Proceedings of the OSACT 2022 Workshop @LREC2022, pages 41–50
Marseille, 20 June 2022

Pre-trained Models or Feature Engineering: The Case of Dialectal Arabic

Chatrine Qwaider (Kathrein Abu kwaik)[*] , Stergios Chatzikyriakidis[◇*], Simon Dobnik[*]
[*]Centre for Linguistic Theory and Studies in Probability, FLoV, University of Gothenburg
{kathrein.abu.kwaik, simon.dobnik}@gu.se
[◇]Department of Philology, University of Crete
{Stergios.Chatzikyriakidis}@uoc.gr

Abstract

The usage of social media platforms has resulted in the proliferation of work on Arabic Natural Language Processing (ANLP), including the development of resources. There is also an increased interest in processing Arabic dialects and a number of models and algorithms have been utilised for the purpose of Dialectal Arabic Natural Language Processing (DANLP). In this paper, we conduct a comparison study between some of the most well-known and most commonly used methods in NLP in order to test their performance on different corpora and two NLP tasks: Dialect Identification and Sentiment Analysis. In particular, we compare three general classes of models: a) traditional Machine Learning models with features, b) classic Deep Learning architectures (LSTMs) with pre-trained word embeddings and lastly c) different Bidirectional Encoder Representations from Transformers (BERT) models such as (Multilingual-BERT, Ara-BERT, and Twitter-Arabic-BERT). The results of the comparison show that using feature-based classification can still compete with BERT models in these dialectal Arabic contexts. The use of transformer models have the ability to outperform traditional Machine Learning approaches, depending on the type of text they have been trained on, in contrast to classic Deep Learning models like LSTMs which do not perform well on the tasks.

Keywords: Dialect Identification, Sentiment Analysis, Machine learning, Deep Learning, Feature engineering, Language modelling

1. Introduction

The last decade has not only seen the emergence and development of social media platforms, but also, and relating to the latter, an increased interest in the automatic processing of Arabic dialects. A number of researchers have investigated several tasks related to Dialectal Arabic (DA) Natural Language Processing (NLP) that range from purely theoretical issues of syntax and morphology (Chiang et al., 2006; Habash et al., 2005) to more applied tasks like language generation and machine translation (Zbib et al., 2012; Meftouh et al., 2015; Diab and Habash, 2014).

Regardless of the increase in the interest of processing Arabic dialects, this research is still in its developing stage and the lack of significant and valuable resources is well-known. Currently, a lot of the NLP research handles the problem of Dialectal Arabic by introducing and building different kind of resources, e.g. lexicons, corpora, tree-banks and others that are usually focused on the specific task they attempt to address (Guellil et al., 2019). Dialectal Arabic resources are still suffering from the lack of available data that would enable a full investigation of the newly introduced Deep Learning (DL) networks on it.

Furthermore, the research that supports DA differs in terms of the tasks and the datasets used, a fact that leads to different results that are hard to compare. Some researchers and developers still support the use of traditional ML techniques in Arabic NLP given the limited size of available corpora (Abdul-Mageed et al., 2014), while others try to overcome and fine-tune complex DL networks (Heikal et al., 2018). In the case of corpora that are of limited size, feature-based ML approaches still give better results than DNNs for DA NLP tasks(Qwaider et al., 2019). In this paper, we investigate the performance of different approaches on DA on two NLP tasks: Dialect Identification (DI) and Sentiment Analysis (SA). We explore various datasets that have different sizes, balanced and imbalanced, hand-crafted and user-generated. In addition, we employ several features such as n-gram language models, pre-trained word embeddings, and pre-trained language models (Devlin et al., 2018). For classification tasks, we try traditional ML algorithms like Support Vector Machine (SVM), fully connected dense layers and Long Short Term Memory (LSTM) networks. For Sentiment Analysis, we achieve the state-of-the-art on the corpora used. In addition, our approach is one of the few that applies BERT for the Dialect Identification task.

The paper is organised as follows: Section 2 discusses recent related work in DI and SA, while Section 3 introduces the datasets used throughout our experiments. Section 4 presents the experiments, settings and, results. The section is in Section 5, while the conclusions can be found in Section 6.

