IJICTR International Journal of Information & Communication Technology Research

Volume 4- Number 2- March 2012 (37-45)

Performance Improvement of Language Identification Using Transcription Based Sequential Approaches & Sequential Kernels Based SVM

Seyed Abbas Hosseini Amereei
Laboratory for Intelligent Sound & Speech Processing
Amirkabir University of Technology
Tehran, Iran
sabbas@aut.ac.ir

Mohammad Mehdi Homayounpour
Laboratory for Intelligent Sound &Speech Processing
Amirkabir University of Technology
Tehran, Iran
homayoun@aut.ac.ir

Received: November 30, 2011- Accepted: February 26, 2012

Abstract— **In this paper a generative frontend based on both phonetic and prosodic features, and also a couple of approaches based on phonetic transcription- Aggregated Phone Recognizer followed by Language Models (APRLM) and Generalized Phone Recognizer followed by Language Models (GPRLM), are investigated. APRLM and GPRLM have few disadvantages since they need phonetic transcription of speech data, and also they use fewer level of information while the generative frontend built upon an ensemble of Gaussian densities uses prosodic and phonetic information altogether. Furthermore, no transcription of speech data is needed in Support Vector Machine (SVM)-based approaches, and they showed better performances in our experiments too. In addition, APRLM and GPRLM are more time consuming than SVM-based approaches. We used Mel-Frequency Cepstral Coefficients (MFCC) in APRLM and GPRLM, and Shifted Delta Cepstrum (SDC) and Pitch Contour Polynomial Approximation (PCPA) features in SVM-based methods. Probabilistic Sequence Kernel (PSK) and Generalized Linear Discriminant Sequence (GLDS) kernels are used in SVM experiments. SVM using GLDS and PSK kernels outperforms GMM in all our LID experiments conducted by applying PCPA features and LID performance improved about 2.1% and 5.9% respectively. The combination of Probabilistic Characteristic Vector using PCPA (PCV-PCPA) and Probabilistic Characteristic Vector using SDC (PCV-SDC) provides further improvements.**

Keywords-component; Language Identification, Probabilistic Characteristic Vector, Pitch Contour Polynomial Approximation, Probabilistic Sequence Kernel, Generalized Linear Discriminant Analysis, APRLM and GPRLM.

I. INTRODUCTION

Automatic Language Identification (LID) is a process of determining the language identity corresponding to a given set of spoken queries. It is an important technology in many applications, such as spoken language translation, multilingual speech recognition, and spoken document retrieval [1]. One of the best known approaches to automatic spoken language recognition is the Parallel Phone Recognition followed by Language Modeling (PPRLM) [2]. In PPRLM, a number of phone recognizers tokenize the input utterance separately in the LID frontend and then the sequences are passed through the LID backend that is composed of some language models. The language models score all input sequences independently and then the average of results shows the language of input signal. We proposed a couple of new methods, Aggregated Phone Recognizer followed by Language Models (APRLM) and Generalized Phone Recognizer

followed by Language Models (GPRLM), which outperform PPRLM in our recent works [1, 3]. In order to investigate all methods in more detail, APRLM and GPRLM will be explained in this paper. These methods need transcription to train phone recognizers while phonetic transcription is not always available, and it is only one of the drawbacks of these methods. Therefore other approaches not requiring any transcription are investigated.

Using pitch contour information for LID was discussed in [5], where pitch contour was approximated using Legendre polynomial, and then polynomial coefficients named PCPA have been used to train GMMs for LID system. GMM models a language using a mixture of Gaussian components. Applying support vector machine (SVM) with either GLDS kernel or PSK kernel instead of the GMM has improved the performance of LID task based on PCPA according to [6]. GLDS is a sequence kernel, mapping variable length sequence of speech signals to fixed length vectors by using correlation matrices of features [9]. PSK is a sequence kernel similar to GLDS mapping input signals to fixed length vectors, however, it is made of several GMMs.

