

Deep Models for Arabic Dialect Identification on Benchmarked Data

Mohamed Elaraby Muhammad Abdul-Mageed

Natural Language Processing Lab

University of British Columbia

mohamed.elaraby@alumni.ubc.ca, muhammad.mageed@ubc.ca

Abstract

The Arabic Online Commentary (AOC) (Zaidan and Callison-Burch, 2011) is a large-scale repository of Arabic dialects with manual labels for 4 varieties of the language. Existing dialect identification models exploiting the dataset pre-date the recent boost deep learning brought to NLP and hence the data are not benchmarked for use with deep learning, nor is it clear how much neural networks can help tease the categories in the data apart. We treat these two limitations: We (1) benchmark the data, and (2) empirically test 6 different deep learning methods on the task, comparing performance to several classical machine learning models under different conditions (i.e., both binary and multi-way classification). Our experimental results show that variants of (attention-based) bidirectional recurrent neural networks achieve best accuracy (acc) on the task, significantly outperforming all competitive baselines. On blind test data, our models reach 87.65% acc on the binary task (MSA vs. dialects), 87.4% acc on the 3-way dialect task (Egyptian vs. Gulf vs. Levantine), and 82.45% acc on the 4-way variants task (MSA vs. Egyptian vs. Gulf vs. Levantine). We release our benchmark for future work on the dataset.

1 Introduction

Dialect identification is a special type of language identification where the goal is to distinguish closely related languages. Explosion of communication technologies and the accompanying pervasive use of social media strongly motivates need for technologies like language, and dialect, identification. These technologies are useful for applications ranging from monitoring health and well-being (Yepes et al., 2015; Nguyen et al., 2016; Nguyen et al., 2017; Abdul-Mageed et al., 2017), to real-time disaster operation management (Sakaki et al., 2010; Palen and Hughes, 2018), and analysis of human mobility (Hawelka et al., 2014; Jurdak et al., 2015; Louail et al., 2014). Language identification is also an enabling technology that can help automatically filter foreign text in some tasks (Lui and Baldwin, 2012), acquire multilingual data (e.g., from the web) (Abney and Bird, 2010), including to enhance tasks like machine translation (Ling et al., 2013).

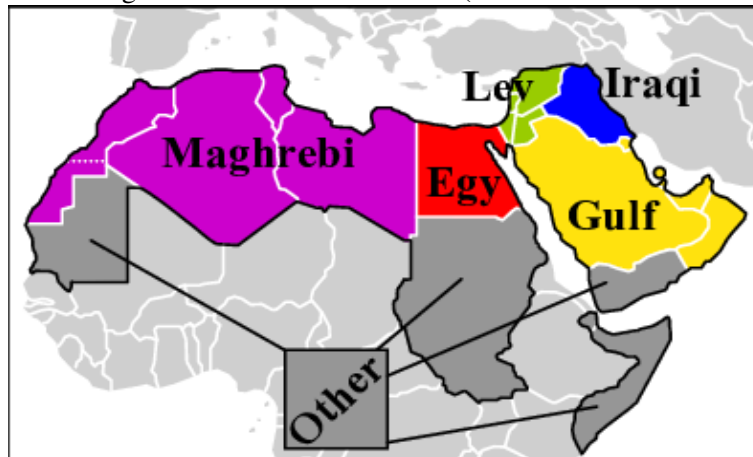
Arabic. In this paper our focus is on *Arabic*, a term that refers to a wide collection of varieties. These varieties are the result of the interweave between the native languages of the Middle East and North Africa and Arabic itself. Modern Standard Arabic (MSA), the modern variety of the language used in pan-Arab news outlets like AlJazeera and in educational circles in the Arab world, differs phonetically, phonologically, lexically, and syntactically from the varieties spoken in everyday communication by native speakers of the language (Diab et al., 2010; Habash, 2010; Abdul-Mageed, 2015; Abdul-Mageed, 2017). These ‘everyday’ varieties constitute the *dialects* of Arabic. Examples of these are Egyptian (EGY), Gulf (GLF), Levantine (LEV), and Moroccan (MOR). In addition to MSA and dialects, Classical Arabic also exists and is the variety of historical literary texts and religious discourse.

Arabic Dialects. Language varieties, including those of Arabic, can be categorized based on shared linguistic features. For Arabic, one classical categorization is based on geographical locations. For example, in addition to MSA, Habash et al. (2012), provides 5 main categories, as shown in Figure 1. This same classification is also common in the literature, and includes:

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>.