2. Related work

In this paper we focus on two NLP tasks: Dialect Identification and Sentiment Analysis. Three main approaches are presented: (i) traditional ML with feature engineering, (ii) LSTM DL architectures and (iii) pre-trained language models.

The vast majority of research as regards Dialect Identification, uses traditional ML with feature engineering

(Tachicart et al., 2017; Obeid et al., 2019; Zaidan and Callison-Burch, 2014). Recently, after the introduction of the MADAR corpus (Bouamor et al., 2018), which covers 25 Arabic dialects, a good amount of research followed. Salameh et. al; (2019) present a fine-grained Dialect Identification model, where a character-gram language model with Multinomial Naive Bayes (MNB) classifier is used to identify the label of 25 dialects is used. The top five ranked systems at the MADAR shared task, focus on traditional character feature classification (Abu Kwaik and Saad, 2019; Meftouh et al., 2019; Ragab et al., 2019; Mishra and Mujadia, 2019). All these papers conclude that neural methods did not do as well as traditional ML approaches which is likely the result of limited training data. However, despite this inability of DL architectures to outperform traditional ML models, a number of researchers have turned towards straightforward DL architectures. For example, (Ali, 2018) proposes a deep learning CNN network based on character feature extraction to distinguish among MSA and dialects. De Francony et al. (2019), compare two approaches for Arabic fine-grained Dialect Identification, one using an RNN (BLSTM, BGRU) with hierarchical classification and and another using a voting classifier approach based on NB and Random Forest. In the same vein, (Fares et al., 2019) try different combinations of deep learning networks with different kinds of features on the MADAR corpus. These last two works both conclude, in line with the results from the MADAR task systems, that traditional ML algorithms outperform deep learning networks arguing that this might be because of the small size of the used corpus.

Recently, pre-trained language models such as BERT have been used for Dialect Identification. In (Zhang and Abdul-Mageed, 2019; Talafha et al., 2020; Beltagy et al., 2020), different Dialect Identification models based on BERT are introduced for the MADAR (Bouamor et al., 2019) and the NADI shared tasks[1].

Sentiment Analysis is a supervised classification task where a proposed model should be able to classify a sentence into two or more sentiment classes. The dominant approach for Arabic Sentiment Analysis in the last couple of years, as in the case of Dialect Identification, has been the feature-based and language modelling approach using ML classification algorithms like SVM, Multinomial Naive Bayes Classifier and others (Mountassir et al., 2012; Aly and Atiya, 2013; Omar et al., 2013; Elawady et al., 2014; Al-Saqqa et al., 2018). Some works use linguistics features such as the stems, lemmas, or part-of-speech, in addition to the Arabic variety (MSA, dialect), while others use more specific features depending on the kind of the dataset, e.g. userID (person, organisation) and the gender of the user found in datasets that use Twitter data (Abdul-Mageed et al., 2014; Shoukry and Rafea, 2012). However, most research uses language models by extracting words and character n-grams and investigating different ML classifiers (Duwairi et al., 2014).

Recently, as for Dialect Identification, researchers and developers started using deep learning networks for Sentiment Analysis with word embeddings and pre-trained language models. A CNN feature extractor and transformation network was proposed in (Soumeur et al., 2018) to determine the sentiment of Algerian users' comments on various Facebook brand pages of companies in Algeria, while (Baly et al., 2017) present an LSTM network with pre-trained word embeddings to build a 5-scale Sentiment Analysis model for 4 Arabic dialects. A combination of word and document embeddings in addition to a set of semantic features were used in (Abdullah et al., 2018) for Arabic tweets. The features are applied into a CNN-LSTM network followed by a fully connected layer. Heikal et al., (2018) propose an ensemble DL model that combines an LSTM with a CNN to predict the sentiment class of Arabic tweets exploiting the Arabic Sentiment Tweets Dataset (ASTD). Deep LSTM-CNN networks were used in (Mohammed and Kora, 2019) and a new 40K-tweets dataset collected from Twitter focusing on Egyptian dialects. Similarly, (Kwaik et al., 2019) propose a DL model that uses AraVec word embeddings with two Bi-LSTMs followed by 15 parallel CNN layers.