Using support vector machines with generative frontend was proposed in [7] in which the authors have used a probabilistic sequence kernel (PSK) to map variable length speech signals to fixed length vectors called Probabilistic Characteristic Vectors (PCV) and showed that PSK overcomes GLDS kernel in LID tasks. Subsequently, we combined Probabilistic Characteristic Vector based on SDC (PCV-SDC) and Probabilistic Characteristic Vector based on Pitch Contour Polynomial Approximation (PCV-PCPA). Consequently, we proposed new features by concatenating PCV-SDC and PCV-PCPA vectors in [6]. SDC feature will be explained in the Subsection II. Our proposed features and the method proposed in [6] improved LID performance as well. Two PCV features were used in this work, one based on SDC features and the other based on PCPA features. In other words, for more improvement, PCV-SDC and PCV-PCPA have been concatenated in LID systems applying SVM with PSK kernels. Since the proposed feature vectors contain both phonetic and prosodic information, the LID performances have been improved. In this paper, we are trying to investigate all the advantages and flaws of the mentioned methods in a number of experiments and find the best practice in more detail. Section 2 describes the speech features for LID followed by section 3 and section 4 through which the proposed LID features and LID systems are presented in order. In section 5, we explain the used dataset, and the conducted experiments are presented in section 6. Section 7 concludes this paper.

II. SPEECH FEATURES FOR LID

In practice, there are several features used for LID tasks. This section explains a short summary of those features applied to propose new features in this paper.

A. *Shifted Delta Cepstrum (SDC)*

Feature vector extraction for LID systems is typically performed by constructing a feature vector at frame time t. This feature vector consists of cepstra and delta cepstra. However, Torres in [7] showed that improved LID performance could be obtained by using Shifted Delta Cepstrum (SDC) feature vectors created by stacking delta cepstra computed across multiple speech frames. The SDC features are specified by a set of 4 parameters, N, d, P and k, where 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. Accordingly, kN parameters are used for each SDC feature vector, as compared with 2N for conventional cepstra and delta-cepstra feature vectors. For example, the final vector at frame time t is given by the concatenation of all $\Delta c(t + iP)$ values, where $\Delta c(t) = c(t + iP + d) - c(t + iP - d)$.

B. *Pitch Contour Polynomial Approximation (PCPA)*

Pitch Contour Polynomial Approximation was proposed in [5] in which pitch contour is extracted, segmented and approximated successively.

- First of all, pitch contour extraction is done by using Praat program [8]. A proposed method by Boersma is adopted. This method utilizes autocorrelation function to detect vocalic segments and find pitch candidates. Then Viterbi algorithm is used to find the most suitable contour path. Detailed description of each parameter was discussed in Boersma's paper [8]. We conduct the feature extraction method using parameter values listed in Table 1. Pitch values are extracted from 30ms speech frames. The range of pitch values varies from 50 to 500 Hz. The maximum number of candidates and voicing threshold values are also 5 and 0.04 respectively. After pitch contour extraction, pitch contours were smoothed using a median filter.

Table 1. Pitch extraction parameter settings

Pitch Extraction Parameter Settings	
Analysis window length	30 ms
Pitch floor (Hz)	50
Pitch Ceiling (Hz)	500
Max. number of candidates	5
Voicing threshold	0.04

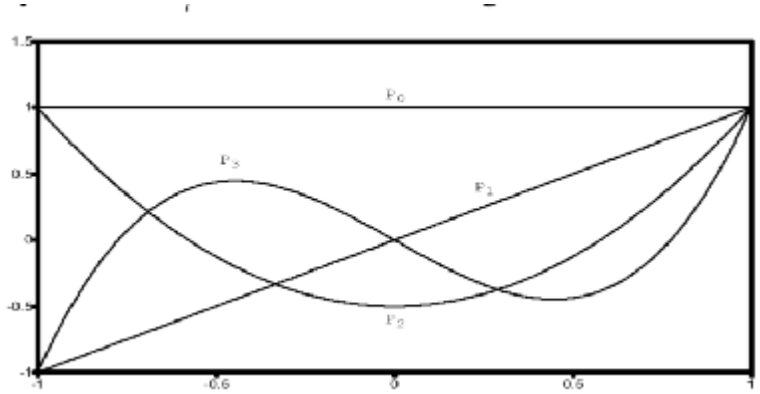

Fig 1. Illustration of Legendre polynomials

• Due to the spontaneous speech, vocalic portion of speech signal may cross syllable or word boundaries. Non-zero pitch values are supposed as pitch contours and therefore pitch contours between two zero values are supposed as one pitch contour segment. Segments are ignored if their length is shorter than four frames, because those lengths are too short to be approximated by 3-rd order polynomial.