- **Egyptian:** The variety spoken in Egypt, which is widely spread due to the historical impact of Egyptian media
- **Gulf:** A variety spoken primarily in Saudi Arabia, UAE, Kuwait and Qatar
- **Iraqi:** The variety spoken by the people of Iraq
- **Levantine:** The variety spoken primarily by the Levant (i.e., people of Syria, Lebanon, and Palestine)
- **Maghrebi:** The variety spoken by people of North Africa, excluding Egypt

Figure 1: One categorization of Arabic dialects (Zaidan and Callison-Burch, 2011)



Arabic dialectal data. For a long time, Arabic dialects remained mostly spoken. Dialects started to find their way in written form with the spread of social media, thus affording an opportunity for researchers to use these data for NLP. This motivated Zaidan and Callison-Burch (2014) to create a large-scale repository of Arabic texts, the Arabic Online Commentary (AOC). The resource is composed of $\sim 3\text{M}$ MSA and dialectal comments on a number of Arabic news sites. A portion of the data ($> 108\text{K}$ comments) is manually annotated via crowdsourcing. The dataset was exploited for dialect identification in Zaidan and Callison-Burch (2014) and later in Cotterell and Callison-Burch (2014). These works, however, pre-date the current boom in NLP where deep neural networks enable better learning (given sufficiently large training data). Cotterell and Callison-Burch (2014) use n -fold cross validation in their work, thus making it costly to adopt the same data split procedure to develop deep learning models. This is the case since deep models can take long times to train and optimize. For this reason, it is desirable to benchmark the AOC dataset for deep learning research. This motivates our work. We also ask the empirical question: To what extent can we tease apart the Arabic varieties in AOC using neural networks. Especially given (a) the morphological richness of Arabic and (b) the inter-relatedness (e.g., lexical overlap) between Arabic varieties, it is not clear how accurately these varieties can be automatically categorized (using deep learning methods). To answer these important questions, we investigate the utility of several traditional machine learning classifiers and 6 different deep learning models on the task. Our deep models are based on both recurrent neural networks and convolutional neural networks, as well as combinations (and variations) of these.

Overall, we offer the following contributions: (1) We benchmark the AOC dataset, especially for deep learning work, (2) we perform extensive experiments based on deep neural networks for identifying the 4 Arabic varieties in AOC under various classification conditions, allowing us to perform well on the task, and (3) we carry out an analysis to uncover how the varieties in the data relate to one another based on shared lexica. The rest of the paper is organized as follows: In Section 2 we review related work, in Section 3 we briefly describe the AOC dataset. In Section 4 we describe our models, Section 5 is

Variety	Example
MSA	<p>شذوا المهمة يا منتخبنا ، انتم لا تسرون وحدكم (1) Go, go, our team; you've our passionate support. بصراحة لقد عجزت عن الكلام فلا الجبل يبرر هذه الفعلة ولا حتى اي سبب اخر (2) Frankly, I'm speechless. Neither ignorance nor any other reason justify this action.</p>
EGY	<p>بتقولوا ان اعلامنا وفضائيتنا نجروا الازمة وكانوا مجرمين في نظركم (3) You say our media and satellite channels initiated the crisis and were criminals in your review. قمر صناعي ولا قمر ١٤ ، في الشمس لو عرف القمر يلف في الجو (3) Either its a satellite or a full moon [playful for "beautiful female"], it will never rotate in its orbit correctly.</p>
GLF	<p>يعطي العافية على من اقترح هذا القرار ومن ساهم في تطبيقه (5) Healthy be the one who proposed this decision, and those who contributed in applying it. عندنا بعض الناس ما يرضيهم شيء ولا يكفيهم ولا يبغون يشتغلون شيء (6) Nothing would please nor be enough for some of these people; they don't even want to put any efforts.</p>
LEV	<p>وانا كمان لا يصلح سيارتي ولا على بابي وابوي واخوي بيتكفلوا فيها (7) And I also won't repair my car, nor do I care. My brother and dad will take care of it. لان مو معقول توصل الامور لهلدرجة وكانت رح تصير مليون مشكلة بسبب اشاعاتهم لطلب الجامعة (8) Because it isn't reasonable for things to get to that bad. There is a million problems college students have because of their rumors.</p>

Table 1: Example comments from the 4 varieties in the AOC dataset

where we describe our experimental set up, Section 6 provides our experimentation results. Section 7 is a visualization-based analysis of our results. Section 8 is where we conclude our work and overview future directions.

