)$$

where, *Correct* is the number of speech samples whose language was correctly identified, and *Total* is the total number of samples given for testing. Accuracy percentage for each language as well as the overall system accuracy is shown in Table I.

TABLE I: LID System Accuracy

| **Number of GMM's** | | 2048 | | | |
|---|---|---|---|---|---|
| **Training duration** | | 45 min | | | |
| **Audio duration** | | ***3s*** | ***10s*** | ***30s*** | ***60s*** |
| **Languages** | ***English*** | 80% | 80% | 80% | 80% |
| | ***French*** | 70% | 80% | 75% | 75% |
| | ***Spanish*** | 70% | 80% | 85% | 90% |
| | ***German*** | 70% | 75% | 75% | 80% |
| **Overall System Accuracy** | | **72.5%** | **78.75%** | **78.75%** | **80%** |

## IX. Web Implementation

As stated earlier in this work, our system is accessible online at http://odin.fi-b.unam.mx/labdsp/LIDSystem. For the online implementation we have used up-to-date front-end web technologies such as HTML5, CSS3 and JQuery; and GNU Octave, a high level interpreted language for numerical computations as the system back-end.

The interface of our web implementation consists of two straightforward steps; a user uploads an audio file containing speech and then, clicks on "Identify Language" button and wait until language identification is performed. Fig. 4 shows the web interface for the latter step of the language identification process.

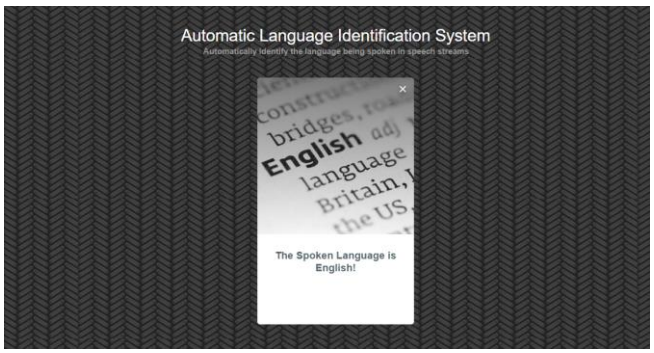


Fig. 4. Web interface for our LID System.

## X. Conclusions

We have implemented a web-based LID system accessible online at http://odin.fi-b.unam.mx/labdsp/LIDSystem/. We have followed an acoustic-phonetic approach as it is one of the most effective ways to represent the characteristics of a language. We derived SDC feature vectors from MFCCs, and then created language models using high order GMMs. The accuracy of our system ranges from 72.5% to 80% depending on the duration of the speech samples. Our online implementation comprises up-to-date front-end web technologies and open source software for numerical computations. Our system can be greatly improved by increasing training duration and we also plan on adding more languages as future work.

## References

[1] Muthusamy, K. Yeshwant, and R. A. Cole, "Automatic language identification using telephone speech," *Neural Networks in Telecommunications*, pp. 233-254, Springer US, 1994.
[2] Muthusamy, K. Yeshwant, E. Barnard, and R. A. Cole, "Reviewing automatic language identification," *Signal Processing Magazine, IEEE*, vol. 11, no. 4, pp. 33-41, 1994.
[3] P. Motlıcek, "Feature extraction in speech coding and recognition," Technical Report of PhD Research Internship in ASP Group, OGI-OHSU, 2002.
[4] D. A. Reynolds, W. M. Campbell, W. Shen, and E. Singer, "Automatic language recognition via spectral and token based approaches," *Springer Handbook of Speech Processing*, pp. 811-824, Springer Berlin Heidelberg, 2008.
[5] P. A. Torres-Carrasquillo, E. Singer, 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," *Interspeech*, 2002.
[6] D. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometrics*, pp. 827-832, 2015.
[7] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

**Mauricio M. Olvera** was born in Mexico City, Mexico in 1992. He received the B.S degree in telecommunications engineering from the National Autonomous University of Mexico (UNAM), Mexico City, Mexico in 2016.

In 2014, he was an exchange student at Purdue University, West Lafayette, Indiana, United States, where his interest in digital signal processing arose. Since April 2015, he has been with the Department of Digital Signal Processing, Faculty of Engineering, UNAM, where he is an assistant professor. His current research interests include automatic language identification, digital signal processing and unsupervised machine learning.

**Angel Sanchez** was born in Mexico City, Mexico, in 1991. He received the B.S degree in computer engineering from the National Autonomous University of Mexico (UNAM), Mexico City, Mexico in 2016.

In 2014 he joined the Coordination of Planning and Development (CPD), Faculty of Engineering, UNAM, as an assistant professor, and in 2016 he became the engineer in charge of the systems area and served as representative of the CPD department to the computer advisory committee of the Faculty of Engineering, UNAM. His current research interests include IoT, cloud computing, robotics and parallel computing.

**Larry H. Escobar** was born in Guatemala, Central America, in 1961. He received the B.S degree in mechanical and electrical engineering and the M.S degree in electrical engineering from the National Autonomous University of Mexico (UNAM), Mexico city, Mexico, in 1992 and 1997, respectively.

In 1991, he joined the Department of Electrical Engineering at UNAM as an assistant professor; in 1992 as an associate professor, and in 1994 he became a full-time professor.

His current research interests include digital filtering, spatial filtering and real-time digital signal processing. Currently, he is coordinator of the Department of Digital Signal Processing, Faculty of Engineering, UNAM. He has published about 45 papers about his technical area in congresses, magazines and books. He has directed 20 undergraduate engineering theses and over 10 postgraduate engineering theses. During his academic work, Prof. Escobar has received tree special catedra awards from the Faculty of Engineering, UNAM.