• Each segmented pitch contour f_k was approximated by a 3rd order Legendre polynomial in the sense of minimum mean square error:

$$f = \sum_{i=0}^{M} a_i P_i \qquad (1)$$

Where i is the pitch contour index, M is the highest polynomial order, a_i is i-th order coefficient, and P_i is i-th order Legendre polynomial. In most cases, small value of M is sufficient so that we let M=3 here. Legendre polynomials P_i are illustrated in Fig 1.

Notice that P_0 stands for the height of pitch contour, P_1 stands for the slope of pitch contour, P_2 stands for the curvature of pitch contour, and P_3 stands for the S-curvature of pitch contour. With this representation, a feature vector $\vec{V}$ is formed including the length of pitch contour D and four coefficients a_0, a_1, a_2 and a_3. In addition, the authors have claimed that other features are not helpful [5].

C. *Probabilistic Characteristic Vector (PCV)*

In fact, spoken languages differ in the inventory of speech sound units used to produce words and sentences [7]. Although the frequency of occurrence and the order of these sounds appearing in the spoken utterances differ from one language to another, common speech sounds are shared considerably across languages. A universal inventory of speech sounds can be established by combining those from a predefined set of languages referred as basis languages. In [7], authors used a method to extract characteristic vector to classify languages using SVM. Since this method maps variable length speech signals to a fixed length vector and this vector is used as a kernel for SVM, it is named Probabilistic Sequence Kernel (PSK). In that paper, for each language, a Gaussian Mixture Model (GMM) is trained based on SDC features of a language. These GMMs form the core of PSK kernel. PSK kernels are computed using the SDC features and are used during training and testing in SVM models. So, each variable length audio file of the dataset will be represented by a characteristic vector generated by PSK. This characteristic vector is named Probabilistic Characteristic Vector (PCV) because it is based on occurrence probability. Since Probabilistic Characteristic Vectors are generated using SDC features, they are named as PCV-SDC in this paper.

III. PROPOSED FEATURES FOR LID SYSTEMS

Probabilistic Characteristic Vector (PCV) based on PCPA (PCV-PCPA) is the authors' proposed features. As previously mentioned, for generating PCV-SDC vectors, SDC features are used. We suggest using PCPA instead of SDC to generate PCV vectors. Hence, the new vectors are called PCV-PCPA. PCV-PCPA vectors represent prosodic information of languages. In other words, PCPA features are used in the generative frontend.

Besides, since PCV-SDC and PCV-PCPA illustrate phonetic and prosodic properties of languages respectively, we combined these vectors together to have further information entirely. So the new feature is called PCV-SDC&PCPA.

IV. LID SYSTEMS

A. *Sequential Approaches based on Transcription*

• APRLM: Phone recognizer is the most important part of an appropriate PRLM LID system. In [1], we proposed an aggregation algorithm in which multiple sequences tokenized by multiple tokenizers are aggregated and used for training and recognition in a specific kind of PRLM. The output of phone recognizer includes the sequence of phones and their corresponding time interval and log-likelihoods. Substitution, insertion and deletion of phones are three common errors that may occur in phone recognition. Also, start and end point of each recognized phone may be determined mistakenly. These errors are unavoidable in PRLM and PPRLM. In this paper, we have reduced these errors by voting among multiple sequences being tokenized by multiple phone recognizers.

In Aggregated PRLM (APRLM), aggregated phone sequences are used to generate language models produced by computing n-gram statistics of phone sequences. In recognition, score of any phone

sequence is computed by each language model. At last, scores obtained from language models determine the language of test utterance. APRLM in general is similar to PRLM in producing language models, computing test utterances scores and deciding about language.

