Country-level Arabic Dialect Identification Using Small Datasets with Integrated Machine Learning Techniques and Deep Learning Models

Maha J. Althobaiti
Department of Computer Science, Taif University, KSA
maha.j@tu.edu.sa

Abstract

Arabic is characterised by a considerable number of varieties including spoken dialects. In this paper, we presented our models developed to participate in the NADI subtask 1.2 that requires building a system to distinguish between 21 country-level dialects. We investigated several classical machine learning approaches and deep learning models using small datasets. We examined an integration technique between two machine learning approaches. Additionally, we created dictionaries automatically based on Pointwise Mutual Information and labelled datasets, which enriched the feature space when training models. A semi-supervised learning approach was also examined and compared to other methods that exploit large unlabelled datasets, such as building pre-trained word embeddings. Our winning model was the Support Vector Machine with dictionary-based features and Pointwise Mutual Information values, achieving an 18.94% macros-average F1-score.

1 Introduction

Arabic Dialect Identification (ADI) no doubt caught the attention of researchers (Althobaiti, 2020a,b). Recently, many events and campaigns have been organised to handle ADI and all associated challenges (Molina et al., 2016; Malmasi et al., 2016; Zampieri et al., 2017; Bouamor et al., 2019; Abdul-Mageed et al., 2020, 2021). Dialectal Arabic (DA) started to appear online in text form with the rise of Web 2.0, which has allowed users to generate online content, such as social media posts, blogs, discussion forums, and emails. However, the ADI problem has become challenging due to a lack of data to represent a wide spectrum of Arabic dialects and their complex taxonomy. In this paper, we examined the possibility of exploiting small datasets in order to build an ADI system with solid performance. We experimented with an integration technique between two classical Machine Learning (ML) approaches. We also automatically created dictionaries based on Pointwise Mutual Information (PMI) and labelled datasets. We exploited the automatically created dictionaries to enrich the feature space when training the models. A semi-supervised learning paradigm, namely, co-training was also designated to exploit the noisy and unlabelled data available hugely online. We also investigated another approach to leverage large unlabelled data, such as building pre-trained word embeddings to be used later in the training process. Two Deep Learning (DL) models, the Gated Recurrent Unit (GRU) and the Bidirectional Long-Short Term Memory (BiLSTM), were investigated along with a co-training method and word embeddings (either pre-trained, or learned during training).

2 Datasets

The dataset used in this paper is the NADI corpus, consisting of 31,000 labelled tweets and encompassing 21 country-level Arabic dialects. The dataset is divided into three parts: 68% for training (21,000 tweets), 16% for the development set (5,000 tweets), and 16% for the test set (5,000 tweets). The NADI shared task organisers also shared with participating teams around 10M unlabelled tweets.

The text pre-processing steps are essential in any Natural Language Processing (NLP) application and may affect positively or negatively the final outputs. The pre-processing pipeline we followed in our study before training our proposed models includes tokenising, the removal of stop words and one-character words, and cleaning the texts. The cleaning step involves the removal of diacritical marks, non-alphanumeric characters including emojis and excessive white spaces.

Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 265–270
Kyiv, Ukraine (Virtual), April 19, 2021.

3 Feature Extraction

In this section, we illustrate the extracted features and the text representation of the input data before feeding them to the proposed models.

3.1 Pointwise Mutual Information (PMI)

The PMI represents a statistical measure of association strength between two events x and y, given their individual and joint distributions (Manning et al., 1999). In our study, the PMI was utilised to assess the significance of specific words to their corresponding Arabic dialects by measuring the correlation between the word and that Arabic dialect. Mathematically, it is the log of the probability of the word being utilised in a certain Arabic dialect divided by the product of the probability of the word and the probability of that Arabic dialect independently.

$$pmi(w, T_{dialect}) = \log \frac{p(w, T_{dialect})}{p(w) * p(T_{dialect})}$$

Where w is a word in the corpus and $T_{dialect}$ represents the tweets from Arabic dialects in the corpus. To compute the required probabilities for the words and Arabic dialects, we relied on the training corpus of the NADI shared task for the country-level DA identification as illustrated in Section 2. The words with the highest PMI scores for a particular dialect have a high probability to appear in tweets of that specific Arabic dialect. On the other hand, the negative PMI score between a word and a certain Arabic dialect indicates the lack of relatedness between this word and that dialect. Clearly, the use of PMI helps to rank the words in the dataset according to their degree of association to each Arabic dialect. We used PMI scores to train some of our proposed models, as follows:

1. Dictionary-based features: We automatically created a dictionary for each Arabic dialect by exploiting the labelled training data and the PMI scores of the dataset's vocabulary in relation to relevant Arabic dialects. That is, only words with positive PMI scores higher than 1.5 were automatically added to the dictionary of each dialect. Consequently, the dictionary of each Arabic dialect includes only dialectal words that correlate most to that dialect. The distribution of the subtask 1.2 dataset, however, is not balanced. In addition, the tweets of each country may be biased to the topics mentioned in this dataset's tweets. Thus, the unbalanced data and limited topics may result in some MSA words with positive PMI values higher than 1.5. To this end, we utilised the MSA corpus provided by the NADI organisers for the subtask 1.1 in order to prevent MSA words from being included in the dictionary of each dialect. The dictionary-based features are used as binary features to indicate whether the word in a tweet exists in the dictionary of an Arabic dialect or not.

