INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Automatic Dialect Detection in Arabic Broadcast Speech

Ahmed Ali[1,5], Najim Dehak[2,3], Patrick Cardinal[4], Sameer Khurana[1], Sree Harsha Yella[2], James Glass[2], Peter Bell[5], Steve Renals[5]

[1]Qatar Computing Research Institute, HBKU, Doha, Qatar
[2] MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA
[3] JHU Center for Language and Speech Processing (CLSP), Baltimore, MD, USA
[4] École de technologie supérieure, Département de Génie Logiciel et des TI, Montréal, Canada
[5]Centre for Speech Technology Research, University of Edinburgh, UK
{amali,skhurana}@qf.org.qa, najim@jhu.edu

Abstract

In this paper, we investigate different approaches for dialect identification in Arabic broadcast speech. These methods are based on phonetic and lexical features obtained from a speech recognition system, and bottleneck features using the i-vector framework. We studied both generative and discriminative classifiers, and we combined these features using a multi-class Support Vector Machine (SVM). We validated our results on an Arabic/English language identification task, with an accuracy of 100%. We also evaluated these features in a binary classifier to discriminate between Modern Standard Arabic (MSA) and Dialectal Arabic, with an accuracy of 100%. We further reported results using the proposed methods to discriminate between the five most widely used dialects of Arabic: namely Egyptian, Gulf, Levantine, North African, and MSA, with an accuracy of 59.2%. We discuss dialect identification errors in the context of dialect code-switching between Dialectal Arabic and MSA, and compare the error pattern between manually labeled data, and the output from our classifier. All the data used on our experiments have been released to the public as a language identification corpus.

Index Terms: Dialect Identification, Vector Space Modelling

1. Introduction

The task of Dialect Identification (DID) is a special case of the more general problem of Language Identification (LID). LID refers to the process of automatically identifying the language class for given speech segment or text document. DID is arguably a more challenging problem than LID, since it consists of identifying the different dialects within the same language class. The importance of addressing DID can be gauged from its growing interest in the Automatic Speech Recognition (ASR) community [1]. A good DID system can facilitate the identification of dialectal segments from an untranscribed mixed-speech dataset. This process can help reduce the ASR word error rate (WER) for dialectal data by training ASR systems for each dialect, or by adapting the ASR models to a particular dialect.

The natural language processing (NLP) community has aggregated dialectal Arabic into five regional language groups: Egyptian (EGY), North African or Maghrebi (NOR), Gulf or Arabian Peninsula (GLF), Levantine (LAV), and Modern Standard Arabic (MSA). An objective comparison of the varieties of Arabic dialects could potentially lead to the conclusion that Arabic dialects are historically related, but not synchronically, and are mutually unintelligible languages like English and Dutch. Normal vernacular can be difficult to understand across different Arabic dialects [2]. Arabic dialects are thus sufficiently distinctive, and it is reasonable to regard the DID task in Arabic as similar to the LID task in other languages. Table 1 shows two phrases across the different dialects, it is clear from this example that there are lexical variations across the different dialects which motivates us to consider it.

Two broad LID approaches have been investigated in the literature: low-level acoustic features, and high-level phonetic and lexical features. In the lexical area, words, roots, morphology, and grammars [3, 4] have been studied. Acoustic features such as shifted delta cepstral coefficients [5] and prosodic features [6] using Gaussian mixture models (GMMs), i-vector representations and support vector machine (SVM) classifiers [5] have been shown to be effective for LID. More recent work explored the use of frame-by-frame phone posteriors (PLLRs) [7] as new features for LID. New subspace approaches based on non-negative factor analysis (NFA) for GMM weight decomposition and adaptation [8] were also applied to both LID and DID tasks. GMM weight adaptation subspaces seem to provide complementary information to the classical i-vector framework. Finally, phoneme sequence modeling and its n-gram subspace have been studied for both Arabic DID [9] and LID [10].

EGY	GLF	LAV	MSA	NOR	Translation
ازايك AzAYk	اشلونك A$lwnk	اشلونك / كيفك kyfk / A$lwnk	كيف حالك kyf HAlk	واش راك wA$ rAk	How are you?
انت فن Ant fyn	وينك wynk	وينك wynk	اين انت Ayn Ant	وين راك wyn rAk	Where are you?

