# Birzeit Arabic Dialect Identification System for the 2018 VarDial Challenge

**Rabee Naser**
Electrical & Computer Engineering Dept
Birzeit University
West Bank, Palestine
rabinasser@gmail.com

**Abualsoud Hanani**
Electrical & Computer Engineering Dept
Birzeit University
West Bank, Palestine
ahanani@birzeit.edu

## Abstract

This paper describes our Automatic Dialect Recognition (ADI) system for the VarDial 2018 challenge, with the goal of distinguishing four major Arabic dialects, as well as Modern Standard Arabic (MSA). The training and development ADI VarDial 2018 data consists of 16,157 utterances, their words transcription, their phonetic transcriptions obtained with four non-Arabic phoneme recognizers and acoustic embedding data. Our overall system is a combination of four different systems. One system uses the words transcriptions and tries to recognize the speaker dialect by modeling the sequence of words for each dialect. Another system tries to recognize the dialect by modeling the phone sequences produced by non-Arabic phone recognizers, whereas, the other two systems use GMM trained on the acoustic features for recognizing the dialect. The best performance was achieved by the fused system which combines four systems together, with F1 micro of 68.77%.

## 1 Introduction

Work on accent and dialect recognition in the literature is still traditionally split into acoustic-only, acoustic-lexical and acoustic-phonetic classification systems. Most of the state-of-the-art systems focus on acoustic methods. In the past work on the VarDial 2017/2016 (Zampieri et al., 2017)(Malmasi et al., 2016) shared tasks, the presented work capitalized on the combination of i-vector technique, which represents each utterance by a low-dimensional vector estimated from the variability subspace (DeMarco and COX, 2013), and lexical word sequence extracted by Arabic ASR. Deep Neural Networks (DNN) has been successfully used for modelling both the acoustic features and the lexical features extracted from the words sequences. (Ionescu and Butnaru, 2017) got the first rank on the ADI shared task with F1 score of 76.32% in the last year challenge (Zampieri et al., 2017) . They used multiple kernel approach for this task. Our previous work on the same task (Hanani et al., 2017) ranked in the fourth place with our fused system that combines word-entropy and character-string entropy with the acoustic i-vector system. The best F1 (micro) score we had achieved was 62.87%.

The VarDial 2018 (Zampieri et al., 2018) consists of the same five shared tasks of the last year edition. One of these tasks is the Arabic Dialect Identification (ADI) which addresses the multi-dialectal challenge in spoken Arabic in broadcast news domain. Previously, organizers of the ADI shared acoustic features and lexical word sequence extracted from large-vocabulary speech recognition (LVCSR). This year, they add phonetic features, which allow the use of both acoustic and phonetic features, which are helpful for distinguishing between different dialects. In the previous work, the most successful ADI systems that combine acoustic with lexical features. With the added phonetic features, the Phonotactic systems such as Phone Recognizer followed by Language Model (PRLM) is investigated and compared with the lexical and acoustic based systems.





*Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 210–217
Santa Fe, New Mexico, USA, August 20, 2018.

## 2 Data Description

The training data presented by the organizers of the ADI shared task contained embedding acoustic features, lexical words recognized by an Arabic Automatic Speech Recognition (AASR) system, and phonemes recognized by four non-Arabic (Czech,English,Hungarian and Russian) phone recognizers. A list of the WAV files with links was also provided. The dataset is already divided into three subsets; train, dev and test. The train subset consists of 14591 utterances, dev subset consists of 1566 utterances, whereas test subset consists of 6837 utterances. Each utterance was labeled with one of the five target Arabic dialects (Egyptian, Levantine, Gulf, North Africa, MSA). 5435 testing utterances were added to the original 1492 testing utterances from the mgb_3 dataset .

| **Dialect** | **Training** | **Development** | **Testing** |
|---|---|---|---|
| Egyptian | 3177 | 315 | 1445 |
| Gulf | 2873 | 265 | 1397 |
| Levantine | 3117 | 348 | 1465 |
| North African | 3205 | 355 | 1324 |
| MSA | 2219 | 283 | 1206 |
| Total | 14591 | 1566 | 6837 |

Table 1: The ADI Data for VarDial 2018 shared task

## 3 Systems Description

Our overall system consists of multiple systems. Some of them use acoustic features for modeling target dialects, some use lexical features extracted from the words sequence and some use phonetic features extracted from phones and GMM tokens sequences. The main difference from the last year systems is the deep neural networks incorporation in the acoustic and words/phones sequences.