2 Related work

Work on Arabic dialect identification has focused on both spoken (Ali et al., 2015; Belinkov and Glass, 2016; Najafian et al., 2018; Shon et al., 2018; Shon et al., 2017; Najafian et al., 2018) and written form (Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2014; Cotterell and Callison-Burch, 2014; Darwish et al., 2014; Abdul-Mageed et al., 2018). Early works have focused on distinguishing between MSA and EGY. For example, Elfardy and Diab (2013) propose a supervised method for sentence-level MSA-EGY categorization, exploiting a subset of the AOC dataset (12, 160 MSA sentences and 11, 274 of user commentaries on Egyptian news articles). The authors study the effect of pre-processing on classifier performance, which they find to be useful under certain conditions. Elfardy and Diab (2013) report 85.5% accuracy using 10-fold cross-validation with an SVM classifier, compared to the 80.9% accuracy reported by Zaidan and Callison-Burch (2011). Similarly, Tillmann et al. (2014) exploit the same portion of the AOC data Elfardy and Diab (2013) worked on, to build an MSA-EGY classifier. The authors report an improvement of 1.3% over results acquired by Zaidan and Callison-Burch (2014) using a linear classifier utilizing an expanded feature set. Their features include n-grams defined via part of speech tags and lexical features based on the AIDA toolkit (Elfardy et al., 2014). The work of Darwish et al. (2014) is also similar to these works in that it also focuses on the binary MSA-EGY classification task, but the authors exploit Twitter data. More specifically, Darwish et al. (2014) collected a dataset of 880K tweets on which they train their system, while testing on 700 tweets they labeled for the task. The authors explore a range of lexical and morphological features and report a 10% absolute gain over models trained with n-grams only. Our work is similar to these works in that we exploit the AOC dataset and develop MSA-EGY, binary classifiers. However, we model the task at more fine-grained levels as well (i.e., 3-way and 4-way classification).

Huang (2015) focus on the 4-way classification task using the AOC categories (MSA, EGY, GLF, LEV). The authors report improved classification accuracy using a simple word-level n-gram model trained on the manually annotated portion of AOC as well as unannotated Facebook data. The authors employ an ensemble of co-training and self-training semi-supervised learning methods exploiting 165M data points from Facebook posts. Huang (2015) report an accuracy of 87.8% on 10% of the manually annotated AOC dataset. Our work is similar to Huang (2015) in that we consider the 4-way classification task, but we do not exploit any external data. In addition, Huang (2015) did not release their data splits,

nor benchmark the task on AOC. Their results are not directly comparable to our work for these reasons.

Finally, our work has some similarity to general works on language detection (Jurgens et al., 2017; Jauhiainen et al., 2017; Kocmi and Bojar, 2017; Jauhiainen et al., 2018) and geographical location (Rahimi et al., 2018; Mahmud et al., 2014; Rahimi et al., 2017).

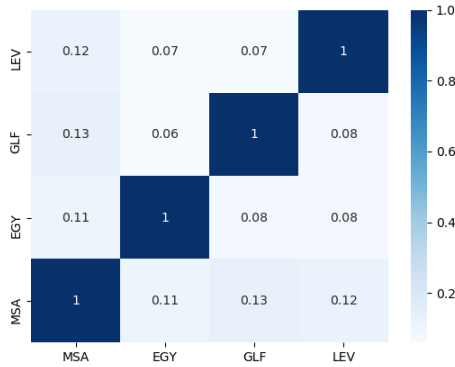
3 Dataset: Arabic Online Commentary (AOC)

As we mentioned earlier, our work is based on the AOC dataset. AOC is composed of 3M MSA and dialectal comments, of which 108,173 comments are labeled via crowdsourcing. For our experiments, we randomly shuffle the dataset and split it into 80% training (*Train*), 10% validation (*Dev*), and 10% test (*Test*). Table 2 shows the distribution of the data across the different splits. We were interested in identifying how the 4 varieties relate to one another in terms of their shared vocabulary, and so we performed an analysis on the training split (*Train*) as shown in the heat map in Figure 2. The Figure presents the percentages of shared vocabulary between the different varieties after normalizing for the number of data points in each class. As the Figure shows, both the GLF and LEV dialects are lexically closer to (i.e., share more vocabulary with) MSA than EGY is (does). This finding is aligned with the intuition of native speakers of Arabic that EGY diverges more from MSA than the GLF and LEV varieties. This empirical finding lends some credibility to this intuition.

Table 2: Distribution of classes in our AOC *Train* split

Variety	MSA	EGY	GLF	LEV	ALL
Train	50,845	10,022	16,593	9,081	86,541
Dev	6,357	1,253	2,075	1,136	10,821
Test	6,353	1,252	2,073	1,133	10,812

Figure 2: Heat map for shared vocabulary between different data variants



4 Models

4.1 Traditional models

Traditional models refer to models based on feature engineering methods with linear and probabilistic classifiers. In our experiments, we use (1) logistic regression, (2) multinomial Naive Bayes, and (3) support vector machines (SVM) classifiers.