Each observation is determined by a phone, its start and end points, and its score, and any sequence includes many observations. Here, we propose simple voting among observations. Also, weighted voting can be used by considering score of each observation.

Aggregation algorithm is performed in 8 steps:

a- Create a new empty sequence (aggregated sequence).

b- Create a new phonetic symbol (aggregated observation) and add it to the end of aggregated sequence.

c- Consider the first observation in each original sequence and choose the most used start point among them, and apply it as start point of the aggregated observation.

d- Consider the first observation in each sequence and choose the most used end point among them, and choose it as end point of the aggregated observation.

e- Consider the first observation in each sequence and choose the most used phone among them, and choose it as aggregated observation phone.

f- In all of the original sequences, remove any observation overlapping more than half of its length with the new aggregated observation. In other words, if the end point of the aggregated observation is beyond the mean value of the start point and the end point of the original observation, we suppose that more than of its half is covered by aggregated observation and the original observation can be removed from its sequence.

g- The algorithm ends when all sequences are empty, otherwise go to step b.

h- The end.

For more details, the algorithm is explained by an example given in table 2.

In table 2, four sequences are aggregated in which each observation is shown by {phone (start point, end point)}. Underlined expressions show incorrectness in observations. ASeq means aggregated sequence. According to the table, it can obviously be seen that insertion, deletion and substitution errors decrease using the proposed aggregation algorithm. The time of changing phones is also adjusted more accurately by aggregation procedure.

The first observation in Seq1 is "a (0, 9)", in Seq2 is "a (0, 8)", in Seq3 is "g (0, 10)" and in Seq4 is "a (0, 10)". By voting among the first observation in all sequences, it seems that "a (0, 10)" can be considered as the best choice. The aggregation algorithm removes all first observations from the original related sequences. Now, "c (10, 16)", "f (9, 12)", "c (11, 17)" and "g (11, 16)" are the first observation of sequences Seq1, Seq2, Seq3 and Seq4 respectively. "c (11, 16)" will be selected as the next aggregated observation. By continuing the aggregation algorithm "d (17, 21)", "x (22, 29)" and "s (30, 35)" will be the next selected observations in the aggregated observation sequence.

Phone recognizers have been trained on the phonetically labeled messages of the initial training segment of OGI-MLTS corpus. Subsequently, phone sequences are tokenized by these phone recognizers.

• GPRLM: APRLM is a powerful method, but it is time consuming and therefore its front-end is complicated and its hardware implementation might be expensive. In [3], we have introduced GRPLM or Generalized PRLM as depicted in Fig 2. GPRLM uses a general phone recognizer containing multiple single language phone recognizers. GPRLM is more time efficient than APRLM in its front end, however, has all APRLM advantages.

As mentioned before, phone recognizer is the most important part of a PRLM family LID system. In GPRLM approach, we have suggested a method in which a sequence is tokenized by a single Generalized Phone Recognizer. The Generalized Phone Recognizer has been trained to recognize phones of multiple languages. HTK was used in our experiments to generate phone recognizer. The phone recognizer has been trained to recognize phones of all languages considered in a given LID language identification system, i.e. for each phone of each language an HMM phone model would be trained. So there is only one phone recognizer, but it can recognize all phones of languages that the LID system is used to identify. Once the phone recognizer is trained, its output phone sequence can include any phone from each of LID system languages. So, the phone recognizer is a language-independent phone recognizer.

In GPRLM similar to PRLM, language models are produced by computing n-gram statistics of phone sequences. In recognition phase, score of any phone sequence is computed by each language model. Finally, scores obtained from language models determine the language of test utterance. As shown in Fig. 3, GPRLM is similar to PRLM in producing language models, computing test utterances scores and identifying the language of test utterance.

In the traditional phone recognizer, input signal is mapped to a phone sequence containing phones of a specific language. In other words, this phone recognizer includes one model for each phone of that language. Phone models of multiple languages can be combined to generate a language dependent phone sequence for each input signal. For example, suppose a couple of languages (A and B) are phonetically labeled in a data set, therefore for each phone in those languages one model is trained. In phone recognition phase, input signal is tokenized by all phone models.

