Arabic Dialect Identification Using iVectors and ASR Transcripts

Shervin Malmasi
Harvard Medical School, USA
Macquarie University, Australia
shervin.malmasi@mq.edu.au

Marcos Zampieri
University of Cologne
Germany
mzampie2@uni-koeln.de

Abstract

This paper presents the systems submitted by the MAZA team to the Arabic Dialect Identification (ADI) shared task at the VarDial Evaluation Campaign 2017. The goal of the task is to evaluate computational models to identify the dialect of Arabic utterances using both audio and text transcriptions. The ADI shared task dataset included Modern Standard Arabic (MSA) and four Arabic dialects: Egyptian, Gulf, Levantine, and North-African. The three systems submitted by MAZA are based on combinations of multiple machine learning classifiers arranged as (1) voting ensemble; (2) mean probability ensemble; (3) meta-classifier. The best results were obtained by the meta-classifier achieving 71.7% accuracy, ranking second among the six teams which participated in the ADI shared task.

1 Introduction

The interest in Arabic natural language processing (NLP) has grown substantially in the last decades. This is evidenced by several publications on the topic and the dedicated series of workshops (WANLP) co-located with major international computational linguistics conferences.[1]

Several Arabic dialects are spoken in North Africa and in the Middle East co-existing with Modern Standard Arabic (MSA) in a diglossic situation. Arabic dialects are used in both spoken and written forms (*e.g.* user-generated content) and pose a number of challenges for NLP applications. Several studies on dialectal variation of Arabic have been published including corpus compilation for Arabic dialects (Al-Sabbagh and Girju, 2012; Cotterell and Callison-Burch, 2014), parsing (Chiang et al., 2006), machine translation of Arabic dialects (Zbib et al., 2012), and finally, the topic of the ADI shared task, Arabic dialect identification (Zaidan and Callison-Burch, 2014; Sadat et al., 2014; Malmasi et al., 2015).

In this paper we present the MAZA entries for the 2017 ADI shared task which was organized as part of the VarDial Evaluation Campaign 2017 (Zampieri et al., 2017). The ADI shared task dataset (Ali et al., 2016) included audio and transcripts from Modern Standard Arabic (MSA) and four Arabic dialects: Egyptian, Gulf, Levantine, and North-African.

2 Related Work

There have been several studies published on Arabic dialect identification applied to both speech and text.[2] Examples of Arabic dialect identification on speech data include the work by Biadsy et al. (2009), Biadsy (2011), Biadsy and Hirschberg (2009), and Bahari et al. (2014). Identifying Arabic dialects in text also became a popular research topic in recent years with several studies published about it (Zaidan and Callison-Burch, 2014; Sadat et al., 2014; Tillmann et al., 2014; Malmasi et al., 2015).

To our knowledge, however, the 2017 ADI is the first shared task to provide participants with the opportunity to carry out Arabic dialect identification using a dataset containing both audio and text (transcriptions). The first edition of the ADI shared task, organized in 2016 as a sub-task of the DSL shared task (Malmasi et al., 2016c), used a similar dataset to the ADI 2017 dataset, but included only transcriptions.

[1] https://sites.google.com/a/nyu.edu/wanlp2017/

[2] See Shoufan and Al-Ameri (2015) for a survey on NLP methods for processing Arabic dialects including a section on Arabic dialect identification.

Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 178–183,
Valencia, Spain, April 3, 2017.

3 Methods and Data

We approach this task as a multi-class classification problem. For our base classifier we utilize a linear Support Vector Machine (SVM). SVMs have proven to deliver very good performance in discriminating between language varieties and in other text classification problems, SVMs achieved first place in both the 2015 (Malmasi and Dras, 2015a) and 2014 (Goutte et al., 2014) editions of the DSL shared task.[3]

3.1 Data

The data comes from the aforementioned Arabic dialect dataset by Ali et al. (2016) used in the 2016 edition of the ADI shared task. It contains audio and ASR transcripts of broadcast, debate, and discussion programs from videos by Al Jazeera in MSA and four Arabic dialects: Egyptian, Gulf, Levantine, and North-African. In 2016, the organizers released only the transcriptions of these videos and in 2017 transcriptions are combined with audio features providing participants with an interesting opportunity to test computational methods that can be used both for text and speech. We combined all the train/dev data (25,311 samples). The test set contained 1,492 instances.

3.2 Features

In this section we describe our features and evaluate their performance under cross-validation.