4.2 Deep Learning Models

Recently, deep learning models have been successfully applied to the tasks of language modeling and text classification. For these reasons, we experiment with a number of popular models, as follows: (1) convolutional neural networks (CNN), (2) long-short term memory (LSTM), (3) convolutional LSTM (CLSTM), (4) bidirectional LSTM (BiLSTM), (5) bidirectional gated recurrent units (BiGRU), and (6)

BiLSTM with attention. While there are other variations of how some of these models learn (Vaswani et al., 2017), we believe these models with the variations we exploit form a strong basis for our benchmarking objective. In all our models, we use pre-trained word vectors based on word2vec to initialize the networks. We then fine-tune weights during learning.

Word-based Convolutional Neural Network (CNN): This model is conceptually similar to the one described in Kim (2014), and has the following architecture:

- *Input layer:* an input layer to map word sequence w into a sequence vector \mathbf{x} where x_w is a real-valued vector ($x_w \in \mathbb{R}^{d_{emb}}$, with $d_{emb} = 300$ in all our models) initialized from external embedding model and tuned during training. The embedding layer is followed by a dropout rate of 0.5 for regularization (in this case to prevent co-adaptation between hidden units).
- *Convolution layer:* Two 1-D convolution operations are applied in parallel to the input layer to map input sequence \mathbf{x} into a hidden sequence \mathbf{h} . A filter $k \in \mathbb{R}^{w_{d_{emb}}}$ is applied to a window of concatenated word embedding of size w to produce a new feature c_i . Where $c_i \in \mathbb{R}$, $c_i = k \odot x_{i:i+w-1} + b$, b is the bias $b \in \mathbb{R}$, and $x_{i:i+w-1}$ is a concatenation of x_i, \dots, x_{i+w-1} . The filter sizes used are 3 and 8 and the number of filters used is 10. After each convolution operation a non-linear activation of type Rectifier Linear Unit (ReLU) (Nair and Hinton, 2010) is applied. Finally different convolution outputs (the two convolutional maps in our case) are concatenated into a sequence $\mathbf{c} \in \mathbb{R}^{n-h+1}$ (where n is the number of filters and h is the dimensionality of the hidden sequence) and passed to a pooling layer.
- *Maxpooling:* Temporal max-pooling, which is the 1-D version of pooling, is applied over the concatenated output of the multiple convolutions \mathbf{c} , as mentioned above. The sequence \mathbf{c} is converted into a single hidden vector \mathbf{c}' by taking the maximum values of extracted feature map $\mathbf{c}' = \max\{\mathbf{c}\}$. The size of \mathbf{c}' is $\sum_i n_i w_i$ where n_i is the number of filters and w_i the width of these filters.
- *Dense layer:* A 100 dimension fully-connected layer with a ReLU non-linear activation is added to map vector \mathbf{c}' into a final vector \mathbf{c}'' . For regularization, we employ a dropout rate of 0.8 and an l2-norm.
- *Softmax layer:* Finally, the hidden units \mathbf{c}'' is converted into probability distribution over l via softmax function, where l is the number of classes.

Long-Short Term Memory (LSTM): In our experiments, we use different variations of recurrent neural networks. The first one is LSTM (Hochreiter and Schmidhuber, 1997). We use a word-based LSTM, with the following architecture:

- *Input layer:* The input layer is exactly the same as the one described in the CNN model above.
- *LSTM layer:* We use a vanilla LSTM architecture consisting of 100 dimensions hidden units. The LSTM is designed to capture long-term dependencies via augmenting a standard RNN with a memory state C_t , with $C_t \in \mathbb{R}$ at time step t . The LSTM takes in a previous state h_{t-1} and input x_t , to calculate the hidden state h_t as follows:

$$\begin{aligned}
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
 C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
 h_t &= o_t \odot \tanh(C_t)
 \end{aligned} \tag{1}$$

where σ is the sigmoid, \tanh is the hyperbolic tangent function, and \odot is the dot product between two vectors. The i_t , f_t , o_t are the *input*, *forget*, and *output* gates, and the \tilde{C}_t is a new memory cell

vector with candidates that could be added to the state. We use the same regularization as we apply on the dense layer in the CNN model mentioned above.

- *Softmax layer*: Similar to that of the CNN above as well.

Convolution LSTM (CLSTM): This model is described in Zhou et al. (2015). The model architecture is similar to the CNN described earlier, but the fully-connected (dense) layer is replaced by an LSTM layer. The intuition behind the CLSTM is to use the CNN layer as a feature extractor, and directly feed the convolution output to the LSTM layer (which can capture long-term dependencies).

