American Journal of Sciences and Engineering Research

E-ISSN -2348 – 703X, Volume 4, Issue 6, 2021

Unsupervised Domain Adaptation for Language Recognition by Regularization of a Neural Network

Ahmed Alzahmi

Department of Electrical Engineering, Faculty of Engineering, University of Tabuk, Tabuk, Saudi Arabia.

aalzahmi@ut.edu.sa, ahmed.alzahmi@yahoo.com

Abstract: *Automatic language identification systems experience significant degradation in performance when the acoustic characteristics of the test signals differ significantly from the characteristics of the training data. In this article, we study the unsupervised domain adaptation of a system trained on telephone conversations to radio transmissions. We present a method of regularizing a neural network by adding to the cost function a term measuring the divergence between the two domains. Experiments on the OpenSAD15 corpus allow us to select the Maximum Mean Discrepancy to perform this measurement. This approach is then applied to a modern language identification system based on x-vectors. On the RATS corpus, for seven of the eight radio channels studied, the approach allows, without using annotated data from the target domain, to surpass the performance of a system trained in a supervised manner with annotated data from this domain.*

Keywords*: unsupervised domain adaptation, language identification, regularization, maximum mean discrepancy, robustness.*

I. Introduction

A language identification system is usually trained on an environment-specific body of learning. If the test data is not from the same distribution as the training data (and therefore has different characteristics), system performance can drop significantly. In this article, we investigate the effect of changing the transmission channel between training and test data. The training data are telephone conversations. We want to apply such a system to radio communications, for which we do not have annotated data. This problem is called unsupervised domain adaptation.

The possibility of adapting a classification system operating on a source domain to a target domain is based on the assumption that the data distributions of the two domains share common characteristics that can be used for classification (Ben-David et al., 2010). Therefore domain adaptation can be achieved using invariant representations between domains. To this end, two types of approach have emerged (Bousquet & Rouvier, 2019): feature-based methods, which transform the representations of data from the source domain in order to make them similar to the target domain, and model-based methods.

During a model-based adaptation, the parameters of the model are determined taking into account the objective of generalization to the target domain. We propose a model-based approach applying to a neural network whose parameters are obtained by minimizing a cost function. A regularization term is added to the cost function in order to take into account the invariance constraint between the domains. Different regularization functions have been proposed in the image processing and text analysis literature: deep CORAL

 Received-22-10-2021, Accepted- 29-10-2021

(Sun & Saenko, 2016) Maximum Mean Discrepancy (Long et al., 2015), antagonistic cost functions (Ganin et al., 2015). So far none of these approaches have been applied to language recognition.

In this work, we first compare three cost functions for the adaptation to radio channels of a neural network trained for the language identification task: the distance between the means of the distributions, deep CORAL and the Maximum Mean Discrepancy. We show that the latter makes it possible to cancel out the drop in performance due to the absence of annotated data on the target domain.

Secondly, we study a language identification system corresponding to the state of the art (Snyder et al., 2018; Plchot et al., 2018), made up of a feature extractor, an extractor of vectors representative of the audio segments and of a final classifier. Such a system does experience significant degradation in performance due to the change in transmission channel between training and test data. Our approach applied to the vector extraction module representative of the audio segment reduces this degradation and even leads to better performance than that of a system trained in a supervised way on the target domain.

II. Method of Unsupervised Domain Adaptation of a Neural Network

We are placing ourselves in the context of an unsupervised domain adaptation. We have annotated data (x_S, y_S) from a source domain defined by its D_S distribution, and x_T unannotated data from a target domain defined by its D_T distribution. The x_S and x_T are the audio data and the y_S are the associated language labels. The objective of the domain adaptation task is to train a powerful language identification system on the target domain.

2.1. Regularization of the cost function

We are interested in a classification model for the language identification task. It is a neural network f_θ of parameters ϑ. Its parameters are learned in a supervised manner by minimizing L_{CE} cross-entropy on the source domain:

$$\min_\theta \mathbb{E}_{(xs,ys)\sim Ds} [L_{CE}(f_\theta, x_S, y_S)] \quad (1)$$

Based on the observation that the model error on the target domain can be controlled by the sum of source domain error and a measure of divergence between domains (Ben-David et al., 2010), our model-based method of unsupervised domain adaptation consists in adding a L_R regularization function to the cost function. The optimization problem becomes:

$$\min_\theta \mathbb{E}_{(xs,ys)\sim Ds} [L_{CE}(f_\theta, x_S, y_S)] + \lambda L_R(f_\theta, D_S, D_T) \quad (2)$$

L_R is a measure of the divergence of network representations between the D_S and D_T distributions. λ is a parameter representing the compromise between good classification performance on the source domain and the invariance of representations between domains. In practice, a layer of the network is chosen and the cost function L_R is measured for the activations thereof. We use the notation $\Phi_f(x)$ for the values of the activations of this layer for a network f and an input data x. In our experiments, we place ourselves on the output layer of the network.