We employ two lexical surface feature types for this task, as described below. These are extracted from the transcriptions without any pre-processing (*e.g.* case folding or tokenization) on texts prior to feature extraction. Pre-processing was not needed as the data are computer-generated ASR transcripts. We also used the provided iVector features, as described below.

- **Character n-grams:** This sub-word feature uses the constituent characters that make up the whole text. When used as n-grams, the features are n-character slices of the text. Linguistically, these substrings, depending on the length, can implicitly capture various sub-lexical features including single letters, phonemes, syllables, morphemes & suffixes. Here we examine n-grams of size 1–8.

- **Word n-grams:** The surface forms of words can be used as a feature for classification. Each unique word may be used as a feature (i.e. unigrams), but the use of bigram distributions is also common. In this scenario, the n-grams are extracted along with their frequency distributions. For this study we evaluate unigram features.

- **iVector Audio Features:** Identity vectors or iVectors are a probabilistic compression process for dimensionality reduction. They have been used in speech processing for dialect and accent identification (Bahari et al., 2014), as well as for language identification systems (Dehak et al., 2011).

We now report our cross-validation results on the training data. We began by testing individual feature types, with results displayed in Figure 1.

We observe that many character n-grams outperform the word unigram features. Character 4-grams and above obtained higher results than those obtained using word unigrams. The best transcript-based results were obtained with character 6-grams achieving 76.2% accuracy. The audio-based iVector features, however, performed substantially better with 85.3% accuracy. This is a very large difference of almost 10% accuracy compared to the performance obtained using words and characters.

Having demonstrated that these features are useful for this task, we proceed to describe our systems in the next section.

3.3 Systems

We created three systems for our submission, as described below.

3.4 Voting Ensemble (System 1)

The best performing system in the 2015 edition of the DSL challenge (Malmasi and Dras, 2015a) used SVM ensembles evidencing the adequacy of this approach for the task of discriminating between similar languages and language varieties. In light of this, we decided to test two ensemble methods. Classifier ensembles have also proven to be an efficient and robust alternative in other text classification tasks such as language identification (Malmasi and Dras, 2015a), grammatical error detection (Xiang et al., 2015), and complex word identification (Malmasi et al., 2016a).

[3] See the 2014 and 2015 DSL shared task reports for more information (Zampieri et al., 2015; Zampieri et al., 2014) and Goutte et al. (2016) for a comprehensive evaluation on the first two DSL shared tasks.

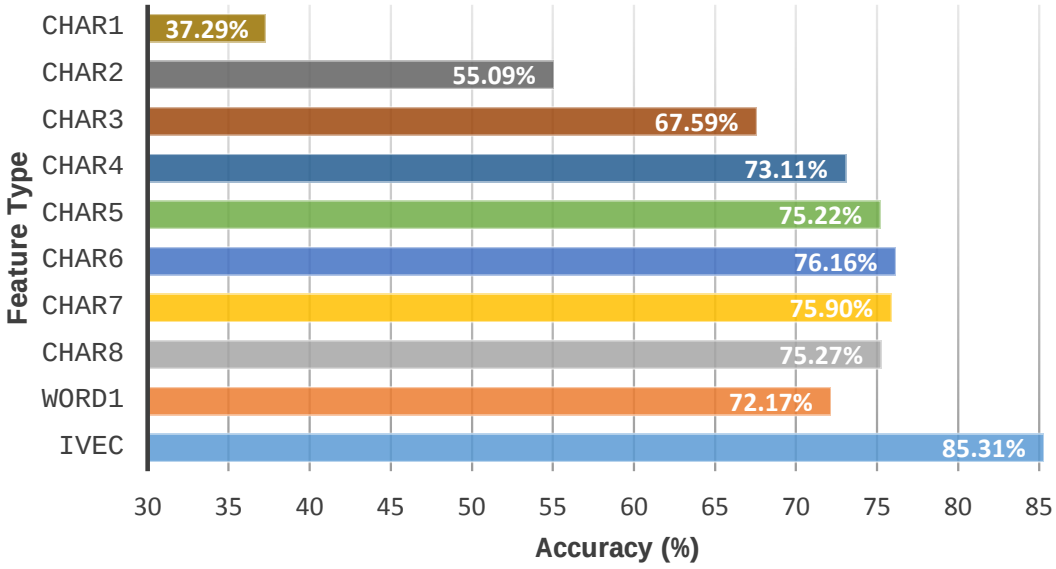

Figure 1: Cross-validation performance for each of our individual feature types.