Bidirectional LSTM (BiLSTM): One limitation of conventional RNNs is that they are able to make predictions based on previously seen content only. Another variant of RNNs that addresses this problem is Bidirectional RNNs (BRNNs), which process the data in both directions in two separate hidden layers. These two hidden layers are then fed forward to the same output layer. BRNNs compute three sequences; a forward hidden sequence \vec{h} , a backward hidden sequence \overleftarrow{h} , and the output sequence y . The model transition equations are as below:

$$\begin{aligned}\vec{h} &= \mathcal{H}(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t+1} + b_{\vec{h}}) \\ \overleftarrow{h} &= \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \\ y_t &= W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y\end{aligned}\quad (2)$$

Where \mathcal{H} can be any activation function. Combining BRNN and LSTM gives the BiLSTM model. In our experiments we use a 100 hidden units dimension to ensure a fair comparison with LSTM's results. We apply the same regularization techniques applied for the LSTM layer described above.

Bidirectional Gated Recurrent Units (BiGRU): Gated Recurrent Unit (GRU) (Chung et al., 2014) is a variant of LSTMs that combines the *forget* and *input* gates into a single update gate z_t by primarily merging the *cell* state and *hidden* state. This results in a simpler model composed of an *update* state z_t , a *reset* state r_t , and a new simpler *hidden* state h_t . The model transition equations are as follows:

$$\begin{aligned}z_t &= \sigma(W_z.[h_{t-1}, x_t]) \\ r_t &= \sigma(W_r.[h_{t-1}, x_t]) \\ \tilde{h}_t &= \tanh(W.[r_t * h_{t-1}, x_t]) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t\end{aligned}\quad (3)$$

The Bidirectional GRU (BiGRUs) can be obtained by combining two GRUs, each looking at a different direction similar to the case of BiLSTMs above. We employ the same regularization techniques applied to the LSTM and BiLSTM networks.

Attention-based BiLSTM

Recently, using an attention mechanism with a neural networks has resulted in notable success in a wide range of NLP tasks, such as machine translation, speech recognition, and image captioning (Bahdanau et al., 2014; Xu et al., 2015; Chorowski et al., 2015). In this section, we describe an attention mechanism that we employ in one of our models (BiLSTM) that turned out to perform well without attention, hoping the mechanism will further improve model performance. We use a simple implementation inspired by Zhou et al. (2016) where attention is applied to the output vector of the LSTM layer. If H is a matrix consisting of output vectors $[h_1, h_2, ..h_T]$ (where T is the sentence length), we can compute the attention vector α of the sequence as follows:

$$\begin{aligned}\mathbf{e}_t &= \tanh(h_t) \\ \alpha_t &= \frac{\exp(e_t)}{\sum_{i=1}^T \exp(e_i)}\end{aligned}\quad (4)$$

Finally, the representation vector for input text \mathbf{v} is computed by a weighted summation over all the time steps, using obtained attention scores as weights.

$$\mathbf{v} = \sum_{i=1}^T \alpha_i h_i \quad (5)$$

The vector \mathbf{v} is an encoded representation of the whole input text. This representation is passed to the *softmax* layer for classification.

5 Experiments

We perform 3 different classification tasks: (A) *binary classification*, where we tease apart the MSA and the dialectal data, (B) *3-way dialects*, where we attempt to distinguish between EGY, GLF, and LEV; and (C) *4-way variants* (i.e., MSA vs. EGY vs. GLF vs. LEV). Since our goal in this work is to explore how several popular traditional settings and deep learning model architectures fare on the dialect identification task, we use classifiers with pre-defined hyper-parameters inspired by previous works as described in Section 4. As we mention in Section 3, we split the data into 80% *Train*, 10% *Dev*, and 10% *Test*. While we train on *Train* and report results on both the *Dev* and *Test* sets in the current work, our goal is to invest on hyper-parameter tuning based on the development set in the future. Benchmarking the data is thus helpful as it facilitates comparisons in future works.¹

5.1 Pre-processing

We process our data the same way across all our traditional and deep learning experiments, as follows:

- **Tokenization and normalization:** We tokenize our data based on white space, excluding all non-unicode characters. We then normalize *Alif maksura* to *Ya*, reduce all *hamzated Alif* to plain *Alif*, and remove all non-Arabic characters/words (e.g., “very”, “50\$”).
- **Input sequence quantization:** In our experiments, we fix the vocabulary at the most frequent 50K words. The input tokens are then converted into indices ranging from 1 to 50K based on our look-up vocabulary.
- **Padding:** For the deep learning classifiers, all input sequences are truncated to arbitrary maximum sequence length of 30 words per comment. Comments of length < 30 are zero-padded. This number can be tuned in future work.

5.2 Traditional Classifier Experiments

We have two settings for the traditional classifiers: (1) presence vs. absence (0 vs. 1) vectors based on combinations of unigrams, bigrams, and trigrams; and (2) term-frequency inverse-document-frequency (TF-IDF) vectors based on combinations of unigrams, bigrams, and trigrams (Sparck Jones, 1972). We use scikit-learn’s (Pedregosa et al., 2011) implementation of these classifiers.

