NOISY CHANNEL ADAPTATION IN LANGUAGE IDENTIFICATION

Sriram Ganapathy, Mohamed Omar, Jason Pelecanos

IBM T.J Watson Research Center, Yorktown Heights, NY, USA.

ABSTRACT

Language identification (LID) of speech data recorded over noisy communication channels is a challenging problem especially when the LID system is tested on speech data from an unseen communication channel (not seen in training). In this paper, we consider the scenario in which a small amount of adaptation data is available from a new communication channel. Various approaches are investigated for efficient utilization of the adaptation data in a supervised as well as unsupervised setting. In a supervised adaptation framework, we show that support vector machines (SVMs) with higher order polynomial kernels (HO-SVM) trained using lower dimensional representations of the the Gaussian mixture model supervectors (GSVs) provide significant performance improvements over the baseline SVM-GSV system. In these LID experiments, we obtain 30% reduction in error-rate with 6 hours of adaptation data for a new channel. For unsupervised adaptation, we develop an iterative procedure for re-labeling the development data using a co-training framework. In these experiments, we obtain considerable improvements (relative improvements of 13 %) over a self-training framework with the HO-SVM models.

Index Terms**—** Language Identification, Noisy Communication Channel, Supervised and Unsupervised Adaptation.

1. INTRODUCTION

Although state-of-the-art LID systems perform reasonably well in clean speech environments, the task of language identification in noisy environments continues to be challenging. One of the goals of the DARPA Robust Automatic Transcription of Speech (RATS) program [1], is the language identification of speech signals received over communication channels that are extremely noisy and highly distorted. In this database, a clean source signal is transmitted over eight different radio channels where the variation across channels results in a range of degradation modes. Each channel induces its own acoustic signature based on its modulation type, carrier channel bandwidth and device operating points.

In a realistic scenario, the eight channels are not representative of the potential radio communication devices used in the field. In order to investigate the effects of an unseen communication channel (not seen in training) on the LID system, we experiment with a leave-one-out strategy by holding out the training data from a particular channel of interest. In these experiments, it was found that there is significant performance degradation on this unseen channel data compared to the seen channels (reported in Sec. 2.3). This can be attributed to the unique characteristics induced by the radio communication channels.

To our knowledge, the problem of acoustic mismatch between training and test conditions is relatively unexplored in language identification. In the past, background model synthesis [2] and feature mapping [3] have been proposed for adapting a speaker recognition system to a new telephone hand-set. However, the effects of radio communication channel are non-linear/time varying and cause acoustic artifacts which are different from the traditional channel mismatch setting.

In order to simplify the problem of channel mismatch, we consider the scenario of having a small amount of development data from an unseen communication channel. We investigate various approaches for the efficient utilization of the development data in supervised and unsupervised frameworks. The baseline LID system is based on support vector machine-Gaussian mixture model super-vector (SVM-GSV) system [4] using shifted delta cepstrum (SDC) features with a delta computation window of 7-1-3-7 [5].

In the supervised setting, we investigate the application of SVMs with higher-order (HO) polynomial kernels [6] based on a low-dimensional principal component analysis (PCA) representation of the GSVs. HO-SVMs transform the low-dimensional PCA vectors onto high dimensional feature spaces and try to separate the language classes in this high dimensional space by modeling the dependencies across different input dimensions. We use the supervised adaptation data from the new channel for learning the HO-SVM models. This approach provides significant improvements over the baseline system (about 30% reduction in equal error-rate with 6 hours of adaptation data for the new channel).