Table 1: *Lexical examples in Arabic and Buckwalter format.*

In this paper we investigate three Vector Subspace Models (VSMs) for Arabic DID based on 1) lexical, 2) phonetic, and 3) i-vectors. We conduct a thorough feature selection study of these models to better understand their interaction. A further contribution of this work is the release of an Arabic DID system so others can extend and improve DID performance on this task.[1]

2. Vector Space Models

2.1. Senone based Utterance VSM

Senone refers to an n-gram phone sequence. In our case $n \leq 4$. VSM construction takes place in two steps: first, a phoneme recognizer is used to extract the senone [11] sequence for a given

[1]https://github.com/Qatar-Computing-Research-Institute/dialectID

http://dx.doi.org/10.21437/Interspeech.2016-1297

speech utterance. The phoneme sequence is obtained by automatic vowelization of the training text, followed by vowelization to phonetization (V2P). The 36 chosen phonemes cover all the dialectal Arabic sounds. Further details about the speech recognition pipeline, training data, and phoneme set is given in [12]. For the phoneme sequence, we process the phoneme lattice, and obtain the one-best transcription, ignoring silences as well as noisy silences. Each speech utterance ($\mathbf{u}$) is then represented as a high dimensional sparse vector ($\vec{u}$):

$$\vec{u} = (A(f(u, s_1)), A(f(u, s_2)), \ldots, A(f(u, s_d))), \quad (1)$$

where $f(u, s_i)$ is the number of times a senone s_i occurs in the speech utterance u, and A is the scaling function. We experiment with both an identity scaling function and $tf.idf$ scaling function, commonly used in the field of Natural Language Processing [13] to downweight the contribution of the words (in our case senones) that occur in almost all documents (in our case utterances), as these words (senones) do not provide any discriminative information about the documents (utterances).

The vector space is then represented by the matrix, $\mathbf{U_s} \in \mathbb{R}^{d \times N}$ (see Fig 1). This approach and the notation used to define a VSM is directly inspired by the seminal works in the area of VSM of Natural Language in [14, 15, 16] and in LID [17].

$$\mathbf{U_s} = \begin{array}{c} \\ s_1 \\ s_2 \\ \vdots \\ s_d \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & u_2 & \ldots & u_N \end{array} \\ \begin{bmatrix} A(f(s_1,u_1)) & A(f(s_1,u_2)) & \ldots & A(f(s_1,u_N)) \\ A(f(s_2,u_1)) & A(f(s_2,u_2)) & \ldots & A(f(s_2,u_N)) \\ \vdots & \vdots & \ddots & \vdots \\ A(f(s_d,u_1)) & A(f(s_d,u_2)) & \ldots & A(f(s_d,u_N)) \end{bmatrix} \end{array}$$

Figure 1: *Senone-based utterance VSM. Column vectors of the matrix correspond to the speech utterance vector representation formed using equation 1. d is the size of the senone dictionary, and N is the total number of speech utterances in the dialectal speech database.*

2.2. Word based Utterance VSM

The word-based utterance VSM ($\mathbf{U_w}$) is constructed in two steps in a manner similar to the senone features: An ASR system is used to extract the word sequence for each utterance in the speech database. Details about the ASR system can be found in [12]. Each speech utterance ($\mathbf{u}$) is then represented as a high-dimensional sparse vector ($\vec{u}$):

$$\vec{u} = (A(f(u, w_1)), A(f(u, w_2)), \ldots, A(f(u, w_{d'}))), \quad (2)$$

where $f(u, w_i)$ is the number of times a word w_i occurs in the speech utterance u and A is the scaling function which has the same interpretation as for $\mathbf{U_s}$ (above). Vocabulary size was 55k. The tri-gram dictionary size was 580k which we used to construct the word based VSM