5.3 Deep Learning Experiments

All our deep models are trained for 10 epochs using the RMSprop optimizer. The model’s weights W are initialized from a normal distribution $W \sim N$ with a small standard deviation of $\sigma = 0.05$. Our models are trained using the Keras (Chollet and others, 2015) library with a Tensorflow (Abadi et al., 2016) backend. We train each of our 6 deep learning classifiers across 3 different settings pertaining the way we initialize the embeddings for the input layer in each network. The three embedding settings are:

1. **Random embeddings:** Where we initialize the input layer randomly.
2. **AOC-based embeddings:** We make use of the $\sim 3M$ unlabeled comments in AOC by training a “continuous bag of words” (CBOW) (Mikolov et al., 2013) model exploiting them. We adopt the settings in Abdul-Mageed et al. (2018) for training our model to acquire 300 dimensional word vectors.

¹The benchmarked data can be obtained by emailing the authors. See also project repository at: https://github.com/UBC-NLP/aoc_id.

Method	Binary		Three-way		Four-way	
	<i>Dev</i>	<i>Test</i>	<i>Dev</i>	<i>Test</i>	<i>Dev</i>	<i>Test</i>
Traditional Classifiers						
Baseline (majority class in <i>Train</i>)	58.75	58.75	58.75	58.75	46.49	46.49
Logistic Regression (1+2+3 grams)	84.18	83.71	86.91	85.75	75.75	78.24
Naive Bayes (1+2+3 grams)	84.97	84.53	87.51	87.81	80.15	77.75
SVM (1+2+3 grams)	82.79	82.41	85.51	84.27	74.5	75.82
Logistic Regression (1+2+3 grams TF-IDF)	83.96	83.24	86.71	85.51	75.81	78.24
Naive Bayes (1+2+3 grams TF-IDF)	83.52	82.91	86.61	86.87	73.21	75.81
SVM (1+2+3 grams TF-IDF)	84.07	83.61	86.76	85.93	76.65	78.61
Deep Learning - Random Embeddings						
CNN (Kim, 2014)	85.69	85.16	81.63	81.11	66.34	68.86
CLSTM (Zhou et al., 2015)	84.73	84.17	78.91	78.32	64.58	65.25
LSTM	85.41	85.28	78.61	78.51	70.21	68.71
BiLSTM	84.11	83.77	85.82	84.99	75.94	77.55
BiGRU	82.81	82.77	84.88	84.45	74.56	76.51
Attention-BiLSTM	85.5	85.23	86.12	85.93	79.97	80.21
Deep Learning - AOC Embeddings						
CNN (Kim, 2014)	85.02	84.51	76.81	76.53	64.23	64.17
CLSTM (Zhou et al., 2015)	85.17	84.73	76.81	75.71	64.61	63.89
LSTM	85.04	84.07	83.89	82.67	70.01	68.91
BiLSTM	85.33	84.88	86.21	86.01	76.12	78.35
BiGRU	85.39	85.27	86.92	86.57	79.61	80.11
Attention-BiLSTM	85.77	85.71	87.01	86.93	80.25	81.12
Deep Learning - Twitter-City Embeddings (Abdul-Mageed et al., 2018)						
CNN (Kim, 2014)	86.68	86.26	85.51	85.36	74.13	75.61
CLSTM (Zhou et al., 2015)	86.61	86.28	82.77	82.56	79.41	77.51
LSTM	85.52	85.07	84.41	84.61	75.21	78.53
BiLSTM	87.16	86.99	87.31	87.11	82.81	81.93
BiGRU	87.65	87.23	87.11	86.18	83.25	82.21
Attention BiLSTM	87.61	87.21	87.81	87.41	83.49	82.45

Table 3: Experimental results, in accuracy, on our *Dev* and *Test* AOC splits

- Twitter-City embeddings:** This is based on the CBOW word2vec model released by Abdul-Mageed et al. (2018). The authors train their models on a $\frac{1}{4}$ billion tweets dataset collected from 29 different cities from 10 Arab countries. The authors use a window of size 5 words, minimal word frequency set at 100 words, and 300 dimensional word vectors to train this model.

6 Results

Table 3 shows our results in accuracy across the three classification tasks (i.e., *binary*, *3-way*, and *4-way*), as described in Section 5. Our baseline in each task is the majority class in the respective *Train* set. As Table 6 shows, among traditional models, the Naive Bayes classifier achieves the best performance across all three tasks both on *Dev* and *Test* data. As a sole exception, SVMs outperforms Naive Bayes on the *Test* set for the 4-way classification task. As best accuracy, traditional classifiers yield 84.53 (*binary*), 87.81 (*3-way*), and 78.61 (*4-way*) on the *Test* splits.