For unsupervised adaptation, we investigate the application of the co-training algorithm described in [7, 8]. Co-training is a machine learning algorithm that uses multiple

This work was supported in part by Contract No. D11PC20192 DOI/NBC under the RATS program. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

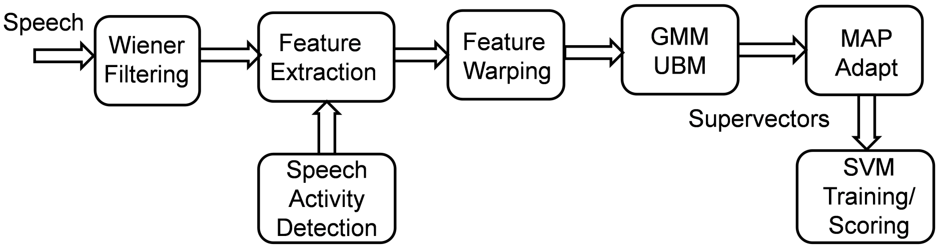

Fig. 1. *Block Schematic of the SVM-GSV LID System.*

weak classifiers and incrementally uses the unlabeled data. The assumption in co-training is that the classifiers can co-train each other, as one can label samples that are difficult for the other. The most confidently classified examples from one classifier are added to the original pool of labeled data and used to the train the other classifier. The process can continue for several iterations. Co-training has been successfully applied to a wide-variety of classification problems (for example, email classification [9], sentence recognition [10] etc).

In this paper, we use the training data from the seen channels as the supervised data for training the classifiers. We investigate the use of HO-SVM and multi-layer perceptron (MLP) based classifiers in the co-training framework to estimate the labels for the unsupervised development data from a new channel. In these unsupervised adaptation experiments with 30 hours of adaptation data, the co-training approach provides 13% reduction in error-rate over a self-training framework with the HO-SVM models.

The rest of the paper is organized as follows. In Sec. 2, we describe the baseline SVM-GSV LID system and report the performance of the LID system on the seen and unseen channels. Adaptation experiments in a supervised framework are described in Sec. 3. Unsupervised adaptation using the iterative co-training procedure is described in Sec. 4. Sec. 5 concludes with a summary of the adaptation experiments.

2. BASELINE LID SYSTEM

2.1. System Description

The block schematic of the baseline LID system is shown in Fig. 1. The speech signal is processed by Wiener filtering [11]. The Wiener filter output is used for feature extraction which calculates the shifted delta cepstrum [5] using a computation window of 7-1-3-7. These 98 dimensional features are used to train a Gaussian mixture model-Universal background model (GMM-UBM) with 1024 mixture components. Then, for a given speech recording, maximum-a-posteriori (MAP) adaptation is performed using the UBM and the adapted mixture component means are concatenated to form a long super-vector (SV). The SVs are used to train language specific support vector machines (SVM) which learn to discriminate the target language from the rest. For testing, the SV from the test sample is evaluated using each SVM to generate scores and the language identity claim is verified by comparing the score against a pre-set threshold. The performance of the LID system is measured in terms of equal error rate (EER).

Table 1. Performance (EER %) of LID system in matched and mismatched conditions.

Matched Train on All	
Channel D	3.5
Channel H	4.1
Avg. of all channels	3.6
Mis-matched-Leave D Out	
Unseen Channel D	15.1
Avg. of seen channels	3.5
Mis-matched Leave H Out	
Unseen Channel H	12.4
Avg. of seen channels	3.2

2.2. Database

The development and test data for the LID experiments use the LDC releases of phase-I RATS LID development [1]. This consists of speech recordings from previous NIST-LRE clean recordings as well as other RATS clean recordings passed through eight (A-H) noisy communication channels. The five target languages are Arabic, Farsi, Dari, Pashto and Urdu. In addition to this, the database consists of several other imposter languages. In our experiments, the UBM is trained using $73,143$ recordings with 270 hours of data from each of the eight noisy communication channels and the test set consists of $14,328$ recordings. These recordings contain of 120 seconds of speech.

2.3. Results

The baseline experiments use the entire development data from all the channels (referred to as the Matched case). We also experiment with the situation where the speech data from one channel is omitted from the training (Mismatched case).

Table 2. Performance (EER %) of the LID system in matched and mismatched conditions using 6 hours of supervised development data for the baseline SVM-GSV system and the PCA-SVM system with HO kernels.