With the advent of pre-trained language models, a considerable amount of research concentrated on building and training their dialectal models by applying Arabic BERT as a first layer of the model instead of using a word-embeddings layer. In (Antoun et al., 2020) a Transformer-based Model for Arabic Language Understanding called AraBERT is proposed and applied on different Dialectal Arabic NLP tasks such as Sentiment Analysis and Question Answering. Some projects have built their own dialectal BERTs to be used in their specific models such as DziriBERT for Algerian dialects (Abdaoui et al., 2021) and ARabiziBERT where Arabizi is a written form of spoken Arabic that relies on Latin characters and digits (Baert et al., 2020).

Despite a large number of work on DA Dialect Identification and Sentiment Analysis, it is still an open question whether using old-fashion ML algorithms with feature engineering is better than using more sophisticated deep learning networks and pre-trained language models. This is because the results reported in the literature are based on models that are trained on datasets that differ in terms of size, the dialects covered, classification methods, or even the quality of the dataset. In this paper, we make a comparison using the same corpora on which we apply several models.

3. Datasets

We will use a number of well-known corpora. For the task of Dialect Identification we use the following:

- PADIC (Meftouh et al., 2015): a Parallel Arabic Dialect Corpus (PADIC) that was collected from Algerian telephone conversations, transcribed and

[1] https://sites.google.com/view/nadi-shared-task

then translated to other dialects. It is composed of 6.400 sentences for each dialect. The corpus contains five dialects where two of them present Algerian dialects (Algeria, Annaba), one from Tunisia and two dialects from Levantine (Palestine, Syria), in addition to MSA.

- SHAMI (Kwaik et al., 2018): a Levantine dialect corpus, includes 66.251 documents which were collected from different domains such as sports, social life, cooking, and others and it covers the four Levantine dialects. The corpus is unbalanced in term of number of documents per dialect with 10.830, 37.760, 10.643, 7.018 for Lebanese, Syrian, Palestinian and Jordanian respectively.
- MADAR-6 (Bouamor et al., 2018): a parallel corpus in the travel domain that covers, in addition to MSA, five different Arabic dialects from five Arabic cities: Beirut (BEI), Cairo (CAI), Doha (DOH), Rabat (RAB), Tunisia (TUN), therefore it is called MADAR-6. The corpus is composed of 10.000 documents for each dialect.

We focus on binary classification where the document is classified as either positive or negative. The three corpora we use are:

- ATSAD (Kwaik et al., 2020): an Arabic Tweets Sentiment Analysis Dataset (multi-dialects). The corpus has been collected from Twitter during April 2019 and employs emojis as seeds for extraction of candidate instances. It is a balanced binary corpus which was partly annotated by human experts and then self training techniques were applied to annotate the rest of tweets. The corpus contains 18.173 and 18.695 negative tweets and positive tweets respectively.
- 40-K tweets (Mohammed and Kora, 2019): an Egyptian binary balanced corpus where all tweets were pre-processed and cleaned manually by two experts. The total size is 40,000 tweets, where 20.002 tweets are negative and 19.998 are positive.
- ASTD (Nabil et al., 2015): an Arabic Sentiment Tweets Dataset focusing on the Egyptian dialect. The corpus is composed of 10k tweets classified for objective and subjective sentiment. It is unbalanced dataset since there are 1.681 negative documents and 818 positive ones.

4. Experiments

In this section we describe our experiments and the models used for both Dialect Identification and Sentiment Analysis on dialectal Arabic. For both tasks, we make use of three corpora as shown in the previous section. We split the datasets into 90% for training set and 10% for testing. The 90% training part is further split into 80% for training and 20% for validation. Tables 1 and 2 show the total size of the corpora alongside the number of sentences for every set: training, validation, and testing.

We investigate the performance and the differences between the models. We performed a number of experiments where we focus on some common popular architectures. First of all, we apply BERT as a pre-trained language model followed by a classification layer. Then we compare it with a model with pre-trained word embeddings (AraVec) (Soliman et al., 2017) and an LSTM network. We also investigate the performance of feature extraction language model on both traditional ML algorithms like SVM and on a fully connected classification layer. Figure1 shows a diagram summarising all the experiments.