Multiple Systems were investigated for the purpose of participation in the ADI shared task, we had the results of only one system ready at the the submission deadline.

We will first introduce the participating system. And then we will state the other systems that were prepared for the shared task and we will discuss their results. We will show also potential candidate systems that could have raised the rank of our team.

### 3.1 Embedding Features SVM

SVM models have shown in (Hanani et al., 2017) that it is a good classifier with low dimension interpretation of the data files. We have chosen to train two separate SVM models for the training and development data instead of using one model. The chosen SVM model is a multiclass SVM that predicts the class of the utterance instead of the traditional 2 class SVM. We have presented the 600 embedding features directly as inputs to the models. And the predicted classes of each model was recorded.

### 3.2 Feed Forward Neural Network

We have trained two separate feed-forward DNN models with multiple layers each. The inputs of the DNN models were once again the acoustic embedding features. The output of the DNNs is a layer of five values, each value represents the probability that a given utterance belongs to the corresponding dialect. The dialect that corresponds to the output with highest value is recorded as the predicted dialect of the model.

### 3.3 ADI Run

As stated before, we had only one valid run in the shared task and it was a combination of the previous systems. Every test file has four predictions, these predictions were gathered in one vector. Then we have used a voting technique to specify the final predicted dialect. This system will serve as the baseline to our post shared task work. We have completed our proposed systems and proposed new systems that

have shown encouraging results. The following systems are the systems that we continued our work on after the official ADI run.

### 3.4 SVM Baseline

We reconfigured and tuned the SVM model. The training and development data was concatenated, and the system performed slightly better than the original system scoring up to 53.82%.

### 3.5 WAV Fles Processing

We used the wav files to extract acoustic features. The files were divided into frames by the length of 25ms. 19 Mel-Scale Cepstral Coefcients (MFCC) was extracted, then we applied the Rasta Filtration. we appended the Shifted-Delta Cepstra to each frame features, resulting in 38 features per frame. These feature were used in the acoustic based models.

### 3.6 GMM-UBM Model

we have built different Gaussian mixture models with different number of Gaussian mixtures. A GMM-UBM model for each number of Gaussian mixtures was built. we used all of the training and development data to build one Universal Background GMM (referred as UBM). We used a K-means algorithm to initialize the model parameters. We then ran the expectation maximization algorithm for four iterations to estimate the UBM model parameters.

we adapted the UBM to every dialect using dialect specific data and MAP adaptation technique. The resulting GMM models then was used for predicting the labels of the testing data. The class model with maximum score determines the predicted dialect. The GMM-UBM model performed poorly comparing to the embedding features result scoring only 35.1% for 2048 mixtures.

### 3.7 GMM Tokenizer

We used the UBM model described above as a tokenizer, that converts a sequence of acoustic features into a sequence of the gaussian components which gives the highest probability for each frame. Comparing with phonotactic system, the Gaussian component with the highest probability is recognized as the phoneme of the frame. A vector of the recognized phonemes is extracted for each file.

N-gram vector is extracted for each utterance by counting the occurrences of each n-gram. The n-gram vectors are then fed to a Multiclass SVM model. Uni-gram and bi-gram vectors of 128 mixtures models was applied, and only uni-gram vectors was tested for the 2048 mixtures due to memory limitation. The systems resulted in 40.28%, 46.18%, 47.86% respectively.

### 3.8 Word TF-IDF Model

We gathered all the word files from training and development data to extract a vocabulary of 68707 unique words. Then we provided the word files to a tf-idf vector extractor, which takes the utterances as an input and then produces a vector of 68707 features, each feature represents the term frequency-inverse document frequency of a single word in the vocabulary.

The term frequency represents how frequent the word is in the text, the word which occurred the most will have the highest score. But the words that are frequent in all the documents will have little information for the classifier. The Inverse Document Frequency is used to reduce the score of the words that are frequent in most of the documents.