2.3. i-vector-based Utterance VSM

2.3.1. Bottleneck Features (BN)

Recently, bottleneck features extracted from an ASR DNN-based model were applied successfully to language identification [18, 19, 20]. In this paper, we used a similar bottleneck features configurations as in our previous ASR-DNN system for MSA speech recognition [21]. This system is based on two successive DNN models. Both DNNs use the same setup of 5 hidden sigmoid layers and 1 linear BN layer, and they were both based on tied-states as target outputs. The senone labels of dimension 3040 are generated by a forced alignment from an HMM-GMM baseline trained on 60 hours of manually transcribed Al-Jazeera MSA news recordings [12]. The input to the first DNN consists of 23 critical-band energies that are obtained from Mel filter-bank. Pitch and voicing probability are then added. 11 consecutive frames are then stacked together. The second DNN is used for correcting the posterior outputs of the first DNN. In this architecture, the input features of the second DNN are the outputs of the BN layer from the first DNN. Context expansion is achieved by concatenating frames with time offsets of -10, -5, 0, 5, and 10. Thus, the overall time context seen by the second DNN is 31 frames.

2.3.2. Modeling

An effective and well-studied method in language and dialect recognition is the i-vector approach [8, 22, 5]. The i-vector involves modeling speech using a universal background model (UBM) – typically a large GMM – trained on a large amount of data to represent general feature characteristics, which plays a role of a prior on how all dialects look like. The i-vector approach is a powerful technique that summarizes all the updates happening during the adaptation of the UBM mean components to a given utterance. All this information is modeled in a low dimensional subspace referred to as the total variability space. In the i-vector framework, each speech utterance can be represented by a GMM supervector, which is assumed to be generated as follows:

$$M = u + Tv$$

Where u is the channel and dialect independent supervector (which can be taken to be the UBM supervector), T spans a low-dimensional subspace and v are the factors that best describe the utterance-dependent mean offset. The vector v is treated as a latent variable with the i-vector being its maximum-a-posteriori (MAP) point estimate. The subspace matrix T is estimated using maximum likelihood on large training dataset. An efficient procedure for training and for MAP adaptation of i-vector can be found in [23]. In this approach, the i-vector is the low-dimensional representation of an audio recording that can be used for classification and estimation purposes. In our experiments, the UBM was a GMM with 2048 components, BN features were used, and the i-vectors were 400-dimensional.

In order to maximize the discrimination between the different dialect classes in the i-vector space, we combine Linear Discriminant Analysis (LDA) and Within Class Co-variance Normalization [5]. This intersession compensation method has been used with both SVM [5] and cosine scoring [8].

3. Dataset

3.1. Train Data

The training corpus was collected from the Broadcast News domain in four Arabic dialects (EGY, LAV, GLF, and NOR) as well as MSA. Data recordings were carried out at 16Khz. The recordings were segmented to avoid speaker overlap, removing any non-speech parts such as music and background noise. More details about the training data can be found in [8]. Although the test database came from the same broadcast domain, the recording setup is different. The test data was downloaded directly from the high quality video server for Aljazeera (bright-

cove) over the period of July 2104 until January 2015, as part of QCRI Advanced Transcription Service (QATS) [24].

Data	EGY	GLF	LAV	NOR	MSA	ENG
Train	13	9.5	11	9	10	10
Test	2	2	2	2	2	2

Table 2: *Number of hours of speech available for each dialect.*

3.2. Test Data

The test set was labeled using the crowdsource platform CrowdFlower, with the criteria to have a minimum of three judges per file and up to nine judges, or 75% inter-annotator agreement (whichever comes first). More details about the test set and crowdsourcing experiment can be found in [25]. The test set used in this paper differs from that used in [8] for two reasons: First, the crowdsourced data is available to reproduce the results, and thus can be used as a standard test set for Arabic DID; second, the new test set has been collected using different channels, and recording setup compared to the training data, which makes our experiments less sensitive to channel/speaker characteristics.

The train and test data can be found on the QCRI web portal[2]. Table 2 and Table 3 present some statistics about the train and the test data.

Data	EGY	GLF	LAV	NOR	MSA	ENG
Train	1720	1907	1059	1934	1820	1649
Test	315	348	238	355	265	452

Table 3: *Number of speech utterances for each dialect.*

4. Experiments

4.1. Choosing the Best Classifier

We first studied the best classification approach for the DID task from a set of two generative models: n-gram language model [26] and Naive Bayes [27], and two discriminative classifiers: linear SVM [28] and Maximum Entropy [29]. We measured the performance of each model on the DID task, in the word or lexical-based utterance vector space, which is constructed using the approach mentioned in section 2, using identity scaling function A, and performing no dimensionality reduction. Hence, the dimensionality of an utterance vector, $\vec{u}$, is the same as the size of the lexicon, which in our case was 55k. Results can be seen in table 4. As the linear SVM performs the best, it is our choice of classifier for the rest of the experiments.