In order to evaluate the performance of the models, we use accuracy together with the following two measures:

- Mathews correlation coefficient (MCC): A measure used in ML to measure the quality of classification model (Matthews, 1975). It is a balanced measure which could also be used for imbalanced classification problem (Boughorbel et al., 2017). The MCC has a value between -1 (total disagreement between prediction and observation) to +1 (perfect prediction), and 0 value indicate random prediction. MCC is calculated from the confusion matrix according to Equation 1

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

- F-score: also a well-known measure for classification success in ML. The F-score is the harmonic mean between the precision and the recall (Derczynski, 2016). Through all the experiments we chose the F-score to optimize on the validation set, as some of datasets are not balanced, so accuracy could not be a good choice for optimization.

4.1. BERT for Dialectal Arabic

The main component of BERT or Bidirectional Encoder Representations from Transformer (Devlin et al., 2018) is a Transformer which is an encoder-decoder attention mechanism that has been build to learn the contextual relations between sequence of words in any text and generate a language model. It takes a sequence of words (sub-words) as an input layer. These tokens are embedded into vectors and then go through the transformer encoder. The output of BERT is a sequence of vectors, where each vector presents an input token. To apply fine-tuning on BERT for any classification task or language generation task, a fully connected classification layer with a soft-max activation function is built on top of the output vectors.

As we work on Dialectal Arabic, a natural thing to do is to use the Arabic versions of BERT. On top of the BERT model we add a classification layer for our two tasks which are trained separately. We use the following BERT models:

Dataset	# Dialects	Total size	Train_set	Val_set	Test_set
PADIC	6	33,502	25,560	6,391	3551
Shami	4	66,251	47,699	11,925	6,626
MADAR-6	6	60,000	43,200	10,800	6,000

Table 1: Corpora statistics for the Dialect Identification task

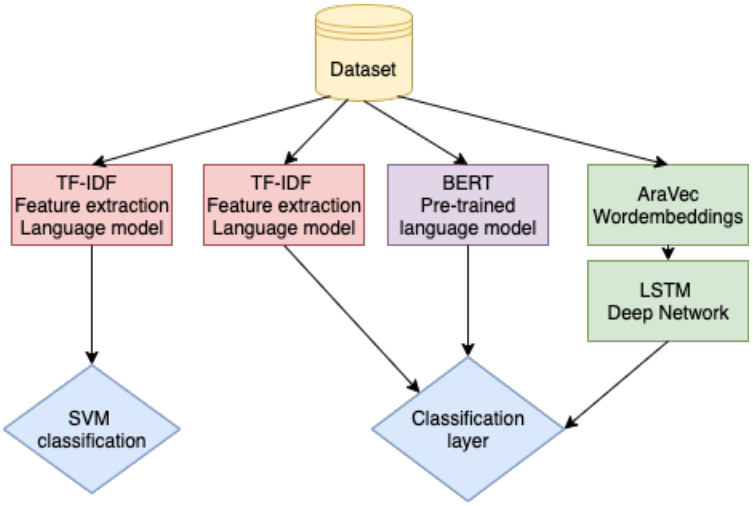

Figure 1: Fours different models used in the classification tasks in the experiments

Dataset	Total	Train	Val	Test
ATSAD	22,542	16,229	4,058	2,255
40-K	40,000	28,800	7,200	4,000
ASTD	2499	1,799	450	250

Table 2: Corpora statistics for the Sentiment Analysis task

1. Multilingual-BERT[2]: This is the multi-lingual version of BERT, which contains the top 100 languages with the largest Wikipedia content, including Modern Standard Arabic.

2. Arabic-BERT (Safaya et al., 2020): consists of 4 models of different sizes (Large, Base, Medium and Mini). We use the base model for the experiments. Arabic-BERT has been built with 8,2B words from the OSCAR data (Suárez et al., 2020) and the recent data dump from Wikipedia.

3. AraBERT-Twitter-base (Antoun et al., 2020): AraBERT is an Arabic pre-trained language model based on Google's BERT architecture. It also uses the same BERT-base configuration. There are two versions of AraBERT v1 and v2 where they differ in term of segmentation techniques. AraBERT-Twitter-base is the dialectal version of AraBERTv2. It contains 60M Multi-Dialect Tweets in addition to 200M from the AraBERTv2-base.

