QCRI @ DSL 2016: Spoken Arabic Dialect Identification Using Textual Features

Mohamed Eldesouki, Fahim Dalvi, Hassan Sajjad, *and* **Kareem Darwish**
Qatar Computing Research Institute
Hamad bin Khalifa University
Doha, Qatar
{*mohamohamed, faimaduddin, hsajjad, kdarwish*}`@qf.org.qa`

Abstract

The paper describes the QCRI submissions to the shared task of automatic Arabic dialect classification into 5 Arabic variants, namely Egyptian, Gulf, Levantine, North-African (Maghrebi), and Modern Standard Arabic (MSA). The relatively small training set is automatically generated from an ASR system. To avoid over-fitting on such small data, we selected and designed features that capture the morphological essence of the different dialects. We submitted four runs to the Arabic sub-task. For all runs, we used a combined feature vector of character bigrams, trigrams, 4-grams, and 5-grams. We tried several machine-learning algorithms, namely Logistic Regression, Naive Bayes, Neural Networks, and Support Vector Machines (SVM) with linear and string kernels. Our submitted runs used SVM with a linear kernel. In the closed submission, we got the best accuracy of 0.5136 and the third best weighted F1 score, with a difference of less than 0.002 from the best system.

1 Introduction

The Arabic language has various dialects and variants that exist in a continuous spectrum. They are a result of an interweave between the Arabic language that spread throughout the Middle East and North Africa and the indigenous languages in different countries. With the passage of time and the juxtaposition of cultures, dialects and variants of Arabic evolved and mutated. Among the varieties of Arabic, so-called Modern Standard Arabic (MSA) is the lingua franca of the Arab world, and it typically used in written and formal communications. On the other hand, Arabic dialects, such as Egyptian and Levantine, are usually spoken and used in informal communications, especially on social networks such as Twitter and Facebook.

Automatically identifying the dialect of a piece of text or of a spoken utterance can be beneficial for a variety of practical applications. For instance, it can aid Machine Translation (MT) systems in choosing the most appropriate model for translation.

In this paper we describe our dialect identification system that we used for Arabic dialect identification (sub-task 2) in the 2016 DSL shared task (Malmasi et al., 2016). We submitted a total of 4 runs to the shared task; 2 closed runs and 2 open runs. For closed runs, participants are only allowed to use the provided training set. For open runs, external resources are allowed. We tried several combinations of features such as bag-of-words features based-on words or character n-grams where terms are weighed by term frequency (tf) or term frequency and inverse document frequency (tf-idf). We also experimented with several machine learning classifiers including logistic regression, naive Bayes, neural networks, and Support Vector Machines (SVM) with different kernels. Our best run used an SVM classifier with a linear kernel trained on character n-gram features. Our best run achieved an accuracy of 0.5136 and an F-measure 0.5112. Compared to the systems that participated in the shared task, our system obtained the best accuracy and the third highest weighted F1 score, with a difference of less than 0.002 from the best system.

Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects,
pages 221–226, Osaka, Japan, December 12 2016.

2 Related Work

Arabic dialect identification work could be divided into two main streams, namely: (1) the creation of dialectal Arabic resources , and (2) the development of approaches and techniques for dialect identification. Here we present the most pertinent related work.

One of the early attempts to build Dialectal Arabic annotated resources was done by the COLABA project (Diab et al., 2010). The project harvested blogs about social issues, religion, and politics in four Arabic dialects, namely Egyptian, Iraqi, Levantine (Syrian, Lebanese, Palestinian, and Jordanian) and to a lesser extent Maghrebi. The blogs data was collected via a set of identified URLs as well as 40 dialectal queries from 25 annotators. The project attempted to tackle the non-standard orthography issues of Arabic dialects by defining a phonological scheme which they referred to as CCO. They used lexical features to select the most dialectal content based on the percentage of non-MSA words in the document being identified. They didn't mention any statistics about the data they collected. They used Information Retrieval (IR) for extrinsic evaluation.

The AOC dataset (Zaidan and Callison-Burch, 2011; Zaidan and Callison-Burch, 2014) was created from the content and comments of three newspapers namely Al-Ghad, Al-Riyadh and Al-Youm Al-Sabe', which originate from Jordan, Saudi Arabia, and Egypt respectively. The authors assumed that readers commenting on the different newspapers would use MSA or the dialect of the country of the newspaper. Thus, they considered all the dialectal comments extracted from Al-Ghad as Levantine, from Al-Riyadh as Gulf, and from Al-Youm Al-Sabe' as Egyptian. Out of 3.1 million sentences in AOC, they manually annotated about 108K sentences using crowdsourcing. They considered dialect classification as a language identification task. They built language models for MSA and each of the three dialects and used them to score text segments. The segment would be assigned a label corresponding to the language model with the lowest perplexity. They achieved an accuracy of 69.4%.