The term frequency and inverse-document frequency are calculated by the following equations:

$$tf_t = 1 + \log(count_t) \tag{1}$$

$$idf_t = 1 + \log(\frac{N}{d_t}) \tag{2}$$

$count_t$ represents the count of term $t$ in the document while $N$ represents the total number of utterances and $d_t$ is the number of documents that contains the word $t$, then the $tf - idf$ is calculated by the following equation:

$$tfidf_t = tf_t \times idf_t \tag{3}$$

A Multiclass SVM model was used to train and test the tf-idf vectors, the result was 51.47% accuracy.

### 3.9 Sentence Similarity Model

In this system for each utterance the similarity to all the sentences in the training and development data is calculated resulting in a vector of 16157 features which is the number of training and development data files.

We used the Jaccard similarity algorithm to find the similarity between the utterances. The Jaccard algorithm is also known as intersection over union. The Jaccard similarity between sentences A and B is shown in the following equation:

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{4}$$

The similarity index varies from 0 to 1. The similarity index of 1 means that the two sentences are actually one sentence, while 0 similarity means no similarity. Using the similarity coefficients for the purpose of classification can be justified because sentences are likely to be more similar to sentences from their own dialect than to sentences from other dialects.

An application was developed to produce the similarity vectors, and the 16157 vectors of 16157 features of training and development were used to train an SVM model. The model was used to predict the dialect of the testing data, and the system has 51.89% accuracy.

### 3.10 PRLM Model

We have used the phonemes of the Czech phoneme recognizer to build a Phonotactic Phone Recognizer followed by Language Model (PRLM). Three separate PRLM models for uni-gram, uni+bi-gram and uni+bi+tri-gram sequences were trained. The sequences were used as inputs to SVM models. We used the development data for testing and our results were 37.04%, 38.9% and 38.76%, respectively. We used these results as baseline to test our proposed tf-idf approach on the phoneme level.

### 3.11 Phoneme TF-IDF Model

We have adopted the same system used in the word tf-idf system in the phoneme level. First we developed a system to extract uni-grams and bi-grams and concatenate them in one sentence. Then the sentences were introduced to the word tf-idf vector extractor. An SVM was built to test the performance of the new features, and it scored 41.12% accuracy in identifying development data.

### 3.12 Bottleneck Features

Neural networks had been lately used extensively in machine learning problems. The deep neural networks which consists of multiple interconnected layers between the input and output layers was used in speech recognition tasks(Najafian et al., 2018). DNNs had not been used just as classifiers but they were also used as feature extractors(Ali et al., 2016). Neural networks can be configured to have multiple layers with big numbers of neurons, and between these layers a much smaller layer can be added as illustrated in Figure 1 . The state of this layer can be used as feature vector to represent the original data. The bottleneck features can be used to feed another learning model or another DNN.

In the embedding acoustic features system and the sentence similarity system a total of 16757 features were used to train the SVM models. Bottleneck features were extracted to reduce dimensionality and combine both systems.

A Neural network was built for the embedding features system, and the similarity system with multiple hidden layers, each one of them contained a bottleneck layer of 30 neurons, the DNNs were trained and the state of the 30 neurons was recorded for each of the dataset files.

The bottleneck features were combined for each utterance into 60 features long vectors. The vectors then were used to train an SVM model, the system scored 61.3% F1 macro which outperformed any score presented by any team in this year ADI competition.

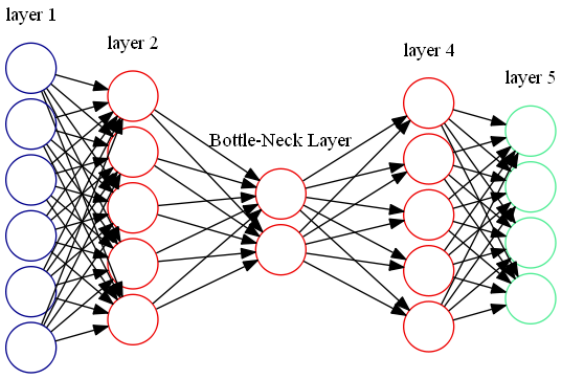


Figure 1: Bottleneck features

### 3.13 Systems Fusing

A fusing system was introduced to fuse the results of all the previously described systems. The raw output of each of the previous models is a vector of 5 features that represents the scores of the dialect specific models for each utterance. The vectors of the testing data were recorded and concatenated.