4.2. Feature Selection Study

Here we examine the dialect information captured by the three utterance VSMs explained in section 2. We also explore the concatenation of the utterance vector representations, and report the results in Tables 5 and 6. Details about the terms in the results table are given below:

- $\mathbf{U_w^i}$: Refers to the utterance VSM in which each utterance is represented by a vector given by equation 2, where A is chosen to be the identity function. The bases of the vectors are the words in the lexicon. SVD is used to reduce the dimensionality of the utterance Vector Space from 55k originally, to 300, 600, 1200, 1600 at which point increase the gain in the classification performance tends to saturate.
- $\mathbf{U_w^{tf.idf}}$: Same as the previous Utterance VSM, except that A is chosen via $tf.idf$ [13] instead of identity function, which gives us significant improvement in accuracy over the previous vector space.
- $\mathbf{U_s^i}$: Refers to the utterance VSM in which each utterance is represented by a vector given by equation 1, where A is chosen to be the identity function. Utterance vector bases corresponds to senones. Just as with the word-based utterance VSM, we use SVD on the vector space and experiment with different dimensions. The utterance Vector Space constructed using senone features is more discriminative than word-based Vector space.
- $\mathbf{U_s^{tf.idf}}$: Refers to the same vector space as the previous one, except that A is chosen to be the $tf.idf$ function. $tf.idf$, does not help in the case of senone features.
- **Feature Combination**: Combining the best senone-based utterance VSM, $U_s^i(600d)$, and the best lexical-based utterance VSM, $U_w^{tfidf}(1200d)$, to form a concatenated feature vector representation. SVD is performed to reduce the dimensions of the feature space. Feature combination does not help and hence we conclude that the two vector spaces are capturing similar information.
- $\mathbf{U_{iVec}^{bnf}}$: Refers to the utterance VSM, where each utterance is represented by a compact $400d$ i-vector (section 2.3). We use the bottleneck features to train the UBM, which is then used to extract the i-vector. We do not experiment with different i-vector dimensions and take the best dimension reported in [5] for the LID task. The i-vector feature space is significantly more discriminative than previously defined feature spaces.
- $\mathbf{U_{iVec+LDA+WCNN}^{bnf}}$: Reducing the dimensionality of the i-vector space using LDA and performing WCNN has been reported to do well in LID tasks [5] and we use the same technique and see a significant improvement in the DID results.
- $\mathbf{U_{iVec+LDA+WCNN}^{bnf} + U_s^i(600d)}$: Finally we concatenate the best senone-based VSM with the best i-vector-based VSM, to form a concatenated vector representation for each utterance and see slight improvements in the results. As the lexical and senone-based representations encode the same information about the dialect, we do not experiment with concatenated lexical and i-vector representations.

Model	ACC	PRC	RCL
n-gram Language Model	40.4%	40.2%	41.3%
Naive Bayes	37.9%	37.5%	50.2%
Max Ent	40%	40%	40.6%
SVM	**45.2**%	44.8%	45.4%

Table 4: *Performance of different classifiers using lexical features, with lexicon size of 55K. ACC, PRC and RCL correspond to accuracy, precision and recall on the test set.*

[2]http://alt.qcri.org/resources/ArabicDialectIDCorpus/

	$d = 300$			$d = 600$			$d = 1200$			$d = 1600$		
	ACC	PRC	RCL	ACC	PRC	RCL	ACC	PRC	RCL	ACC	PRC	RCL
$\mathbf{U_w^i}$	38.3	41.9	39.4	41.7	44.1	42.8	**42.9**	45.6	44	42.9	45	43.8
$\mathbf{U_w^{tfidf}}$	43.3	42.7	43.5	44.6	44	44.9	**45.5**	45.1	45.8	21.9	20.9	21.9
$\mathbf{U_s^i}$	45.2	44.8	45.9	**45.8**	45.1	46.5	45.2	44.7	45.8			
$\mathbf{U_s^{tfidf}}$	**44**	43.9	44.7	44	44.2	44.6	43.9	44	44.3			
Feature Combination	**44.8**	44.2	45.6	44.1	43.4	44.8	44.8	44.1	45.4			