We follow the methodology described by Malmasi and Dras (2015a): we extract a number of different feature types and train a single linear model using each feature type. Our ensemble was created using linear Support Vector Machine classifiers.[4] We used all of the feature types listed in Section 3.2 to create our ensemble of classifiers.

Each classifier predicts every input and also assigns a continuous output to each of the possible labels. Using this information, we created the following two ensembles.

In the first system each classifier votes for a single class label. The votes are tallied and the label with the highest number[5] of votes wins. Ties are broken arbitrarily. This voting method is very simple and does not have any parameters to tune. An extensive analysis of this method and its theoretical underpinnings can be found in the work of (Kuncheva, 2004, p. 112). We submitted this system as run 1.

3.5 Mean Probability Ensemble (System 2)

Our second system is similar to System 1 above, but with a different combination method. Instead of a single vote, the probability estimates for each class[6] are added together and the class label with the highest average probability is the winner. An important aspect of using probability outputs in this way is that a classifier's support for the true class label is taken in to account, even when it is not the predicted label (*e.g.* it could have the second highest probability). This method has been shown to work well on a wide range of problems and, in general, it is considered to be simple, intuitive, stable (Kuncheva, 2014, p. 155) and resilient to estimation errors (Kittler et al., 1998) making it one of the most robust combiners discussed in the literature. We submitted this system as run 2.

3.6 Meta-classifier (System 3)

In addition to classifier ensembles, meta-classifier systems have proven to be very competitive for text classification tasks (Malmasi and Zampieri, 2016) and we decided to include a meta-classifier in our entry. Also referred to as classifier stacking, a meta-classifier architecture is generally composed of an ensemble of base classifiers that each make predictions for all of the input data. Their individual predictions, along with the gold labels are used to train a second-level meta-classifier that learns to predict the label for an input, given the decisions of the individual classifiers. This setup is illustrated in Figure 2. This meta-classifier attempts to learn from the collective knowledge represented by the ensemble of local classifiers.

The first step in such an architecture is to create the set of base classifiers that form the first layer. For this we used the same base classifiers as our ensembles described above.

[4] Linear SVMs have proven effective for text classification tasks (Malmasi and Dras, 2014; Malmasi and Dras, 2015b).

[5] This differs with a *majority* voting combiner where a label must obtain over 50% of the votes to win. However, the names are sometimes used interchangeably.

[6] SVM results can be converted to per-class probability scores using Platt scaling.

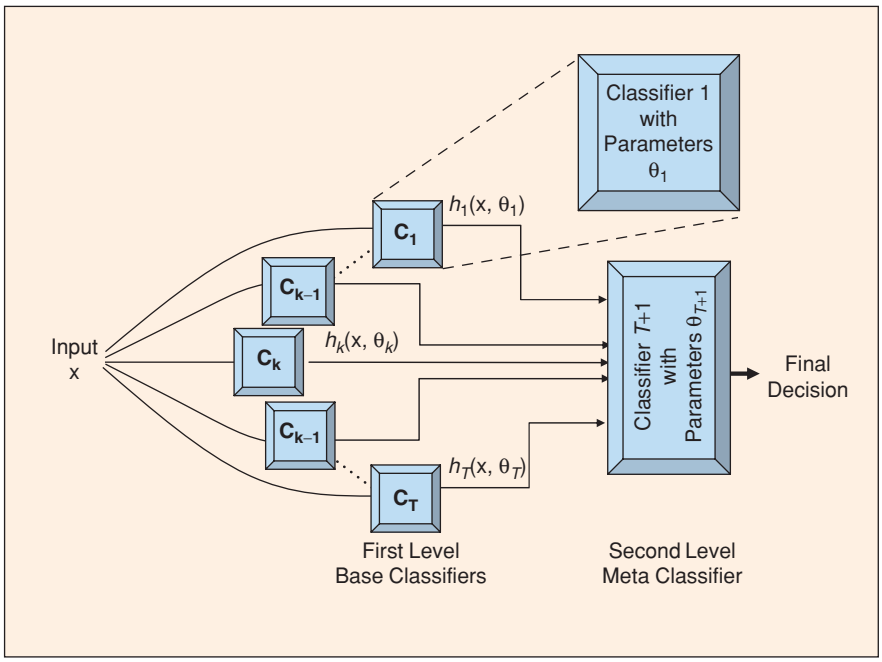

Figure 2: An illustration of a meta-classifier architecture. Image reproduced from Polikar (2006).