2. Tweet's PMI value: Each example in the dataset (i.e., each tweet) has been given a total PMI value, calculated by summing the PMI score of each word in the tweet with each Arabic dialect. That is, for each tweet, 21 PMI values have been computed, which represent the strength of the whole tweet's association with each of the 21 Arabic dialects.

3.2 Word Embeddings

Distributed representations of text at various levels of granularity, including words and sentences, have become the dominant method for vectorising textual data for various NLP tasks (Mikolov et al., 2013b,a, 2018; Pennington et al., 2014; Zahran et al., 2015; Abdul-Mageed et al., 2018). Word embedding representation captures a considerable number of semantic and syntactic word relationships where the words are vectorised by training a neural network on a large corpus. In our study, we vectorised the words of the corpus using two methods of embeddings :

(1) Pre-trained embeddings on 300K Twitter data: We built a word vectors model using 300K tweets from the 10M unlabelled tweets provided by the NADI shared task organisers. The Continuous Bag of Words (CBOW) algorithm was adopted for training the model with the number of dimensions of the embeddings set to 100 (i.e., default value). Other hyper-parameters were set as follows: the window size = 5 and the minimum count of words to consider = 5. We built the word2vec model using the Gensim toolkit in Python (Rehurek and Sojka, 2010). We adopted the same pre-processing steps illustrated in Section 2.

(2) Learned embeddings during the training of the ADI system: The word embeddings layer is learned jointly with a neural network model during training the model on the NADI shared task corpus of country-level dialectal identification.

3.3 Other Features

We used the Term Frequency-Inverse Document Frequency (TFIDF) to represent text when we mod-

elled our ADI systems using classical machine learning approaches. The TFIDF is another numerical measure that represents how important a word is to a specific Arabic dialect. The importance increases proportionally to the number of times the word appears in an Arabic dialect, but is offset by the number of times a word appears in other Arabic dialects (Jones, 1972; Yun-tao et al., 2005; Schütze et al., 2008).

We also utilised word unigram, bigrams, and trigrams as features, as well as the character n-grams where n ranges from 2 to 5.

4 System Description

This section presents the approaches we employed in our experiments to examine various traditional ML and DL models when working on country-level ADI problem, covering 21 countries, with small datasets and various features.

4.1 Classical Machine Learning

Support Vector Machine: We utilised the Support Vector Machine (SVM) to perform the country-level ADI task. The 21 dictionaries built automatically, as explained in Section 3, are used as binary features fed to the SVM in conjunction with other features like Tweet's PMI values and TFIDF vectors, as well as character (2-5)-grams and word (1-3)-grams.

Integrated Machine Learning Technique: It involves integrating two classical ML approaches, where the predictions of one are exploited as additional features to train the second machine learning method. We utilised the Logistic Regression (LR) model as a first classifier whereby its predictions were fed, in conjunction with other features, to the SVM model.

Ensemble Classifier: The final decision of the ensemble classifier employed for country-level ADI relies on hard voting of three individual classifiers: SVM, LR, and a Random Forest (RF).

The features utilised to train each individual classifier in the integrated ML Method and ensemble classifier are: dictionary-based features, Tweet's PMI values, character (2-5)-grams, word (1-3)-grams, as well as TFIDF as a way to represent texts.

Co-Training: We attempted to add additional training data to the limited amount of NADI labelled data (a total of 21,000 training tweets) for country-level DA identification (Blum and Mitchell, 1998). To this end, we utilised an automatic labelling approach, namely co-training to annotate additional data without human intervention. Although the NADI organisers shared 10M unlabelled tweets with the participating teams, we decided to label only 300K tweets.

In our study, the co-training involves the use of three classifiers: LR, SVM, and RF in order to label the same tweet with labels l_1, l_2, and l_3. The tweet is finally labelled l if $l = l_1 = l_2 = l_3$. That is, the tweet will be labelled and included in the final corpus only when the three classifiers agree. We called the 300 annotated tweets that resulted from the co-training technique the $CoTraining_{labelled}$ corpus.