Cond.	Avg. Perf. Seen	Unseen-D	Avg. Perf. Seen	Unseen-H
Matched	3.6	3.5	3.6	4.1
Mis-matched	3.5	15.1	3.2	12.4
SVM adaptation	3.4	8.4	3.2	8.6
PCA-SVM-linear	3.4	9.6	3.1	13.4
PCA-HO-SVM-2nd-order	1.6	6.5	1.4	6.6
PCA-HO-SVM-3rd-order	1.4	6.3	1.4	5.7
PCA-HO-SVM-4th-order	1.4	5.9	1.1	5.2
PCA-HO-SVM-5th-order	1.3	5.9	0.6	4.7

In this case, the results are split on the seen and the unseen channels (the channel which was left out of training). In this paper, we use channels D and H for mismatched experiments (which were the channels with the most significant degradation in unseen conditions.).

The performance of the baseline SVM-GSV system for matched and mismatched conditions is reported in Table 1. The baseline system achieves an EER of 3.6% for the matched condition. As seen here, the performance is significantly degraded if speech data from the communication channel of interest is not seen in training. These can be attributed to the channel specific acoustic characteristics induced by the radio communication system.

3. SUPERVISED ADAPTATION

We consider the scenario of having a small amount of development data from an unseen channel. In this section, we assume that the development data is labeled. The scenario of unsupervised adaptation is discussed in the next section. Further, we also assume that there are equal amounts of development data for each language of interest.

For the pilot experiments, we assume 1 hour of development data from each language of interest and 1 hour of data from each of the imposter languages yielding 6 hours of labeled data from the unseen channel. This corresponds to 2.2% of the total 270 hours available from the channel. The baseline experiment is the re-learning of the SVMs using the additional development data along with the supervised data from the other seen channels. This reduces the error-rate by about 40% from the completely unseen setup as shown in Table 2.

We investigate the use of compact representations for the GSVs using principal component analysis (PCA). As before, the GSVs are extracted using SDC features and the GMM-UBM. The high dimensional SVs (100k dim.) extracted from the training data are used to learn a PCA projection matrix which transforms the SVs to a lower dimensional sub-space by preserving the directions of maximum variance. This is similar to the recent i-vector approaches used in language identification [12].

The dimensionality reduced SVs are used in SVM learning with higher order (HO) polynomial kernels [6]. The HO polynomial kernel can be written as,

$$k(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^d \tag{1}$$

where $\mathbf{x}_i$,$\mathbf{x}_j$ denote the PCA vectors used for SVM training and d denotes the order of the polynomial. We refer to this as the HO-SVM system. For our experiments, we use PCA dimensionality of 800 and experiment with $d = 1, .., 5$, where $d = 1$ represents a linear kernel.

The performance of the HO-SVM with PCA vectors for the seen and unseen channels is also shown in Table 2. The application of the HO-SVM results in significant improvements for the unseen channel (relative improvements of about 30 %). The performance improvements are also significant for the seen channels using HO-SVMs [6]. This may be attributed to the increased language separability in a higher order kernel subspace as opposed to the original high dimensional SV space. These improvements are consistent for the two channels (D,H) considered here.

In Fig. 2, the performance of the LID system when the amount of adaptation data is varied for channel-D is shown. We report the performance using the 2^{nd} and 5^{th} order kernel as well as the baseline SVM-GSV system. The higher order kernels based on PCA projections of SVs provide consistent improvements over the baseline SVM-GSV system even with 1.5 hours of development data. The second-order kernel is better for the unseen channel case. However, the higher order kernels are beneficial for all supervised experiments. In comparison with the baseline SVM-GSV system, we observe a relative improvement of 30% in unseen conditions using only about 6 hours of the new channel data.