Table 5: *Accuracy, Precision and Recall for different senone and lexical feature based Vector Spaces. d is the dimensionality of the Vector Space.* ***Boldfaced*** *numbers are the best accuracy for the corresponding vector space, for a corresponding vector space dimensionality d. A detailed explanation of feature spaces is given in the feature selection study (section 4.2)*

Feature Space	d	ACC	PRC	RCL
$\mathbf{U_{iVec}^{bnf}}$	400	55.3	61	55.9
$\mathbf{U_{iVec+LDA+WCNN}^{bnf}}$	4	58.5	62.3	58.9
$\mathbf{U_{iVec+LDA+WCNN+LNORM}^{bnf}}$	4	58.7	61.9	59.3
$\mathbf{U_{iVec+LDA+WCNN}^{bnf} + U_s^i(600d)}$	604	59.2	62.7	59.5

Table 6: *Accuracy, Precision and Recall for different i-vector based feature spaces. d refers to the dimensionality of the Vector Space. A detailed explanation of feature spaces is given in the feature selection study (section 4.2).*

	EGY	GLF	LAV	MSA	NOR	Total Truth	PRC
EGY	221	15	57	13	9	315	50.3%
GLF	45	121	82	12	5	265	55.8%
LAV	74	43	199	18	14	348	46.9%
MSA	19	17	20	218	5	279	77%
NOR	80	21	66	22	166	355	83.4%
#class	439	217	424	283	199		
RCL	70.2%	45.7%	57.2%	78.1%	46.8%		

Table 7: *Confusion Matrix for DID.*

4.3. One Vs All classification (Sanity Check)

We constructed a senone-based utterance VSM (section 2.1) based on 20 hours of speech; 10 hours English (which we got from [30]) and 10 hours Arabic (randomly sampled from our training data, section 3). Binary classification (English vs Arabic) using an SVM classifier, was then performed and it yielded 100% accuracy on the 1.5 hour test set. The reason to choose the senone-based feature space and not the i-vector-based feature space for classification is to avoid channel mismatch, as the English data came from a different source domain. We did a similar experiment to classify MSA versus all dialectal Arabic and again obtained 100% classification accuracy.

4.4. System Output Combination

We fused the scores of the best senone system and the SVM-based i-vector system. In the fusion steps, the original scores of each system were normalized and combined using the same fusion weights for both systems. This approach yielded a final accuracy of 60.2%, which is the best performance we achieved. One explanation for this gain is that the error patterns for the two feature spaces are quite different, and we were able to confirm that by analyzing the confusion matrix for each system.

5. Discussion

We infer from the confusion matrix in Table 7 that GLF and LAV are the most confusable dialect pair. We believe that this is related to the greater lexical similarity between these two dialects (see Table 1). Note, the confusion matrix is from the best DID system. We borrowed Table 8 from previous work [25] on the test set, which shows the amount of time the same speakers switch between dialect and another (mainly MSA, and their own native dialect). For example, in the second row of Table 8, there are 200 samples from potential Gulf speakers. After manually labeling, there were 106(53%) segments labeled as MSA, 82(41%) validated as GLF, 8(4%) as LAV, and 4 segments were not identified with enough confidence to be considered. This means more than 50% of the random GLF speakers data is infact MSA speech segments. This is strong evidence for the amount of code-switching between one dialect and MSA from the same speaker.

Expected Dialect	EGY	GLF	LAV	NOR	MSA
EGY	65%				32%
GLF		41%	4%		53%
LAV	1%	1%	53%		39%
NOR	1%			69%	28%

Table 8: *Expected dialect of each speech segment from particular dialectal speakers.*

6. Conclusions

This paper presents our efforts on automatic dialect identification for Arabic broadcast speech. We have demonstrated a dialect classifier with an accuracy of 60.2% using system combination. We also achieved 100% accuracy on two binary classification tasks; MSA vs Dialectal Arabic and English vs Arabic. We studied the potential code-switching pattern in our classifier and its correlation with the manual annotation. Further work for this research is to study the code-switch between MSA and dialectal Arabic without considering speaker diarization or silence between speech segments in what can be called dialect diarization. We shall also study deep neural network approaches of classification to learn a more complex non-linear decision boundary.