4.2 Deep Learning

GRU: The Gated Recurrent Unit (GRU) was also examined in our study for the country-level DA identification. Our GRU-based model consists of a GRU layer with 50 units. A softmax layer is applied on top for classification (No. of epochs = 30). For word vectors, an embedding layer was added to the beginning of the network with dimensions equal to 100. In addition, we carried out another experiment in which we initialised the words with pre-trained word embeddings on the $CoTraining_{labelled}$ corpus.

BiLSTM: We developed a Bidirectional Long-Short Term Memory (BiLSTM) network, consisting of 50 hidden units, followed by a fully connected layer with 50 hidden units. Lastly, a dense layer with 21 hidden units and softmax activation function was utilised to predict the country of the dialect. We used 300 as input sequence length, 0.1 for dropout rate, and 30 for epochs. We specified *Adam* as the optimisation algorithm and *categorical_crossentropy* as the loss function. Two experiments were conducted: one using the word embeddings learned during the training, and the other using initialised values of words with pre-trained embeddings on the $CoTraining_{labelled}$ corpus. In both cases of word embeddings the dimension was equal to 100.

In order to examine the co-training method and to make experimental evaluations of various methods, the $CoTraining_{labelled}$ corpus was utilised to implement multiple classical ML and DL models, as well as integrated ML techniques with the same settings as explained in previous subsections.

5 Results

We conducted several experiments in which we examined various classical ML and DL models to perform multi-way ADI using small datasets. We also examined various features, as explained in Section 3 and Section 4. We started with character (2-5)-grams and word (1-3)-grams as features to develop an SVM model. We created dictionaries automatically based on Pointwise Mutual Information and labelled datasets. Including the automatically generated dictionaries in the feature space to train the SVM model boosted the model's performance especially in terms of *recall* measure. The use of dictionaries alone as binary features increased the performance of the SVM model, but adding the Tweet's PMI values to the features set yielded the better SVM performance. Furthermore, we experimented with an integration approach between SVM and LR, as well as set of features including dictionary-based features and PMI values. This approach outperformed the SVM model in terms of precision, as expected, but fell behind slightly in terms of *recall* and *F1-score*. We also experimented with the ensemble method by combining the decisions of three individual classifiers: LR, SVM, and RF. The ensemble classifier with dictionary-based features and Tweet's PMI values surpassed the performance of all models in terms of *precision*, a predictable outcome as the ensemble classifier relied on hard voting. Moreover, the BiLSTM and GRU-based models fell short in comparison to classical ML techniques. The small dataset of 21,000 tweets may significantly affect the overall performance of the DL models.

Table 1 shows the performance of various classical ML and DL models evaluated on development and test sets. Table 2 shows the performance of the models when training on the $CoTraining_{labelled}$ corpus that consists of 300K annotated tweets. The increase in training data size by implementing the co-training labelling method resulted in a drop in the overall performance of all models. The classical ML techniques with PMI values and dictionary-based features still outperformed DL models, even on the 300K annotated tweets. This can be attributed to the fact that labelling new tweets was conducted automatically based on other models trained on a small dataset (i.e., NADI training data). We also considered the available unlabelled tweets as resources to build a pre-trained word embeddings model to initialise the word values in DL models, as explained in Section 3. In order to build the pre-trained word embeddings we used $CoTraining_{labelled}$ corpus, the same data employed to train the models to compare both the co-training automatic labelling approach and the use of pre-trained word embeddings model. The results clearly showed that the two techniques performed on par. For example, the BiLSTM-based model achieved *macro-average F1-score* equal to 12.85 when using NADI training data and word embeddings vectors, which are pre-trained on the $CoTraining_{labelled}$ corpus. On the other hand, using $CoTraining_{labelled}$ corpus as the training data of the BiLSATM model resulted in 12.63 *macro-average F1-score*.

Method	Dev Set		Test Set	
	F1	**ACC**	**F1**	**ACC**
SVM,c2-5,w1-3,dict,PMI	18.71	35.68	18.94	35.94
SVM,c2-5,w1-3,dict	18.52	35.54	18.66	35.53
SVM,c2-5,w1-3	17	35.22	-	-
Integrated ML,c2-5,w1-3,dict,PMI	17.87	36.74	18.06	38.47
Integrated ML,c2-5,w1-3,dict	17.57	36.86	18.08	37.21
Ensemble,c2-5,w1-3,dict	17.45	37.26	-	-
Ensemble,c2-5,w1-3,dict,PMI	17.23	37.48	-	-
BiLSTM, WE_{layer}	13.2	26.16	-	-
GRU, WE_{layer}	12.85	29.14	-	-

Table 1: The performance of classical ML and DL models trained on NADI subtask 1.2 corpus and evaluated on development and test sets. "c2-5" indicates character n-grams where n=2-5, "w1-3" indicates word n-grams where n=1-3, "dict" indicates dictionary-based features, "PMI" is Tweet's PMI value, and "WE_{layer}" indicates word embeddings learned during the training.