Mubarak and Darwish (2014) used user geographical information to build a multi-dialectal corpus from Twitter. Out of 175M tweets collected using Twitter API, they managed to annotate about 6.5M tweets with their dialects. Also, they conducted some analysis on the vocabulary distribution of different Arabic variants. Using the AOC dataset (Zaidan and Callison-Burch, 2011) and MSA corpus composed of 10 years worth of Aljazeera articles (about 114M tokens), they extracted about 45,000 n-grams (uni-, bi-, and tri-) and then manually label them as either MSA, Egyptian, Levantine, Gulf, Iraqi, or Maghrebi. They found that MSA words compose more than 50% of the words in the dialectal text and about 2500 n-gram are truly dialectal.

By showing that many of the most frequent discriminating words for Egyptian Arabic are in fact MSA words, Darwish et al. (2014) argued that Arabic dialect identification system built on the AOC dataset is biased towards the topics in the newspapers from which the corpus was built. Therefore, they discussed the need to identify lexical and linguistic features such as morphological patterns, word concatenations, and verb negation constructs to distinguish between dialectal Arabic and MSA. For evaluation, they used the Egyptian part of the LDC2012T09 corpus. They achieved an accuracy of 94.6% using lexical lists of dialectal words and verbs.

A new Multidialectal Parallel Corpus of Arabic (MPCA) released by Bouamor et al. (2014) was used by Malmasi et al. (2015) to train an SVM classifier to distinguish between MSA, Egyptian, Syrian, Jordanian, Palestinian and Tunisian. The classifier was a meta-classifier that was trained over the probabilities of an ensemble of classifiers that have been trained over different sets of word-level and character-level n-grams. They achieved an accuracy of 74%.

3 Dataset and Methodology

This section analyzes the dataset provided by the shared task and discusses the methodologies and approaches for both preparing the data and for developing our Arabic dialect identification system.

3.1 Dialectal Arabic Dataset

The DSL organizers provided a training dataset that is composed of Automatic Speech Recognition (ASR) transcripts (Ali et al., 2016), where utterances (or sentences) are labeled as Egyptian (EGY), Gulf

(GLF), Levantine (LAV), North-African (NOR), or Modern Standard Arabic (MSA). Each sentence is provided in a separate line in the following tab-delimited format:

```
sentence <tab> target-dialect-label
```

The Arabic sentences are transliterated into the Buckwalter encoding scheme. The training set has 7,619 sentences with a total of 315,829 words, of which 55,992 are unique. The average sentence length is 41 words. Table 1 shows the distribution of sentences, words, and unique words for the different variants. The numbers show that Egyptian has the longest sentences with an average of 53.8 words per sentence.

count	LAV	GLF	NOR	EGY	MSA	**Total**
sentences	**1,758**	1,672	1,612	1,578	999	7,619
words	66,219	64,081	51,593	**84,949**	48,987	315,829
unique words	19,198	17,842	20,271	**20,836**	13,607	55,992

Table 1: The distribution of sentences, words, and unique words for the different Arabic variants.

As expected, the frequent words are mostly stopwords with the words *fy* (in), *mn* (from), and *mA* (what) being the most frequent words across dialects. We retained stopwords as they are important for identifying dialects.

The data used in this shared task is different from data mentioned in the literature in that it is composed of ASR transcripts, and dialects are more common in conversational speech. Since the data was not manually revised (as part of the challenge), we found the following drawbacks in the data:

- The sentences are often quite incoherent and many sentences make no sense.
- Some lines have identical sentences, but with different dialect labels. Consider the following example (line 16 through line 19 in the dataset file):

```
16      $Ark Q EGY
17      $Ark Q GLF
18      $Ark Q LAV
19      $Ark Q NOR
```

Such problems complicate the dialect identification task. Furthermore, the nature and the distribution of the words and phrases in such data is different than the one extracted from sources such as blogs, forums, and tweets. Therefore, using such data to train a classifier (without taking into consideration the aforementioned issues) may yield a classifier that does not capture real patterns for a generalized dialect identifier.

Data Preparation: To perform offline experiments before submitting the official shared task runs, we split the provided training data into an 80/20 train/dev partitions, which would allow us to measure the effectiveness of different classification schemes. However, we used the entire set for training when submitting the final runs.

For some runs, we excluded sentences that are shorter than a certain threshold of words. We tried several threshold between 1 and 5. Furthermore, we also considered removing words with document frequency less than and/or greater than certain thresholds.