Seven fold cross validation was applied on the fused vectors by dividing the testing data into seven segments, leaving one segment at a time for testing and the six other segments for training fusion SVM model.

## 4 Results and Discussion

The results of the ADI shared task of this year are much worse than the results of the 2017 shared task. This was expected since this year's testing data was much bigger than previous year's. 6837 test files were used in 2018 instead of 1492 files in 2017. The number of the files was not the only cause but as it was stated by the organizers this years testing data contained files obtained from YouTube, in addition to the original files used in the previous year.

Six teams participated in the ADI Shared Task. Only the baseline system results were ready at the submission due date. It ranked third as shown in Table 2 and its confusion matrix is shown in Figure 2. The work continued on other systems and models as well as on the baseline model to improve accuracy.

| **Team Name** | **Score** |
|---|---|
| UnibucKernel | 0.5892 |
| safina | 0.5759 |
| BZU | 0.5338 |
| SystranLabs | 0.5289 |
| taraka_rama | 0.5140 |
| Arabic_Identification | 0.4997 |

Table 2: Shared Task Results.

The traditional baseline GMM system performed below last year's result in (Hanani et al., 2017) (35.1% compared to 40.16%). This can be related to the additional testing data coming from different environment. It can be also related to the reduced choice of the MFCC features extracted this year (38 instead of 68 in 2017). Interestingly, the GMM tokenizer performed slightly better than last year in (Hanani et al., 2017) with this year best result of 47.86% in contrast to 46.85% in 2017.

The tf-idf model was a replacement of traditional uni-gram model. It scored 51.47% better than we would have expected and relatively close to the submitted result. It is interesting to compare this system's score to what would a uni-gram word count model score.

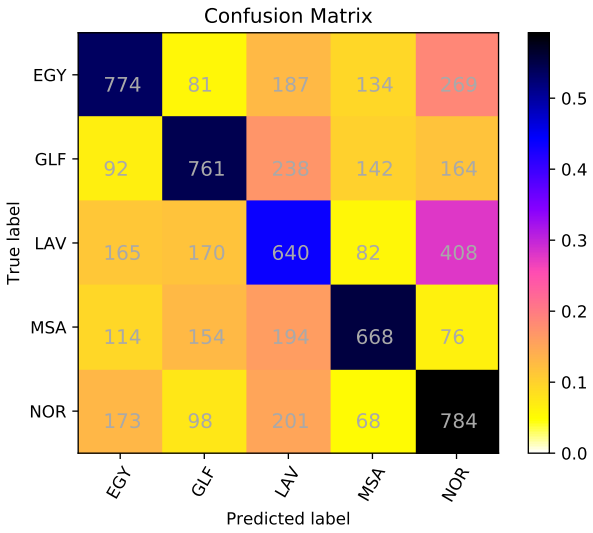


Figure 2: ADI task, RUN1 (Baseline results)

The tf-idf model showed promising results when compared to the PRLM model on czech phone recognizer data. By testing with the testing data, it only scored 35.79%, due to time limitation no further phone recognizer's data or setups were tested.

The result of the sentence similarity system with 51.89% accuracy was slightly better than that of the tf-idf; but the idea of using 16157 features instead of 68707 and getting better results led to further investigation of the use of smaller vectors.

After analyzing the results of each system, it was apparent that different systems predicted different correct utterances. Only 30% of the testing files were predicted correctly by both of the similarity model and the embedding feature SVM model. And 40% of the data was correctly identified only by one of them. That showed us that there is room for improvement by fusing the systems together.