Different L_R regularization functions have been introduced: deep CORAL (Sun & Saenko, 2016), Maximum Mean Discrepancy (Long et al., 2015; Lin et al., 2018), as well as antagonistic (adversarial) cost functions (Ganin

et al., 2016). In this work, we compare three regularization functions, based on the distance between the means of the distributions, on the distance between the second moments (deep CORAL) and on the Maximum Mean Discrepancy.

2.2. Regularization functions

A simple correction to apply to two probability distributions to bring them closer would be to make them share the same mean. Therefore, our first regularization function is the square of the Euclidean distance between the means of the distributions of the two domains:

$$L_{moy} = \| \mathbb{E}_{xs \sim Ds}[\Phi_f(x)] - \mathbb{E}_{xT \sim D_T}[\Phi_f(x_T)] \|_2^2 \quad (3)$$

In the same vein, the deep CORAL cost function (Sun & Saenko, 2016) aims to align the second moments of the two distributions. It corresponds to the square of the Euclidean distance between the covariance matrices of the distributions of each of the two domains:

$$L_{CORAL} = \|C_S - C_T\|_2^2 \quad (4)$$

Where C_S and C_T are the covariance matrices of activations $\Phi_f(x)$ on the D_S and D_T domains. Finally, the Maximum Mean Discrepancy (*MMD*) is a measure of divergence between domains based on a measure of similarity between pairs of samples defined by a semi-defined kernel positive *k*. It takes the value:

$$L_{MMD} = \mathbb{E}_{x_S, x'_S \sim D_S}\left[k\left(\Phi_f(x_S), \Phi_f(x'_S)\right)\right] + \mathbb{E}_{x_T, x'_{ST} \sim D_T}\left[k\left(\Phi_f(x_T), \Phi_f(x'_T)\right)\right] - 2\mathbb{E}_{x_S \sim D_S, x_T \sim D_T}\left[k\left(\Phi_f(x_S), \Phi_f(x_T)\right)\right] \quad (5)$$

When the kernel is the usual scalar product then *LMMD* is equivalent to Lmoy, presented previously. To take into account more precisely the difference between the distributions, we use a Gaussian kernel, of variance noted σ^2:

$$k\left(\Phi_f(x), \Phi_f(x')\right) = \exp\left(-\frac{\|\Phi_f(x) - \Phi_f(x')\|_2^2}{2\sigma^2}\right) \quad (6)$$

MMD regularization is a measure of divergence between two probability distributions, which can be estimated from a finite number of samples, including in high dimensional spaces (Peyré & Cuturi, 2019). In the field of speech processing, it has been used for adaptation feature-based domain of a speaker recognition system (Lin et al., 2018). What's more, the estimation of the Maximum Mean Discrepancy can be done efficiently on a GPU (Feydy et al., 2019).

During training, these three regularization functions will simply be estimated by empirical average over each minibatch, then added to the classification cost, see Equation (2).

III. Selection of the Regularization Function

To isolate the effect of the proposed regularization method, we compare the three functions of regularization on an end-to-end system made up of a single neural network, with the corpus OpenSAD15.

3.1. End-to-end system architecture

Language identification can be done directly with a convolutional neural network (Lozano-Diez et al., 2015). We use a similar architecture described in Table 1. The input features of our system are MFCCs of dimension 12, calculated for frames of 10 ms. We perform the classification by directly using the returned posterior probabilities by the output layer for each language. The system is trained and evaluated with segments of three seconds.

3.2. The OpenSAD 2015 corpus

To study the effect of changing the transmission channel, we carried out our experiments preliminaries on four languages of the OpenSAD15 corpus (NIST, 2016): English, Arabic, Pashto and Urdu.