4. UNSUPERVISED ADAPTATION

In this section, we discuss the scenario of having small quantities of development data without any language label information. The problem is more challenging than the supervised adaptation case. In these experiments we use 5 hours of development data from each target language and 5 hours of data

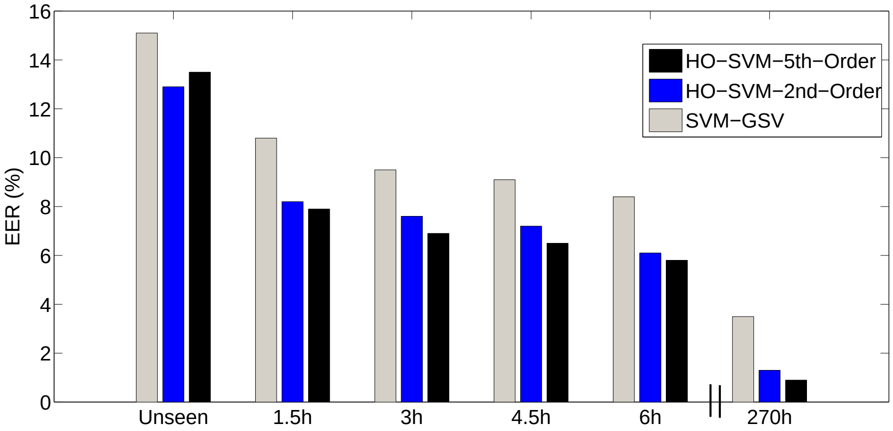

Fig. 2. *Performance as a function of the amount of adaptation data for channel-D for SVM-GSV system and the HO-SVM system with 2nd and 5th order kernels.*

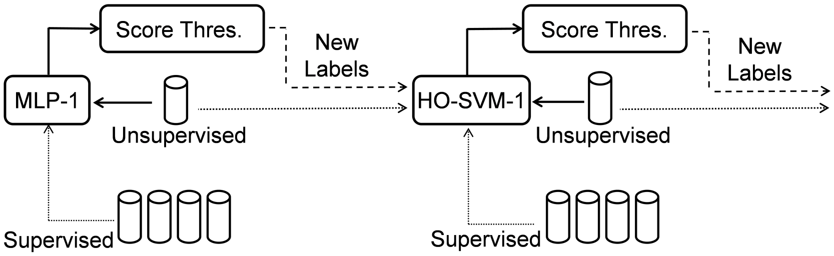

Fig. 3. *First iteration of co-training using MLP and HO-SVM models.*

from the set of non-target languages yielding 30 hours of unsupervised adaptation data. Further, since the HO-SVMs perform better than the baseline SVM-GSV system we use only the HO-SVM system with second order and third order kernel in the unsupervised adaptation experiments. We use the unlabeled development data in PCA training. We use a weighted PCA learning where the new channel data is weighted more compared to the data from the other seen channels.

We investigate the applicability of co-training [8] for this semi-supervised adaptation problem using the supervised data from the seen channels and the unsupervised data from the unseen channel. The co-training algorithm works by learning two or more classifiers trained on the input labeled data which are then used to label the unlabeled adaptation data separately. In our application, we use two classifiers, MLP and SVM models, trained on the PCA vectors. For the unsupervised adaptation data, the confident labels generated by one classifier is used to train the other classifier and vice-versa. The underlying assumption is that the confident labels generated by one classifier may be more beneficial in discriminative learning for the other classifier as opposed to the use of these confident labels for re-training the same classifier (self-training). This is inherently related to the notion that the errors from the two classifiers are less correlated which would result in the confident examples from one classifier boosting the learning of the other classifier.

In this paper, we describe the application of the co-training for language identification where the two classifiers used are the HO-SVM model with a 2^{nd} order kernel and a multi-layer perceptron (MLP) model. Although different feature sets can be used for both classifiers to increase the diversity, we use the same lower dimensional PCA representation as input to both the classifiers. Similar to the SVM learning, three layer MLPs are trained (with a configuration of 800x100x2) to classify the target and non-target examples for each language using a sigmoidal hidden unit and soft-max non-linearity at the output layer. After each iteration of learning, the output scores on the development data for each MLP/SVM model are arranged in a decreasing order and the highly confident examples are chosen for re-training in the next iteration.