In this system we combined the probability outputs of our seven individual classifier and used them to train a meta-classifier via cross-validation. Following Malmasi et al. (2016b), we used a Random Forest as our meta-classification algorithm. We submitted this system as run 3.

4 Results

4.1 Cross-validation Results

We first report the cross-validation results of our three systems on the training data. Results are shown in Table 1.

System	Accuracy
Majority Class Baseline	0.219
Voting Ensemble (System 1)	0.854
Probability Ensemble (System 2)	0.950
Meta-Classifier (System 3)	**0.977**

Table 1: Cross-validation results for the Arabic training data.

We note that all of these methods outperform any individual feature type, with the meta-classifier achieving the best result of 97.7%. This is a very large increase over the weakest system, which is the voting ensemble with 85.4% accuracy. For the voting ensemble 1,165 of the 25,311 training samples (4.60%) were ties that were broken arbitrarily. This is an issue that can occur when there are an even number of classifiers in a voting ensemble.

4.2 Test Set Results

Finally, in this section we report the results of our three submissions generated from the unlabelled test data. The samples in the test set were slightly unbalanced with a majority class baseline of 23.1%. Shared task performance was evaluated and teams ranked according to the weighted F1-score which provides a balance between precision and recall. Accuracy, along with macro- and micro-averaged F1-scores were also reported.

We observe that the meta-classifier achieved the best result among our three entries, following the same relative pattern as the cross-validation results. The meta-classifier system ranked second among the six teams participating in the ADI task.

In Figure 3 we present the confusion matrix heat map for the output of our best system, the meta-classifier. The confusion matrix confirms the assumption that not all classes presented in the dataset are equally difficult to identify. For example, the system is able to identify MSA utterances with substantially higher performance than the performance obtained when identifying any of the four Arabic dialects present in the dataset. We also observe a higher degree of confusion in discriminating between Gulf and Levantine Arabic compared to the other dialects and MSA.

System	Accuracy	F1 (micro)	F1 (macro)	F1 (weighted)
Majority Class Baseline	0.231	—	—	—
Voting Ensemble (run1)	0.6086	0.6086	0.6032	0.6073
Probability Ensemble (run2)	0.6689	0.6689	0.6671	0.6679
Meta-classifier (run3)	**0.7165**	**0.7165**	**0.7164**	**0.7170**

Table 2: MAZA Results for the ADI task.

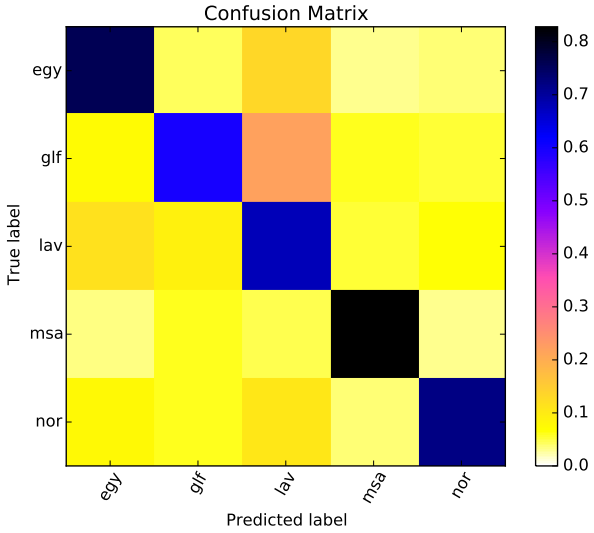

Figure 3: Confusion Matrix for Run 3.

Finally, another important observation is that the test set results are somewhat lower than the cross-validation results. Although this was not specified by the task organizers, it may have been the case that the test data was drawn from a different distribution as the training data. An analysis of the most informative features and the misclassified instances in both the training and test sets may provide an explanation for this difference.

5 Conclusion

We presented three systems trained to identified MSA and four Arabic dialects using iVectors and ASR transcripts. The best results were obtained by a meta-classifier achieving 71.7% accuracy and ranking second in the ADI shared task 2017. To the best of our knowledge, this was the first time that computational methods have been evaluated on Arabic dialect detection using audio and text.

An important insight is that combining text-based features from transcripts with audio-based features can substantially improve performance. Additionally, we also saw that a meta-classifier can provide a significant performance boost compared to a classifier ensemble approach.