Table 1. Architecture of the convolutional neural network

Convolutions according to the time axis			
	kernel/max pooling size	number of filters	
conv. 1	5/2 1024	1024	ReLU
conv. 2	5/2 1024	1024	ReLU
conv. 3	5/2 128	128	ReLU
Statistical aggregation of means and standard deviations (pooling)			
output dimension: 2 × 128 = 256			
Connected layers			
layer name	Dimension		Activation function
Fn 1	256 × 128		ReLU
Fn 2	128 × 4		Softmax.

This is a corpus created from telephone conversations, src channel of the corpus, which were then transmitted by six different radio systems: B, F, G (UHF), E (VHF), D and H (HF).

In order to avoid bias in training, we only use half of the training data. Half of the original audio files are used for the source domain (SRC channel) and the other half, corresponding to different phrases, are used for the target domains (radio channels). This way, the same linguistic content is not present in both domains during learning.

3.3. Results of preliminary experiments

We carry out different convolutional neural network training on the Open-SAD15 corpus. The performance of each of the systems on the channels of interest is shown in Table 2. Performance is measured with an average Equal Error Rate (EER) for three second speech segments. An EER is calculated for each of the four languages in the corpus and the score obtained is the arithmetic mean of the rates for each language.

The network is first trained with data from the telephone channel. This system, which obtains an average EER of 8% on the SRC channel, is totally inoperative on the target channels. However, when a system is trained in a supervised fashion on each of the target channels, then we get an average EER of between 14% and 22%.

The last three rows of Table 2 show the learning performance with each of the proposed regularization functions, using annotated data from the SRC channel and unannotated data on the target domain. The parameter λ (as well as σ^2 for the *MMD*) is selected for each cost function as a function of the performance obtained on a validation set. The results of all the target domains are consistent: the regularization methods improve the average ERAs on the target domain compared to learning on the source domain. A clear hierarchy

appears: the *MMD* with Gaussian kernel is more efficient than deep CORAL, which is itself superior to the constraint on the distance between the means. This result means that, in order to eliminate the distortion due to the change of channel, the system cannot be limited to the first two moments of the distributions but must take into account a more complex geometry.

Table 2. Equal error rate results of different convolutional network training methods for the OpenSAD15 corpus (3 second segments). The source domain is the channel telephone (SRC).

EER method on target domain (%)	EER on target domain (%)					
	B	D	E	F	G	H
supervised on source supervised on target	57	52	48	51	30	50
	18	15	19	15	14	22
average distances	53	44	38	35	12	41
deep CORAL	32	32	26	18	11	20
MMD	19	11	16	13	9	18

For five of the six target domains tested, regularization with the *MMD* cost function provides a better performance on the target domain than supervised learning on this domain, even though the training did not use annotated data. of the target domain.

IV. Application to a State-of-the-Art System

Preliminary experiments made it possible to select the regularization function based on Maximum Mean Discrepancy for the adaptation of a convolutional neural network. We therefore apply this learning method to a state-of-the-art language identification system for this task.

4.1. System architecture

A modern language recognition system (Snyder et al., 2018; Plchot et al., 2018) is generally made up of three modules: a representation extractor for frames located in time, a representation extractor for l whole audio segment and a final classifier.

In our system, the first module extracts stacked multilingual bottleneck features. These are activations of an intermediate layer (bottleneck) of a neural network that has been trained to recognize triphones for seventeen languages of the Babel corpus. We use the trained BUT / PHONEXIA bottleneck feature extractor networks (Fer et al., 2017), which performed well for the NIST LRE 2017 assessment (Plchot et al., 2018). They generate bottleneck features of dimension 80, for each frame of 10 ms.

The second module extracts one representative vector per segment. It is a neural network taking as input the sequence of bottleneck features and supervised training to predict the language used in the segment. We use the architecture of the x-vector system (Snyder et al., 2018), consisting of five layers performing raster processing, followed by a statistical pooling layer and three solid layers. The x-vectors resulting from this architecture are vectors of dimension 512.

Finally, the final classifier takes an x-vector as input and produces a score for each of the target languages. Our final classifier is composed of an LDA (Linear Discriminant Analysis), used to reduce the dimension, a matrix multiplication whitening and an SVM (Support Vector Machine).

We apply the Maximum Mean Discrepancy-based regularization method to the x-vector network, in order to produce channel-change invariant x-vectors. For similar systems dedicated to the speaker recognition task, the model-based adaptation of the x-vector network has reduced the distortion due to language (Rohdin et al., 2019) and acoustic conditions (Bhattacharya et al., 2019), with competing cost functions.