3.2 Methodology

We experimented with several supervised learning algorithms to perform a five-way classification among the five dialect classes. Apart from Multinomial Naive Bayes, we also trained a one-vs-rest logistic regression model and multi-class SVM with linear or string kernels. For SVM optimization, we used Stochastic Gradient Descent (SGD) over multiple passes on the dataset. We also trained a two layer neural network model over the dataset and evaluated its performance. With each of these learning algorithms, we tried several features including word level features and character level features. The shared task allowed for closed runs, in which we were allowed to use the provided training set exclusively, and open runs, in which we were allowed to use external resources. For the open runs, we augmented the shared task training data with the AOC data. Following is the description of the features that we used to train these models.

3.2.1 Word level features

Given that words are small units of semantic meaning, experimenting with word level features was the natural choice. We used words as follows:

Unigrams: As a baseline, we used word unigrams as features. We experimented with using raw word counts in a given sentence, term frequencies (tf), and term frequency and inverse document frequency (tf-idf) vectors.

N-grams: To capture contextual information, we experimented with bigrams and trigrams from the dataset. We collected all bigrams and trigrams and treated each one of them as a term. Our feature vector was then the tf-idf vector over these bigrams or trigrams. This may help capture word ordering differences among different dialects. Moreover, several n-grams only occur in certain dialects, which helps us create a more discriminating feature vector over the sentences in the dataset.

N-gram combinations: Finally, after noticing that all three of the previously computed features, unigrams, bigrams and trigrams provides its own advantage over the dataset, we decided to experiment with different combinations of these features, such as unigrams with bigrams, bigrams with trigrams, all three n-grams, etc. This resulted in a very high-dimensional feature vector.

3.2.2 Character level features

These features are more fine-grained than word level features, which would enable our models to learn morphological and utterance based features. Working with more fine-grained features was also shown to be useful in other natural language processing tasks such as machine translation (Sennrich et al., 2016). Character-based models have also been used in literature to convert Egyptian dialect to MSA in order to aid machine translation of Egyptian dialect (Sajjad et al., 2013; Durrani et al., 2014). Hence, this motivates the use of character level features for this task.

Character N-grams: Similar to word level features, we experimented with character-level bigrams, trigrams, 4-grams and 5-grams. The motivation behind this was drawn from word examples from different dialects that only differ in a few characters. The average word length in the dataset for the closed task is around 4.5 characters. Thus, we decided not to try values of n that are higher than 5.

Character N-gram combinations: Again, similar to word level features, we noticed that each of the n-gram features provided additional discriminating information for our classifiers, and hence we experimented with several combinations.

4 Results

As mentioned in section 3.1, we split the provided training set into training and dev splits. We report here our results on the dev split and on the official shared task runs. Tables 2 and 3 report on the accuracy of different experimental conditions with various learning algorithms and features on the dev set. The last column of Table 3 shows the performance of our best system. Our best system was trained on character bigrams, trigrams, 4-grams and 5-grams together. For this system, we also ignored all sentences of 3 words or less during training, as this has been shown to improve performance. As explained in section 3.1, shorter sentences in the corpus are not very discriminatory in this particular dataset. Hence, keeping them in the training corpus leads to sub-optimal performance. The linear SVM gave us the best results.

Table 4 shows the performance of our best model in the shared task. The baseline is based on the majority class, and our model performs significantly better than the baseline in the closed track. Figure 4 also shows the confusion matrix of our best model on the dev and official test sets. The best performance was achieved on the MSA class, while the worst was on the Gulf dialect. For the open track, our results were considerably poorer than those for the closed class even though we employed more training data. This could be explained by the significant difference in genre and lexical distribution between the task's training data and the AOC data.

More sophisticated models such as the SVM with a string kernel and a 2 Layer neural network did not perform as well as the linear SVM. This is potentially due to the limited size of the training set that does not allow the parameters to be adequately learned from the existing data to generalize as well.

	Raw Counts	**Term Frequencies**	**Unigrams TF-IDF**	**Bigrams TF-IDF**	**trigrams TF-IDF**	**1,2,3-grams TF-IDF**
Naive Bayes	0.5450	0.4339	0.4832	0.4504	0.3544	0.4734
Logistic Regression	0.5556	0.5227	0.5694	0.4523	0.3432	0.5082
SVM (linear)	0.5503	0.5457	**0.5976**	0.4931	0.3958	0.5700
2 Layer NN	0.5030	0.5312	0.5477	0.4536	0.3787	0.4845