The use of bottleneck features improved the performance of the system to 61.3%. The fact that the systems don't need the labels of the testing data, means that it could have been presented to the shared task and it would have easily ranked the first with 2.5% improvement over the best score. The confusion matrix of the best system is shown in Figure 3

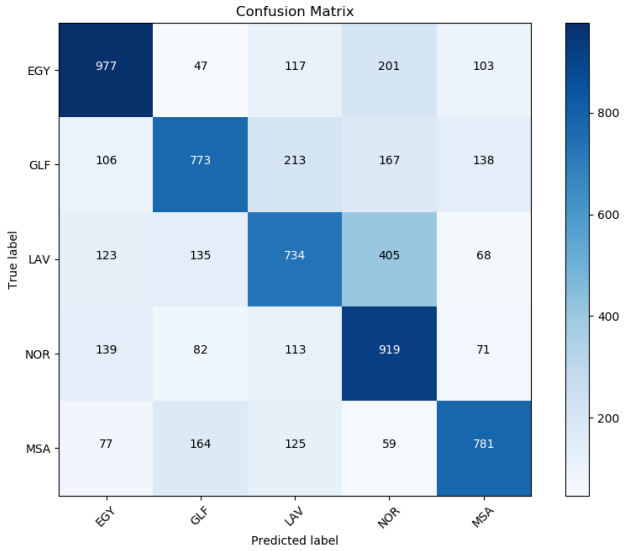


Figure 3: Bottleneck features results

The overall fused system scored 68.77% F1 micro. The confusion matrix of this system is shown in Figure 4

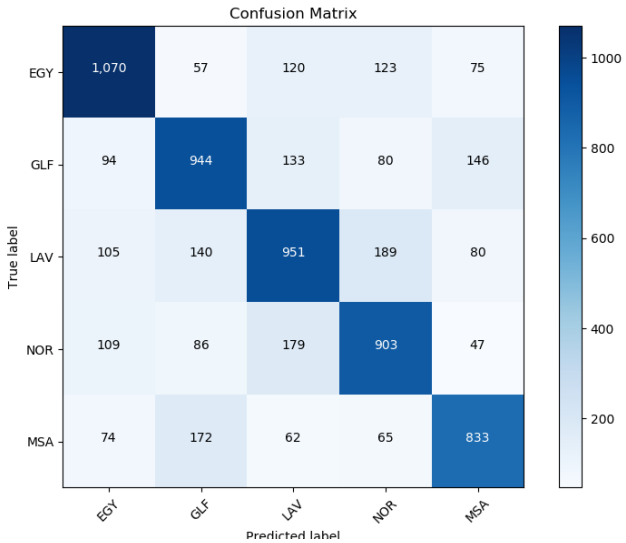


Figure 4: fusing results

## 5 Conclusions and Future Work

In this paper we presented some of the techniques we intended to use for the the shared task. We are looking forward to further investigate the bottleneck features on the word and phoneme level and to study the sentence similarity approach looking into other sentence to sentence relation models. we have used deep learning in different techniques and we have tried to use Long Short Term Memory models that didn't converge in the learning process, so they weren't described in the paper. It would be interesting to analyze the reasons behind that. we would like to divide the data in other ways to include the new testing data in training to examine the effect of the difference of recording environment if it exists.

## References

[Ali et al.2016] Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals. 2016. Automatic Dialect Detection in Arabic Broadcast Speech. In *Proceedings of INTERSPEECH*, pages 2934–2938.

[DeMarco and COX2013] Andrea DeMarco and Stephen J. COX. 2013. Native accent classification via i-vectors and speaker compensation fusion. In *Proceedings of INTERSPEECH*, pages 1472–1476.

[Hanani et al.2017] Abualsoud Hanani, Aziz Qaroush, and Stephen Taylor. 2017. Identifying dialects with textual and acoustic cues. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 93–101, Valencia, Spain, April.

[Ionescu and Butnaru2017] Radu Tudor Ionescu and Andrei Butnaru. 2017. Learning to identify arabic and german dialects using multiple kernels. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 200–209, Valencia, Spain, April.

[Malmasi et al.2016] Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and arabic dialect identification: A report on the third dsl shared task. In *Proceedings of the 3rd Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (VarDial)*, Osaka, Japan.

[Najafian et al.2018] Maryam Najafian, Sameer Khurana, Suwon Shon, Ahmed Ali, and James Glass. 2018. Exploiting convolutional neural networks for phonotactic based dialect identification. In *ICASSP*.

[Zampieri et al.2017] Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Valencia, Spain.

[Zampieri et al.2018] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shuon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.