As Table 3 shows, across the different classification tasks, models initialized with the Twitter-City embeddings (Abdul-Mageed et al., 2018) perform best on the task compared to those initialized randomly or

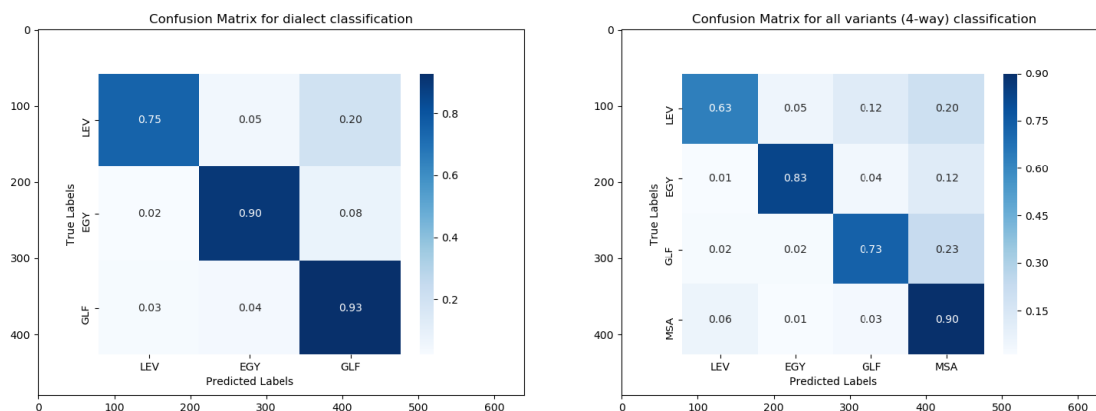
with the AOC embeddings. We also observe that AOC embeddings are better than random initialization. For **binary classification**, as Table 6 shows, BiGRU obtains the best accuracy on both *Dev* (87.65) and *Test* (87.23), with attention-BiLSTM performing quite closely. For the **3-way classification** task (dialect only classifier), while attention-BiLSTM obtains the best result on the *Dev* (87.81), it is slightly outperformed by the Naive Bayes classifier on *Test* (also 87.81). For **4-way classification**, attention-BiLSTM obtained best accuracy on both *Dev* (83.49) and *Test* (82.42).

Aligned with knowledge about deep models, we note the positive effect of larger training data on classification. For example, when we reduce the size of *Train* by excluding the MSA comments (58% of the manually annotated data), traditional classifiers outperform most of the deep learning classifiers on 3-way classification. Similarly, results drop when we thinly spread the data across the 4 categories for 4-way classification.

7 Analysis

Figure 3 is a visualization of classification errors acquired with attention-BiLSTM results (best accuracy in the multi-class tasks). The *left-side (3-way/dialects)* matrix shows how LEV is confused 20% of the time with GLF, directly reflecting the closer lexical distance between the two varieties compared to the distance of either of them to EGY. The *right-side (4-way)* matrix shows that 23% of the GLF errors are confused with MSA, followed by LEV errors (confused with MSA 20% of the time). This is a result of the higher lexical overlap between the two dialects and MSA, as we described in our observations around Figure 2. As Table 2 shows, MSA also dominates *Train* and hence these confusions with MSA are expected.

Figure 3: Analysis of Attention-BiLSTM results. **Left:** Confusion matrix for 3-way predictions. **Right:** Confusion matrix for 4-way classification.



8 Conclusion

We benchmarked the AOC dataset, a popular dataset of Arabic online comments, for deep learning work focused at dialect identification. We also developed 12 different classifiers (6 traditional and 6 based on deep learning) to offer strong baselines for the task. Results show attention-based BiLSTMs to work well on this task, especially when initialized using a large dialect specific word embeddings model. In the future, we plan to exploit sub-word and further tune hyper-parameters of our models.

9 Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). This research was also supported by the Social Sciences and Humanities Research Council of Canada (SSHRC). The research was enabled in part by support provided by WestGrid (<https://www.westgrid.ca/>) and Compute Canada (www.computecanada.ca). We thank Omar Zaidan and Chris Callison-Burch for sharing the AOC data.