Table 2 – Aggregating four phone sequences

Seq.1	**a(0,9), c(10,16), d(17,19), z(20,29), s(30,38)**
Seq.2	a(0,8), *f(9,12)*, c(13,16), d(17,21), x(22,29), q(30,35)
Seq.3	g(0,10), c(11,17), d(18,21), x(22,26), u(27,28), s(29,34)
Seq.4	a(0,10), g(11,16), r(17,21), x(22,35){deletion}
ASeq.	a(0,10), c(11,16), d(17,21), x(22,29), s(30,35)

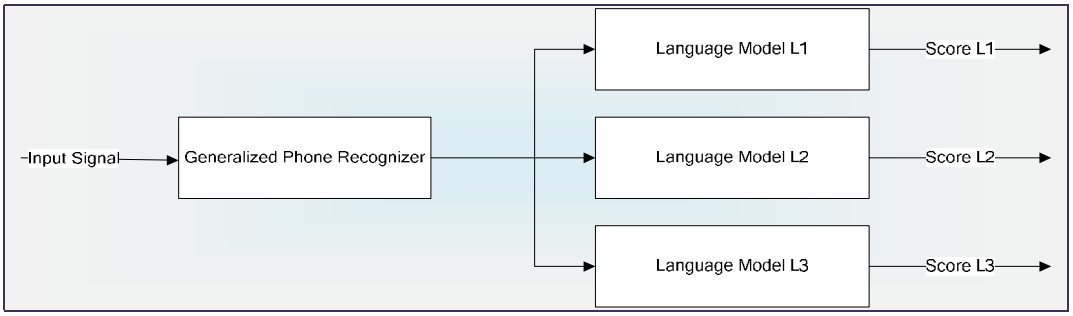

Fig 2. Generalized Phone Recognition followed by Language Modeling (GPRLM) framework

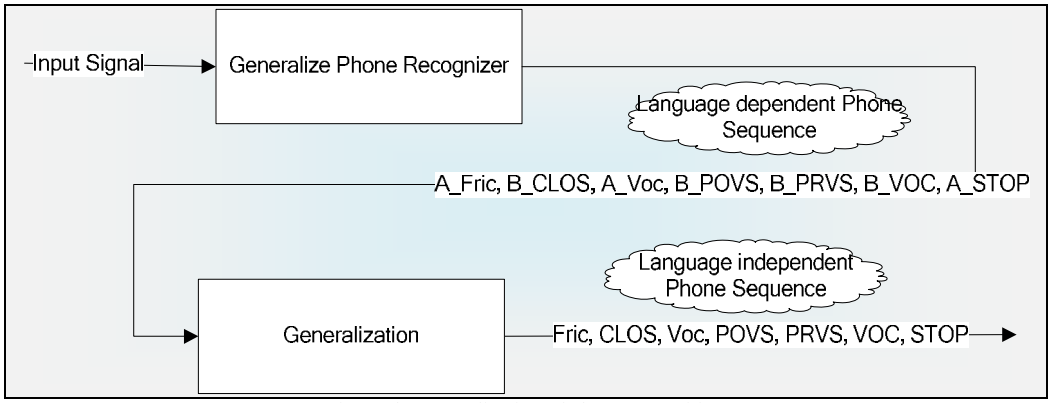

Fig 3. The generalization of phone recognition

B. *SVM based on Sequential Kernels*

GMM and SVM are used for the LID SVM experiments. For each language *l*, a GMM has been trained based on PCPA vectors to classify languages. Here GMM is used to generate the Probabilistic Characteristic Vectors based on either SDC or PCPA features. Besides, Radial Basis Function (RBF) and GLDS kernels have been applied to SVM. The classifiers were trained and tested using LIBSVM, a public domain implementation [10]. In [7], the output of the front end part of PSK that is created by several GMMs is named PCV (Probabilistic Characteristic Vector).

V. CORPUS

We conducted all the experiments on the OGI-MLTS for 45-seconds utterances. The task of evaluation is to detect the presence of a hypothesized target language given a recorded telephony speech. In all the five languages experiments, English, Farsi, French, German and Mandarin are used while in some experiments all ten languages are used. The OGI-MLTS corpus is available from the Linguistic Data Consortium (LDC).