6 Discussion and Analysis

We analysed the outputs of various models on the development dataset. We noticed that many mislabelled tweets are short in length and do not contain

Method	F1	P	R	ACC
Integrated ML,c2-5,w1-3,dict	14.45	17.12	15.88	24.6
SVM,c2-5,w1-3,dict,PMI	14.2	17.09	15.4	24.14
SVM,c2-5,w1-3	14.04	16.45	15.63	23.76
BiLSTM, $WE_{pretrained}$	12.85	13.89	14.17	35.8
BiLSTM, WE_{layer}	12.63	14.24	13.13	22.96
GRU, WE_{layer}	11.51	12.3	13.71	26.68
GRU, $WE_{pretrained}$	11.5	12.73	13.35	36.3

Table 2: The performance of classical ML and DL models trained on $CoTraining_{labelled}$ corpus and evaluated on the development set. "c2-5" indicates character n-grams where n=2-5, "w1-3" indicates word n-grams where n=1-3, "dict" indicates dictionary-based features, "PMI" is Tweet's PMI value, "WE_{layer}" indicates word embeddings learned during the training, and "$WE_{pretrained}$" indicates the use of pre-trained word embeddings.

any distinctive dialectal Arabic words (i.e, words primarily used in one Arabic dialect and can be used to distinguish that dialect from others). For example, the tweet (حتى تكمل اكيد) is labelled as "Iraq" in the corpus but predicted as "Saudi Arabic" by our system of integrated ML technique. In addition, we noticed that a number of tweets contain obvious distinctive dialectal words, however, the model failed to identify the correct dialect. This can be partially attributed to the fact that the training data for NADI subtask 1.2 is small and for 6 out of 21 dialects, the number of training samples is less than 250. On the other hand, although the automatically created dictionaries led to better performance for the classical ML techniques, these dictionaries were collected based on a small dataset (labelled data of NADI subtask 1.2 training corpus). The unbalanced distribution of samples per country in the corpus led to obtaining words with high PMI values for some Arabic dialects and with low PMI values for others, although the words are shared between these dialects. For example, the word (مافيش) is used in Egypt and Yemen, so the PMI values between the word and the two Arabic dialects should not be high , since it is shared between the two dialects. However, the word's PMI value was high with the Yemeni dialect (1.167) and relatively low with Egyptian (0.322), because the number of Egypt samples in the corpus is 4283 while Yemen has only 429 samples. We expect that a large corpus with relatively even numbers of Arabic dialect samples will result in dictionaries with a considerable number of entries and better PMI values.

Normalisation is one of the preprocessing steps when building NLP applications. For dialectal Arabic normalisation is a vital step that should be taken into consideration according to the Arabic NLP application's needs. In order to reduce the data sparsity and increase its accuracy, one should take normalisation of some Arabic letters into account, especially when building pre-trained word embeddings or learning them during the training process.

7 Conclusion

In this paper, we investigated the possibility of exploiting small datasets in order to build a strongly performing ADI system, by integrating various machine learning approaches. We also created dictionaries automatically based on Pointwise Mutual Information and labelled datasets, which enriched the feature space when training models. A semi-supervised learning paradigm, namely co-training was also examined as a way to leverage the large and noisy unlabelled dialectal Arabic datasets. Another approach examined in this study to exploit large unlabelled datasets was building pre-trained word embeddings model to be used later in the training process. Both approaches to leverage large and noisy unlabelled data showed similar results and performed on par. Our winning model was the Support Vector Machine with dictionary-based features and PMI values, achieving 18.94% macros-average F1-score on the NADI subtask 1.2 test set. In terms of precision, the integration method that merged two classical machine learning algorithms surpassed all models with precision equal to 22.70%. We also concluded that pre-processing text is a vital step before building any dialectal Arabic models and needs to be further investigated to understand its side effects and benefits.

References

Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3653–3659.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The Second Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop (WANLP 2021)*.

Maha J Althobaiti. 2020a. Automatic Arabic Dialect Identification Systems for Written Texts: A Survey. *International Journal of Computational Linguistics (IJCL)*, 11(3):61–89.

Maha J Althobaiti. 2020b. Automatic Arabic Dialect Identification Systems for Written Texts: A Survey. *arXiv preprint arXiv:2009.12622*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

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 Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 1–14.

Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12.

Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 52–55.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the Second Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*, pages 46–50.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Zhang Yun-tao, Gong Ling, and Wang Yong-cheng. 2005. An improved TF-IDF approach for text classification. *Journal of Zhejiang University-Science A*, 6(1):49–55.

Mohamed A Zahran, Ahmed Magooda, Ashraf Y Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015. Word representations in vector space and their applications for arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 430–443. Springer.

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*.