4.2. The RATS corpus

We train this system on the RATS corpus (Walker & Strassel, 2012). We use LDC2015S02 and LDC2017S20 deliveries which have five languages: English, Arabic, Farsi, Pashto and Urdu. This corpus has the same characteristics as the OpenSAD15 corpus which is a subset. It contains two additional UHF channels: A and C. As with the OpenSAD15 corpus, we only use half of the corpus so that the same linguistic content is not present on both the source and target domains.

Language identification was studied on the RATS corpus (Mateˇjka et al., 2014; Lei et al., 2014; Han et al., 2013) with the release LDC2018S10, also containing five languages: Arabic, Dari, Farsi , pashto and urdu. For three-second segments and for all channels, the best average EER obtained is 9.59% (Mateˇjka et al., 2014).

4.3. Experiences

First, we train the system on all channels, we get an average EER of 9.36% comparable to the state of the art on the RATS corpus. Then we train the x-vector network in a supervised manner on the source domain (telephone channel) and then on each of the target channels. Finally, for each of the target domains, we apply the regularization based on the Maximum Mean Discrepancy. Remember that the neural network is not used directly to perform the classification but to extract an x-vector representative of the audio segment. To evaluate the properties of the x-vectors thus extracted, we realize several systems by training the final classifier with annotated data, either from the source domain or from the target domain. The performance of each of these systems is shown in Table 3.

Table 3. Equal error rate results of different training methods of the x-vector network and of the final classifier for eight radio channels of the RATS corpus (3-second segments)

Training method		Average EER on the target domain (%)							
x-vector	Final classifier	A	B	C	D	E	F	G	H
supervised on source	supervised on source	50.2	42.3	34.4	39.6	48.5	45.1	17.4	43.6
supervised on source	supervised on target	15.8	15.0	14.1	14.3	21.8	20.5	9.8	18.8
supervised on target	supervised on target	14.6	12.5	12.6	6.7	13.6	13.5	8.6	14.2
MMD	supervised on source	12.7	10.6	11.7	7.6	13.3	11.9	5.5	12.2
MMD	supervised on target	10.2	9.2	11.3	6.0	11.8	10.3	5.1	10.0

It should be noted first that a system trained on the source domain, which nevertheless obtains an average EER of 6.0% on this domain, achieves a very poor performance on the radio channels. At the opposite, Supervised training of the system on the target domain achieves average EERs of between 6.7% and 14.6%. On the other hand, we observe that the supervised training of the final classifier on the target domain with x-vectors trained on the source domain (row 2 of Table 3) is not sufficient to achieve the performance of a fully trained system on the target domain (line 3). This finding justifies the need to develop a domain adaptation method for the x-vector network.

The regularization of the x-vector network with the Maximum Mean Discrepancy is a success. The x-vectors produced by this network have acquired robustness to channel change since a final classifier trained on the source domain with these x-vectors (row 4 of Table 3) achieves good performance on the target domain. In

fact, for all channels except D-channel, domain adaptation of the x-vector network with a final classifier trained on the source domain is more efficient than training the whole system on the target domain (line 3). This result confirms our preliminary experiments: not only the output values but also the activations of the x-vector layer of the network acquire domain invariance thanks to the regularization method.

Finally, training a final classifier on the target domain with the x-vectors obtained by regularization (row 5 of Table 3) leads to mean EERs significantly lower than a system fully trained on the target domain (row 3). So regularization has an interest in improving the quality of x-vectors, even when language labels are available on the target domain. On the other hand, for these x-vectors regularized with *MMD*, training the final classifier on the target domain (line 5) improves classification performance compared to training on the source domain (line 4). The x-vectors are therefore not totally invariant between domains and our approach could be combined with an adaptation of the final classifier.

V. Conclusion

We have introduced an unsupervised domain adaptation method of a neural network for a language identification system. This method consists of a modification of the cost function used when training the neural network by adding a regularization term. By preliminary experiments with a convolutional neural network, we selected a cost function based on the Maximum Mean Discrepancy. Secondly, we applied this approach to a recent language identification system, consisting of a features extractor, an x-vectors extractor and a final classifier. The results demonstrate the effectiveness of the proposed method to take into account the distortion due to the signal transmission channel. When regularization is applied to the x-vector network, it produces vectors that have acquired robustness to domain change and therefore allow the transfer of learning between the source domain and the target domain. In addition, the proposed regularization significantly improves the ability to discriminate the x-vectors thus produced compared to supervised learning. It is therefore relevant even if we have annotated data on the target domain.