Due to the requirements forced by the research center supporting this research, different experiments were conducted using two, four or five languages instead of using all ten languages included in OGI database. It is worth mentioning that the methods proposed and evaluated in this paper are applicable to all ten OGI languages.

VI. EXPERIMENTS

In this section, several conducted experiments are presented, and then the relevant performances are compared.

A. *APRLM and GPRLM experiments*

Two single PRLMs are used; one by English and the other by Farsi phone recognizer. PPRLM and APRLM use all ten language phone recognizers and GPRLM uses a Generalized Phone Recognizer containing all phone models of all ten languages. All PRLMs, APRLM and GPRLM in these experiments

have 10 language models, while PPRLM has 100 language models; therefore, computing scores in PPRLM is time consuming.

In this section, languages are classified pair-wisely. Therefore, 90 experiments (10* 9 language pairs) for each approach are conducted. Table 3 presents the average of accuracies of all these 90 experiments. More details are presented in tables 4, 5 and 6.

Table 3. LID accuracy on 45s utterances

LID	L VS. L' (average)
PRLM (English Tokenizer)	82.9%
PRLM (Farsi Tokenizer)	81.5%
PPRLM	83.4%
APRLM (Aggregated PRLM)	84.7%
GPRLM (Generalized PRLM)	85.9%

Table 4 . LID accuracy: Japanese versus others

Language	PRLM (EN)	PRLM (FA)	PPRLM	APRLM	GPRLM
English	87%	76%	79%	84%	87%
Farsi	87%	84%	90%	95%	95%
French	84%	82%	92%	90%	84%
German	89%	85%	87%	92%	90%
Korean	92%	81%	89%	92%	94%
Mandarin	84%	92%	83%	90%	92%
Spanish	61%	66%	68%	72%	78%
Tamil	92%	95%	92%	95%	97%
Vietnamese	83%	85%	88%	95%	88%
Average	84%	83%	85%	89%	89%

In Table 3, the results of evaluating the PRLM using either English or Farsi tokenizer are presented. In this experiment, PPRLM with ten tokenizers in ten languages, APRLM with an aggregated tokenizer and GPRLM with a generalized phone recognizer are evaluated. The right hand column in the table shows average accuracy of LID task between two languages, for example English vs. Farsi or French vs. German and so forth. The results show that GPRLM outperforms other approaches. Generalized phone recognizer improves LID performance in GPRLM because the phone recognizer fully covers all phones in all embedded languages and its structure is simpler than APRLM. It means that GPRLM covers more phones than single phone recognizer or multiple language-dependent phone recognizers. Table 4 shows LID task on Japanese language versus other nine languages for an instance. Superiority of GPRLM is evident in the table.

The phones of evaluated languages do not always occur in the language used to train a phone recognizer. To resolve this issue, PPRLM was proposed by Hazan [2]. Suppose that we want to identify a language *L* using a PPRLM including phone recognizers for languages A and B. And also suppose language *L* includes phone ‘x’ which is covered only by phone recognizer of language A and phone ‘y’ which is covered only by phone recognizer of language B. Tokenization accuracy of utterance from language *L* determined by A and B tokenizers is imperfect since 'y' is not covered by A and 'x' is not also covered by B phone recognizer. Therefore phone recognition results for both PRLMs would be low. However, PPRLM can compensate this problem but it is not adequate because both two sequences are defective, so their combination cannot satisfy LID task requirements as we expect. A sequence is needed containing both ‘x’ and ‘y’ phones to perform a better LID task. Aggregated phone recognizer is our solution; sequences produced by an aggregated tokenizer contain all phones existing in input utterances, if at least one tokenizer is able to model them. In fact, GPRLM tokenization performance is as accurate as APRLM but GPRLM is better due to its simplicity; time effectiveness and low cost for hardware implementation.