Table 2: Accuracy on dev set with various word-level features

	Bigrams TF-IDF	**Trigrams TF-IDF**	**4-grams TF-IDF**	**5-grams TF-IDF**	**2,3,4,5-grams TF-IDF**	**Best system**
Naive Bayes	0.5030	0.5273	0.4668	0.4655	0.3702	0.3468
Logistic Regression	0.5654	0.6213	0.6318	0.6108	0.6377	0.6619
SVM (linear)	0.5378	0.6154	0.6410	0.6404	0.6588	**0.7007**
2 Layer NN	0.5352	0.6062	0.5845	0.5819	0.6009	0.6237

Table 3: Accuracy on dev set with various character-level features

Test Set	**Track**	**Run**	**Accuracy**	**F1 (micro)**	**F1 (macro)**	**F1 (weighted)**
C	-	baseline	0.2279	-	-	-
C	closed	run1	**0.5136**	**0.5136**	**0.5091**	**0.5112**
C	closed	run2	0.5117	0.5117	0.5023	0.5065
C	open	run1	0.3792	0.3792	0.3462	0.352
C	open	run2	0.3747	0.3747	0.3371	0.3413

Table 4: Results for all runs on the hidden test set.

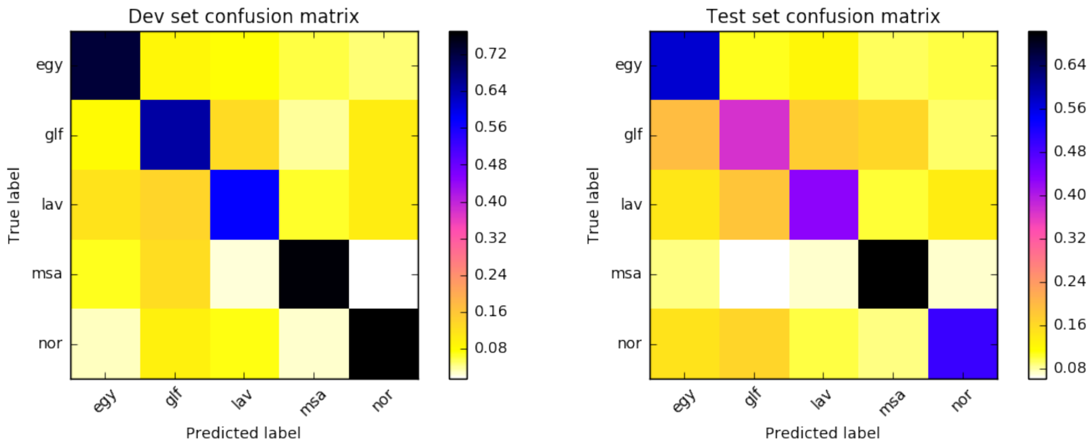
5 Conclusion

In this paper, we described our Arabic dialect detection system that we used to submit four runs to sub-task 2 of the 2016 DSL shared task, which involves the automatic identification of 5 Arabic variants, namely Egyptian, Gulf, Levantine, North-African, and MSA. The training data for the sub-task at hand differs from data used in the literature in two ways, namely:

- The training data is relatively small,
- the training data is composed of ASR output, which makes the data difficult to work with.

For classification, we tried several machine-learning models. Our best performing model used an SVM classifier with a linear kernel that is trained on combined character n-gram where n = 1, 2, 3, 4, and 5 with tf-idf weighting. In the closed submission, we got the best accuracy of 0.5136 and the third best weighted F1 score, with a difference of less than 0.002 from the best system.

For future work, we plan to apply more powerful techniques, such as recurrent neural networks over both words and characters to capture the differences between the dialects better. We will be using larger datasets, since these models usually require large amounts of data to perform well.



(a) Development set

(b) Test set

Figure 1: Confusion matrix for our best model

References

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 *Interspeech 2016*, pages 2934–2938.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of arabic. In *LREC*, pages 1240–1245.

Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. 2014. Verifiably effective arabic dialect identification. In *EMNLP*, pages 1465–1468.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. Colaba: Arabic dialect annotation and processing. In *Lrec workshop on semitic language processing*, pages 66–74.

Nadir Durrani, Yaser Al-Onaizan, and Abraham Ittycheriah. 2014. Improving egyptian-to-english smt by mapping egyptian into msa. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 271–282. Springer.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. 2015. Arabic Dialect Identification using a Parallel Multidialectal Corpus. In *Proceedings of the 14th Conference of the Pacific Association for Computational Linguistics (PACLING 2015)*, pages 209–217, Bali, Indonesia, May.

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.

Hamdy Mubarak and Kareem Darwish. 2014. Using twitter to collect a multi-dialectal corpus of arabic. *ANLP 2014*, page 1.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal Arabic to English. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '13, pages 1–6, Sofia, Bulgaria.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Omar F Zaidan and Chris Callison-Burch. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 37–41. Association for Computational Linguistics.

Omar F Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.