The first iteration of the proposed co-training framework used in the LID system is shown in Fig. 3. Language specific MLP detectors are trained using the supervised training data (from seen channels). Then, the unsupervised development data is passed through each MLP and a portion of the

Table 3. Performance (EER %) of the LID system using 30 hours of unsupervised development data from channel-D for MLP/HO-SVM co-training and self-training using HO-SVM with 2nd order (M2) and 3rd order (M3) kernels.

Cond.	Unseen			Unsup. Adaptation										Sup. Adapt	
				Self Training				Co-Training							
				Iter1		Iter2		Iter1			Iter2				
	MLP	M2	M3	M2	M3	M2	M3	MLP	M2	M3	MLP	M2	M3	M2	M3
Avg. Seen	3.1	1.7	1.5	1.6	1.4	1.6	1.4	3.2	1.6	1.4	3.1	1.6	1.3	1.6	1.3
Unseen-D	12.1	11.0	11.2	9.3	8.8	9.1	8.7	9.4	8.5	8.6	9.3	8.0	7.7	5.0	4.9

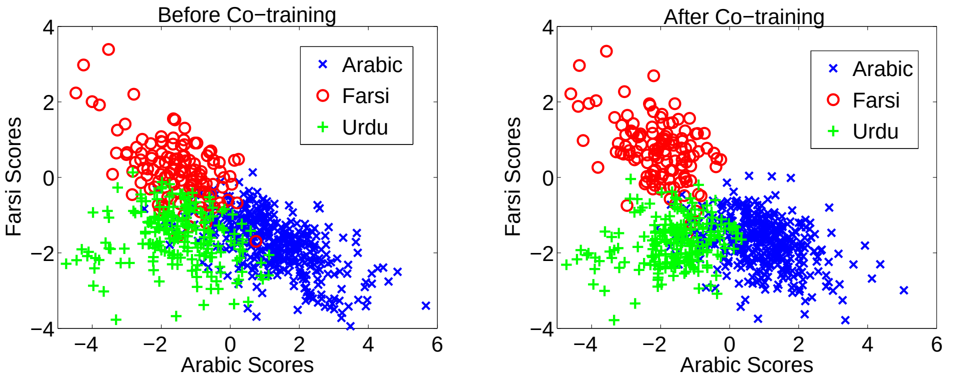

Fig. 4. *Score scatter plots before and after co-training for channel-D test data using the second order kernel.*

labels (selected from recordings with high scores) is used for the HO-SVM re-training along with the supervised data. The trained HO-SVM models are used to detect the language identity of the development data and a portion of the recordings with high scores is selected for MLP re-training in the next iteration.

The results of the LID system using this iterative procedure are shown in Table 3. The performance of the MLP system is worse for the seen channels compared to the HO-SVM[1]. The performance of the HO-SVM in the self-training mode is shown next. We use the HO-SVM with 2^{nd} order kernel (M2) for re-labeling the unsupervised data. The confident labels from the first iteration are used in the second iteration of SVM learning using 2^{nd} and 3^{rd} order polynomial kernels. Here, we use the confident labels from the MLP and the HO-SVM with 2^{nd} order kernel (M2) in an iterative manner to re-label the unsupervised data. This is then used for SVM learning using 2^{nd} and 3^{rd} order polynomial kernels.

The performance of the co-training framework improves over the self-training approach. Using 30 hours of unsupervised data, we obtain a relative improvement of 13% using the co-training framework over the self-training framework. The co-training algorithm also improves the performance of the MLP based LID system. In the co-training and self-training algorithms, we use 10 hours of adaptation data for the first iteration and 15 hours for the second iteration. The performance of the self-training and the co-training algorithm does not improve beyond the second iteration.

The scatter plots for the Arabic and Farsi scores before and after co-training of the HO-SVM models are shown in Fig. 4. The data used for these plots come from Arabic, Farsi and Urdu recordings from the unseen channel-D. As seen here, the separation of the scores for the target languages is improved by the co-training framework.