VI. References

1. BEN-DAVID S., BLITZER J., CRAMMER K., KULESZA A., PEREIRA F. & VAUGHAN J. W. (2010). A theory of learning from different domains. In Machine learning, volume 79, p. 151–175 : Springer.
2. Bhattacharya g., Alam j. & Kenny p. (2019). Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training. In Proc. ICASSP, p. 6041–6045.
3. Bousquet P.-M. & Rouvier M. (2019). On robustness of unsupervised domain adaptation for speaker recognition. In Proc. INTERSPEECH, p. 2958–2962.
4. Fer R., Mateˇ Jka P., Grézl F., Plchot O., Veselý K. & Cˇ Ernocký J. H. (2017). Multilin- gually trained bottleneck features in spoken language recognition. In Computer Speech & Language, volume 46, p. 252–267 : Elsevier.
5. Feydy J., Séjourné T., Vialard F.-X., Amari S.-I., Trouvé A. & Peyré G. (2019). Interpolating between optimal transport and MMD using Sinkhorn divergences. In Proc. The Twenty-second International Conference on Artificial Intelligence and Statistics, p. 2681–2690.
6. Ganin Y., Ustinova E., Ajakan H., Germain P., Larochelle H., Laviolette F., Mar- Chand M. & Lempitsky V. (2016). Domain-adversarial training of neural networks. In The Journal of Machine Learning Research, volume 17, p. 2096–2030.
7. Han K. J., Ganapathy S., LI M., Omar M. K. & Narayanan S. (2013). Trap language identification system for RATS phase II evaluation. In Proc. INTERSPEECH, p. 1502–1506.
8. Lei Y., Ferrer L., Lawson A., Mclaren M. & Scheffer N. (2014). Application of convolutional neural networks to language identification in noisy conditions. In Proc. Odyssey, volume 41, p. 1–8.
9. Lin W.-W., Mak M.-W., LI L. & Chien J.-T. (2018). Reducing domain mismatch by maximum mean discrepancy based autoencoders. In Proc. Odyssey, p. 162–167.

10. Long M., Cao Y., Wang J. & Jordan M. I. (2015). Learning transferable features with deep adaptation networks. In Proc. ICML 2015, p. 97–105.
11. Lozano-Diez A., Zazo Candil R., González Domínguez J., Toledano D. & González- Rodríguez J. (2015). An end-to-end approach to language identification in short utterances using convolutional neural networks. In Proc. INTERSPEECH, p. 403–407.
12. Mateˇ Jka P., Zhang L., NG T., Mallidi S. H., Glembek O., MA J. & Zhang B. (2014). Neural network bottleneck features for language identification. In Proc. Odyssey, p. 299–304.
13. NIST (2016). Evaluation plan for the NIST open evaluation of speech activity detection (Open- SAD15). In www.nist.gov/itl/iad/mig/nist-open-speech-activity-detection-evaluation.
14. Peyré G. & Cuturi M. (2019). Computational optimal transport. In Foundations and Trends in Machine Learning, volume 11, p. 355–607 : Now Publishers, Inc.
15. Plchot O., Mateˇ Jka P., Novotný O., Cumani S., Lozano-Diez A., Slavicek J., Diez M., Grézl F., Glembek O., Mounika K. V., Silnova A., Burget L., Ondel L., Kesiraju S. & Rohdin J. (2018). Analysis of BUT-PT submission for NIST LRE 2017. In Proc. Odyssey, p. 47–53.
16. Rohdin J., Stafylakis T., Silnova A., Zeinali H., Burget L. & Plchot O. (2019). Speaker verification using end-to-end adversarial language adaptation. In Proc. of ICASSP, p. 6006–6010.
17. Snyder D., Garcia-Romero D., Mccree A., Sell G., Povey D. & Khudanpur S. (2018). Spoken language recognition using x-vectors. In Proc. Odyssey, p. 105–111.
18. Sun B. & Saenko K. (2016). Deep CORAL : Correlation alignment for deep domain adaptation. In Proc. ECCV 2016, p. 443–450 : Springer.
19. Walker K. & Strassel S. (2012). The RATS radio traffic collection system. In Proc. Odyssey, p. 291–297.