References

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.
- Muhammad Abdul-Mageed, Anneke Buffone, Hao Peng, Johannes C Eichstaedt, and Lyle H Ungar. 2017. Recognizing pathogenic empathy in social media. In *ICWSM*, pages 448–451.
- Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of arabic dialects. In *LREC*, pages 3653–3659.
- Muhammad Abdul-Mageed. 2015. *Subjectivity and sentiment analysis of Arabic as a morphologically-rich language*. Ph.D. thesis, Indiana University.
- Muhammad Abdul-Mageed. 2017. Modeling arabic subjectivity and sentiment in lexical space. *Information Processing & Management*.
- Steven Abney and Steven Bird. 2010. The human language project: building a universal corpus of the world’s languages. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 88–97. Association for Computational Linguistics.
- Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals. 2015. Automatic dialect detection in arabic broadcast speech. *arXiv preprint arXiv:1509.06928*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Yonatan Belinkov and James Glass. 2016. A character-level convolutional neural network for distinguishing similar languages and dialects. *arXiv preprint arXiv:1609.07568*.
- François Chollet et al. 2015. Keras.
- Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Ryan Cotterell and Chris Callison-Burch. 2014. A multi-dialect, multi-genre corpus of informal written arabic. In *LREC*, pages 241–245.
- Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. 2014. Verifiably effective arabic dialect identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (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.
- Heba Elfardy and Mona Diab. 2013. Sentence level dialect identification in arabic. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 456–461.
- Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2014. Aida: Identifying code switching in informal arabic text. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 94–101.
- Nizar Habash, Mona T Diab, and Owen Rambow. 2012. Conventional orthography for dialectal arabic. In *LREC*, pages 711–718.
- Nizar Y Habash. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.
- Bartosz Hawelka, Izabela Sitko, Euro Beinart, Stanislav Sobolevsky, Pavlos Kazakopoulos, and Carlo Ratti. 2014. Geo-located twitter as proxy for global mobility patterns. *Cartography and Geographic Information Science*, 41(3):260–271.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

- Fei Huang. 2015. Improved arabic dialect classification with social media data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2118–2126.
- Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2017. Evaluation of language identification methods using 285 languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 183–191.
- Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic language identification in texts: A survey. *arXiv preprint arXiv:1804.08186*.
- Raja Jurdak, Kun Zhao, Jiajun Liu, Maurice AbouJaoude, Mark Cameron, and David Newth. 2015. Understanding human mobility from twitter. *PloS one*, 10(7):e0131469.
- David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 51–57.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Tom Kocmi and Ondřej Bojar. 2017. Lanidenn: Multilingual language identification on character window. *arXiv preprint arXiv:1701.03338*.
- Wang Ling, Guang Xiang, Chris Dyer, Alan Black, and Isabel Trancoso. 2013. Microblogs as parallel corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 176–186.
- Thomas Louail, Maxime Lenormand, Oliva G Cantu Ros, Miguel Picornell, Ricardo Herranz, Enrique Frias-Martinez, José J Ramasco, and Marc Barthelemy. 2014. From mobile phone data to the spatial structure of cities. *Scientific reports*, 4:5276.
- Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics.
- Jalal Mahmud, Jeffrey Nichols, and Clemens Drews. 2014. Home location identification of twitter users. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):47.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.
- Maryam Najafian, Sameer Khurana, Suwon Shon, Ahmed Ali, and James Glass. 2018. Exploiting convolutional neural networks for phonotactic based dialect identification. ICASSP.
- Quynh C Nguyen, Dapeng Li, Hsien-Wen Meng, Suraj Kath, Elaine Nsoesie, Feifei Li, and Ming Wen. 2016. Building a national neighborhood dataset from geotagged twitter data for indicators of happiness, diet, and physical activity. *JMIR public health and surveillance*, 2(2).
- Quynh C Nguyen, Kimberly D Brunisholz, Weijun Yu, Matt McCullough, Heidi A Hanson, Michelle L Litchman, Feifei Li, Yuan Wan, James A VanDerslice, Ming Wen, et al. 2017. Twitter-derived neighborhood characteristics associated with obesity and diabetes. *Scientific reports*, 7(1):16425.
- Leysia Palen and Amanda L Hughes. 2018. Social media in disaster communication. In *Handbook of Disaster Research*, pages 497–518. Springer.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.
- Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2017. A neural model for user geolocation and lexical dialectology. *arXiv preprint arXiv:1704.04008*.
- Afshin Rahimi, Trevor Cohn, and Tim Baldwin. 2018. Semi-supervised user geolocation via graph convolutional networks. *arXiv preprint arXiv:1804.08049*.

- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM.
- Suwon Shon, Ahmed Ali, and James Glass. 2017. Mit-qcri arabic dialect identification system for the 2017 multi-genre broadcast challenge. *arXiv preprint arXiv:1709.00387*.
- Suwon Shon, Ahmed Ali, and James Glass. 2018. Convolutional neural networks and language embeddings for end-to-end dialect recognition. *arXiv preprint arXiv:1803.04567*.
- Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.
- Christoph Tillmann, Saab Mansour, and Yaser Al-Onaizan. 2014. Improved sentence-level arabic dialect classification. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 110–119.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.
- Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794.
- Antonio Jimeno Yepes, Andrew MacKinlay, and Bo Han. 2015. Investigating public health surveillance using twitter. *Proceedings of BioNLP 15*, pages 164–170.
- 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.
- Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.
- Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.