In a repeated experiment, the application of the co-training procedure to unseen channel H is shown in Table 4. Here, we use the same procedure as before using the same thresholds. The performance improves by about 21% with the co-training algorithm. These experiments show that the performance improvements obtained on channel D data are consistent with those obtained on the other unseen channel H data.

5. DISCUSSION AND CONCLUSIONS

In this paper, we have discussed the issue of channel mismatch in the case of language identification of speech recorded over noisy radio communication channels. The channel mismatch can be attributed to a number of acoustic distortions introduced by the channel like frequency shifting, companding

[1] The MLP based LID system was not optimized in this work in terms of number of parameters, choice of input feature etc. The main motivation was to use an alternate classifier to boost the performance of the SVM system for an unseen channel.

Table 4. Performance (EER %) of the LID system using 30 hours of unsupervised development data from channel H. for HO-SVM with 2nd order kernel (M2).

Cond.	Avg. Perf. Seen	Unseen Chn.H
Unseen		
M2	1.4	11.0
Co-Training		
M2-ITER1	1.4	9.3
M2-ITER2	1.4	8.7
Fully Supervised		
M2	1.4	5.8

and burst noise. We have addressed the scenario of having a small quantity of development data for improving the performance on a new channel of interest. In a supervised setting, the LID system using lower dimensional PCA representation of the supervectors along with a higher order SVM provides significant improvements over the baseline SVM-GSV system. In an unsupervised setting, the co-training algorithm is applied to boost the the performance of the HO-SVM system.

The application of co-training for unsupervised adaptation experiments made certain important assumptions. These include a uniform selection of development data from each target language and selection of thresholds for each iteration without any validation set. The use of the same input features in the two classifiers also reduces the diversity of the classifiers. Moreover, it is typical in the co-training framework to have the two classifiers focus on different portions of the data. In future, we plan to investigate the application of co-training by relaxing these constraints and by increasing the diversity among the classifiers.

6. ACKNOWLEGMENTS

The authors would like to thank Sibel Yaman, Kyu Han and Weizhong Zhu for their software contributions in implementing the RATS LID systems and Salim Roukos for the discussions.

7. REFERENCES

[1] Strassel S. Walker K., “The RATS Radio traffic collection system,” in *Odyssey Speaker and Language Recognition Workshop*. ISCA.

[2] L.P. Heck and M. Weintraub, “Handset-dependent background models for robust text-independent speaker recognition,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*. IEEE, 1997, vol. 2, pp. 1071–1074.

[3] D.A. Reynolds, “Channel robust speaker verification via feature mapping,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03). 2003 IEEE International Conference on*. Ieee, 2003, vol. 2, pp. II–53.

[4] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, “Acoustic language identification using fast discriminative training,” in *Interspeech*, 2007.

[5] P.A. Torres-Carrasquillo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, and JR Deller Jr, “Approaches to language identification using Gaussian mixture models and shifted delta cepstral features,” in *Seventh International Conference on Spoken Language Processing*, 2002.

[6] S. Yaman, J. Pelecanos, and M. Omar, “On the use of nonlinear polynomial kernel SVMs in language recognition,” in *Proceeding of Interspeech*, 2012.

[7] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[8] K. Nigam and R. Ghani, “Analyzing the effectiveness and applicability of co-training,” in *Proceedings of the ninth international conference on Information and knowledge management*. ACM, 2000, pp. 86–93.

[9] S. Kiritchenko and S. Matwin, “Email classification with co-training,” in *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 2001, p. 8.

[10] U. Guz, S. Cuendet, D. Hakkani-Tur, and G. Tur, “Co-training using prosodic and lexical information for sentence segmentation,” in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[11] A. et al. Adami, “Qualcomm-ICSI-OGI features for ASR,” in *Seventh International Conference on Spoken Language Processing*, 2002.

[12] N. Dehak, P.A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, “Language recognition via i-vectors and dimensionality reduction,” in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.