B. Pair-wise language identification based on Pitch Contour Polynomial Approximation

The Pair-wise language identification is conducted to improve the method presented in [5]. In [5], the authors aimed to identify languages using GMMs based on PCPA features. We use four GMMs (one for each language) trained based on PCPA features. In addition, SVM with PSK kernel and GLDS kernel

based on PCPA are used. It should be noted that we use the trained GMMs for both GMM and SVM-PSK experiments. Each GMM in the experiments contains eight Gaussian components.

Table 5 shows the confusion matrices of pair-wise LID task results. Our proposed methods improve LID task dramatically as shown in Table 5. Each value in the last column in the table shows the average of the performances in each confusion matrix. It is worth mentioning that [5] reported an average of 62.3% for pair-wise language identification performance for four languages (English, Farsi, French and German). The LID performance Average in Table 5 for GMM model is 59.5%. The difference of these two performances may be due to the number of Gaussian components or other configurations in our experiments and the experiments in [5].

C. *Pair-wise language identification by SVM with PSK kernel*

In this section, we plan to investigate the impact of features on LID task. Therefore, SVM with PSK kernel is used based on different features. PCPA, SDC and combination of SDC and PCPA are used in this section. According to [6], the output of PSK was named PCV (Probabilistic Characteristic Vector). In this paper, PCV have been generated based on SDC feature and PCPA feature that have been referred as PCV-SDC and PCV-PCPA respectively. In addition, we concatenate the PCV-SDC and PCV-PCPA to generate a new feature named PCV-SDC&PCPA. Table 6 depicts the result of the experiments conducted by applying SVM with PSK based on the defined features: PCV-SDC, PCV-PCPA and PCV-SDC&PCPA. According to the confusion matrices and the average performances presented in Table 6, our proposed feature outperforms PCV-SDC.

For further comparison, pair-wise LID tasks are conducted with the newly proposed method in Table 7. The proposed method obviously outperforms GPRLM and APRLM. However, it was shown in the previous papers that APRLM and GPRLM outperform PPRLM.

D. *Four languages LID tasks using SVM with PSK kernel*

More experiments were conducted by classification of four languages. In these experiments we envisage to classify four languages, English, Farsi, French and German, by using SVM with PSK. Table 8 shows the results of the experiment which investigates the impact of different features in LID tasks. Table 8 depicts again that our proposed feature outperforms PCV-SDC and improves the four language LID task results about 4%.

Table 5. Pair-wise LID task by GMM based on PCPA features

Method		FA	FR	GE	AVG
GMM	EN	62%	61%	56%	59.5%
	FA		67%	60%	
	FR			51%	
SVM-GLDS	EN	62%	71%	51%	63%
	FA		64%	63%	
	FR			67%	
SVM-PSK	EN	69%	74%	54%	68%
	FA		78%	74%	
	FR			59%	

Table 6. Pair-wise LID task by SVM with PSK based on different features

Feature		FA	FR	GE	AVG
PCV-SDC	EN	79%	92%	82%	84.5%
	FA		95%	80%	
	FR			79%	
PCV-PCPA	EN	62%	71%	51%	63%
	FA		64%	63%	
	FR			67%	
PCV-SDC&PCPA	EN	79%	87%	87%	85.3%
	FA		90%	90%	
	FR			79%	

Table 7. Comparison of pair-wise LID task in different approaches

Feature		FA	FR	GE	AVG
APRLM	EN	77%	68%	72%	
	FA		67%	70%	72.3%
	FR			80%	
GPRLM	EN	77%	68%	69%	
	FA		72%	72%	73.0%
	FR			80%	
New Proposed method (SVM with PSK based on PCV-	EN	79%	87%	87%	
	FA		90%	90%	85.3%
	FR			79%	