Acknowledgements

We would like to thank Preslav Nakov and Ahmed Ali for proposing and organizing the ADI task. We also thank the VarDial workshop reviewers who provided us valuable feedback and suggestions to improve this manuscript.

References

Rania Al-Sabbagh and Roxana Girju. 2012. YADAC: Yet another Dialectal Arabic Corpus. In *Proceedings of LREC*.

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*.

Mohamad Hasan Bahari, Najim Dehak, Lukas Burget, Ahmed M Ali, Jim Glass, et al. 2014. Non-negative Factor Analysis of Gaussian Mixture Model Weight Adaptation for Language and Dialect Recognition. *IEEE/ACM transactions on audio, speech, and language processing*, 22(7):1117–1129.

Fadi Biadsy and Julia Hirschberg. 2009. Using Prosody and Phonotactics in Arabic Dialect Identification. In *Proceedings of INTERSPEECH*.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken Arabic Dialect Identification using Phonotactic Modeling. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*.

Fadi Biadsy. 2011. *Automatic dialect and accent recognition and its application to speech recognition*. Ph.D. thesis, Columbia University.

David Chiang, Mona T Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic Dialects. In *Proceedings of EACL*.

Ryan Cotterell and Chris Callison-Burch. 2014. A Multi-dialect, Multi-genre Corpus of Informal Written Arabic. In *Proceedings LREC*.

Najim Dehak, Pedro A Torres-Carrasquillo, Douglas A Reynolds, and Reda Dehak. 2011. Language Recognition via i-vectors and Dimensionality Reduction. In *Proceedings of INTERSPEECH*.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The NRC System for Discriminating Similar Languages. In *Proceedings of the VarDial Workshop*.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of LREC*.

Josef Kittler, Mohamad Hatef, Robert PW Duin, and Jiri Matas. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Ludmila I Kuncheva. 2004. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons.

Ludmila I Kuncheva. 2014. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, second edition.

Shervin Malmasi and Mark Dras. 2014. Chinese Native Language Identification. In *Proceedings of EACL*.

Shervin Malmasi and Mark Dras. 2015a. Language identification using classifier ensembles. In *Proceedings of the LT4VarDial Workshop*.

Shervin Malmasi and Mark Dras. 2015b. Multilingual Native Language Identification. *Natural Language Engineering*, pages 1–53.

Shervin Malmasi and Marcos Zampieri. 2016. Arabic Dialect Identification in Speech Transcripts. In *Proceedings of the VarDial Workshop*.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. 2015. Arabic Dialect Identification using a Parallel Multidialectal Corpus. In *Proceedings of PACLING*.

Shervin Malmasi, Mark Dras, and Marcos Zampieri. 2016a. LTG at SemEval-2016 Task 11: Complex Word Identification with Classifier Ensembles. In *Proceedings of SemEval*.

Shervin Malmasi, Marcos Zampieri, and Mark Dras. 2016b. Predicting Post Severity in Mental Health Forums. In *Proceedings of the CLPsych Workshop*.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016c. Discriminating between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the VarDial Workshop*.

Robi Polikar. 2006. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45.

Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. 2014. Automatic Identification of Arabic Language Varieties and Dialects in Social Media. In *Proceedings of the SocialNLP Workshop*.

Abdulhadi Shoufan and Sumaya Al-Ameri. 2015. Natural Language Processing for Dialectical Arabic: A Survey. In *Proceedings of the Arabic NLP Workshop*.

Christoph Tillmann, Saab Mansour, and Yaser Al-Onaizan. 2014. Improved Sentence-Level Arabic Dialect Classification. In *Proceedings of the VarDial Workshop*.

Yang Xiang, Xiaolong Wang, Wenying Han, and Qinghua Hong. 2015. Chinese grammatical error diagnosis using ensemble learning. In *Proceedings of the NLP-TEA Workshop*.

Omar F Zaidan and Chris Callison-Burch. 2014. Arabic Dialect Identification. *Computational Linguistics*.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A Report on the DSL Shared Task 2014. In *Proceedings of the VarDial Workshop*.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL Shared Task 2015. In *Proceedings of the LT4VarDial Workshop*.

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 VarDial Workshop*.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F Zaidan, and Chris Callison-Burch. 2012. Machine translation of Arabic Dialects. In *Proceedings of NAACL-HLT*.