7. References

[1] G. Liu, Y. Lei, and J. H. Hansen, "Dialect identification: Impact of differences between read versus spontaneous speech," *EUSIPCO-2010: European Signal Processing Conference, Aalborg, Denmark*, 2010.

[2] C. Holes, *Modern Arabic: Structures, functions, and varieties*. Georgetown University Press, 2004.

[3] D. A. Reynolds, W. M. Campbell, W. Shen, and E. Singer, "Automatic language recognition via spectral and token based approaches," in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. Springer, 2008.

[4] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *Circuits and Systems Magazine, IEEE*, vol. 11, no. 2, pp. 82–108.

[5] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *INTERSPEECH*, 2011, pp. 857–860.

[6] D. Martínez, L. Burget, L. Ferrer, and N. Scheffer, "ivector-based prosodic system for language identification," in *ICASSP*, 2012, pp. 4861–4864.

[7] O. Plchot, M. Diez, M. Soufifar, and L. Burget, "Pllr features in language recognition system for rats," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[8] M. H. Bahari, N. Dehak, L. Burget, A. Ali, J. Glass *et al.*, "Non-negative factor analysis for gmm weight adaptation," *IEEE Transactions on Audio Speech and Language Processing*, 2014.

[9] H. Soltau, L. Mangu, and F. Biadsy, "From modern standard arabic to levantine asr: Leveraging gale for dialects," in *ASRU*, 2011, pp. 266–271.

[10] M. Soufifar, S. Cumani, L. Burget, and J. Černocky, "Discriminative classifiers for phonotactic language recognition with ivectors," in *ICASSP*, 2012, pp. 4853–4856.

[11] M.-Y. Hwang and X. Huang, "Subphonetic modeling for speech recognition," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 174–179. [Online]. Available: http://dx.doi.org/10.3115/1075527.1075566

[12] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, "A complete kaldi recipe for building arabic speech recognition systems," in *SLT*, 2014.

[13] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, 2003.

[14] G. Salton, a. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Magazine Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[15] W. Lowe, S. Mcdonald, W. Lowe, and S. Mcdonald, "Division of Informatics , University of Edinburgh Institute for Adaptive and Neural Computation The Direct Route : Mediated Priming in Semantic Space by The Direct Route : Mediated Priming in Semantic Space," no. April, 2000.

[16] S. Padó and M. Lapata, "Dependency-Based Construction of Semantic Space Models," *Computational Linguistics*, vol. 33, no. December 2004, pp. 161–199, 2007.

[17] H. Li, B. Ma, and C.-H. Lee, "A Vector Space Modeling Approach to Spoken Language Identification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 271–284, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4032773

[18] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, "I-vector representation based on bottleneck features for language identification," 2013.

[19] P. Matejka, L. Zhang, T. Ng, S. H. Mallidi, O. Glembek, J. Ma, and Z. Bing, "Neural network bottleneck features for language identification," in *Odyssey*, 2014.

[20] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *INTERSPEECH*, 2015.

[21] P. Cardinal, N. Dehak, Y. Zhang, and J. Glass, "Speaker adaptation using the i-vector technique for bottleneck features," in *INTERSPEECH*, 2015.

[22] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, pp. 788–798, 2011.

[23] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, pp. 980–988, 2008.

[24] A. Ali, Y. Zhang, and S. Vogel, "QCRI advanced transcription ssystem (QATS)," in *SLT*, 2014.

[25] S. Wray and A. Ali, "Crowdsource a little to label a lot: Labeling a speech corpus of dialectal arabic," in *INTERSPEECH*, 2015.

[26] M. Collins, "Language Modeling," available at http://www.cs.columbia.edu/ mcollins/lm-spring2013.pdf.

[27] A. Ng, "CS229 Lecture notes Generative Learning algorithms," no. 0, pp. 1–14, available at http://cs229.stanford.edu/notes/cs229-notes2.pdf.

[28] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1048–1054, 1999.

[29] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 workshop on machine learning for information filtering*, vol. 1, 1999, pp. 61–67.

[30] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, "Analysis and processing of lecture audio data: Preliminary investigations," in *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*. Association for Computational Linguistics, 2004, pp. 9–12.