Table 8. Performance of four languages LID task using SVM with PSK

Feature	Accuracy
PCV- PCPA	40%
PCV- SDC	65%
PCV- SDC&PCPA	69%

VII. CONCLUSION

In this paper, some LID methods that use phonetically transcribed or non-transcribed speech data for language identification are investigated. These methods were introduced in previous papers including some of our own publications. Due to the importance of the features and the modeling techniques used in LID task, the characteristics of some important and more used features as well as models were investigated, and their merits and demerits were discussed. We applied SVM instead of GMM to improve LID performance using pitch contour polynomial approximation (PCPA). Furthermore, two different SVM sequential kernels including GLDS and PSK, were applied. Using SVM with PSK and GLDS based on PCPA features improved the average performance for pair-wise LID task and this led to 3.5% and 8.5% performance improvement respectively. The most efficient method used in our experiments was SVM-based modeling using PSK kernels. PSK kernels are constructed using a number of GMMs trained using both Pitch Contour Polynomial Approximation (PCPA) and SDC features. In another experiment the impact of using different features using SVM with PSK kernel on LID tasks was studied. PCV-SDC, PCV-PCPA and PCV-SDC&PCPA were compared and it was shown that PCV-SDC&PCPA, as our proposed method, outperforms PCV-SDC in pair-wise LID task by 0.8% in accuracy rate. Furthermore, the proposed approach was compared to our previously proposed methods, APRLM and GPRLM, and a significant improvement was shown in pair-wise LID task. SVM with PSK kernel was also used based on different features and again it was shown that the proposed feature improves 4% the four-language identification accuracy. Subsequently, we combined a couple of features, PCV-SDC and PCV-PCPA, in different level of information i.e. phonetic and prosodic levels to improve LID performance. Furthermore, it is worth mentioning that all the methods in which SVM with sequential kernel is used need no phonetic transcription.

ACKNOWLEDGEMENT

This research was supported by Research Institute for ICT (ex ITRC) under contract T/500/14939.

[1] S. A. Hosseini Amereii, M. M. Homayounpour, "Improvement of language identification performance by Aggregated phone Recognizer, " 17th European Signal Processing Conference (EUSIPCO 2009), Glasgow, Scotland, August 24-28, pp. 1770-1773, 2009.

[2] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech, " IEEE Transactions on Speech and Audio Processing, vol. 4, no. 1, pp. 31-44, 1996.

[3] S. A. Hosseini Amereii, M. M. Homayounpour, "Improvement of language identification performance by Generalized phone Recognizer," Proceeding of the 14th CSI Computer Conference (CSICC 2009) Tehran, Iran, Octobor 20-21, pp. 596-600, 2009.

[4] K. A. Lee, C. You, and H. Li, "Spoken Language Recognition Using Support Vector Machines with Generative Front-end," in Proc. ICASSP 2008, pp.4153-4156, 2008.

[5] C.Y. Lin, H.C. Wang, "Language identification using pitch information," in Proc. ICASSP 2005, Philadelphia, USA, Vol. 1, pp.601-604, 2005.

[6] S. A. Hosseini Amereei, M.M. Homayounpour, "Using Probabilistic Characteristic Vector Based on Both Phonetic and Prosodic Features for Language Identification," in 5th International Symposium on Telecommunications(IST2010), Tehran, Iran, December 4-6, 2010.

[7] P. A Torres-Carrasquillo, 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, pp. 89-92, 2002.

[8] P. Boersma, and D. Weenink, "Praat: doing phonetics by computer," http://www.praat.org.

[9] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," Computer Speech and Language, vol. 20, no. 2-3, pp. 210-229, 2006.

[10] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm .

Seyed Abbas Hosseini Amereei was born in 1983 in Iran. He received his B.Sc. degree in Computer Engineering from University of Tehran in 2005 and his M.Sc. degree in Computer Engineering from Amirkabir University of Technology in 2008. He is now working at University of Calgary, Computer Science Department as a research assistant. His research interests include signal and speech processing, software development and engineering, and agile methods as well.

Mohammad Mehdi Homayounpour was born in 1960 in Iran. He received his B.Sc. degree in Electronics from Amirkabir University of Technology in 1986, his M.Sc. in Telecommunications from Khaje Nasireddin Toosi in 1989, and his Ph.D. in Electrical Engineering from Paris-11 University, Paris, France. He has been a faculty member of Computer Engineering and IT Department at Amirkabir University of Technology (Tehran Polytechnics), Tehran, Iran, since 1995. His research interests include signal & speech processing, natural language processing, hardware design and multimedia.