[2]https://github.com/google-research/bert/blob/master/multilingual.md

The same parameters are used through all the experiments in order to get a reasonable comparison. We use the Adam optimiser where the learning rate is 5e-5 and epsilon is equal to 1e-8 through all the BERT models. We set the batch size to be 32 for multi-lingual BERT and 16 for both Arabic BERT and Twitter-AraBERT models. The preferred number of epochs for fine tuning Multi-BERT is between 2 to 4 (Devlin et al., 2018). In our case 4 epochs was the best choice for Sentiment Analysis, while for Dialect Identification the best number of epochs was 10. For Arabic BERT and Twitter-AraBERT, the number of epochs is between 8 to 10 and we employ early stopping and save the best performed epoch. We also explore max sequence lengths for both tasks and decide on 77 for Sentiment Analysis and 130 for Dialect Identification using Multi-lingual BERT, and 280 and 256 for ArabicBERT and Twitter-AraBERT respectively.

We build the first model by employing Multi-Lingual BERT as a basic layer, and then have a softmax fully connected layer for classification purposes. Table 3 and Table 4 present the output results for the Accuracy, MCC and F-score for the two tasks. For Dialect Identification the accuracy ranges between 0.72 to 0.89 with 10 epochs and has very short training time compared to an end-to-end neural network. In case of Sentiment Analysis we get an accuracy between 0.8 to 0.83 where the model outperforms the state of the art result using

deep learning on the ASTD corpus (Heikal et al., 2018; Kwaik et al., 2019).

Dataset	Accuracy	MCC	F-score
PADIC	0.72	0.67	0.72
Shami	0.88	0.81	0.83
MADAR-6	0.89	0.87	0.89

Table 3: Results of applying Multilingual-BERT on Dialect Identification task

Data_set	Accuracy	MCC	F-score
ATSAD	0.80	0.6	0.80
40-k Tweets	0.83	0.66	0.83
ASTD	0.81	0.51	0.75

Table 4: Results of applying Multilingual-BERT on Sentiment Analysis task

The Multi-Lingual model was not only built for the purpose of Arabic-NLP. For comparison we implement the second model using Arabic-Bert. We used the basic version of Arabic-BERT and then the same soft-max classification layer. The three test measurements (Accuracy, MCC, F-scores) for Dialect identification and Sentiment Analysis are presented in Tables 5 and 6 respectively. The accuracy for DI models range from 0.71 to 0.80 which is less than the previous Multilingual BERT model. This is may be because the later model was trained on different languages so it is easier to fine-tune it to identify or classify languages or dialects. On the other hand, on Sentiment Analysis the models performed better than those using Multilingual-BERT where the accuracy is in the range of 0.83 to 0.90.

Both Multilingual BERT and Arabic-BERT have been trained on MSA data that was collected mainly from news websites and Wikipedia documents. We conduct a third experiment with BERT which was trained on dialectal data, the Twitter-AraBERT. Table 7 shows the test accuracy on the Dialect Identification task. The model is the best among those described previously.

Data_set	Accuracy	MCC	F-score
PADIC	0.71	0.66	0.72
Shami	0.87	0.78	0.81
MADAR-6	0.80	0.76	0.80

Table 5: Results of applying Arabic-BERT on Dialect Identification task

Data_set	Accuracy	MCC	F-score
ATSAD	0.93	0.87	0.93
40-k Tweets	0.83	0.66	0.83
ASTD	0.84	0.63	0.81

Table 6: Results of applying Arabic-BERT on Sentiment Analysis task

The accuracy is now in the range of 0.77 and 0.91. Table 8 shows the accuracy of Sentiment Analysis models which ranges from 0.88 to 0.97. The model outperforms the state-of-the-art on the 40K tweets dataset (Mohammed and Kora, 2019). Here, the authors achieve an average accuracy of 0.81 using LSTM models. In addition, it outperforms the state-of-the-art on the ASTD corpus (Heikal et al., 2018). Among the three BERT models, Twitter-AraBERT is the best performing model when the data used for training is mostly dialectal.

Data_set	Accuracy	MCC	F-score
PADIC	0.77	0.73	0.77
Shami	0.91	0.86	0.87
MADAR-6	0.91	0.90	0.91

Table 7: Results of applying Twitter-AraBERT on Dialect Identification task

Data_set	Accuracy	MCC	F-score
ATSAD	0.97	0.94	0.97
40-k Tweets	0.91	0.82	0.91
ASTD	0.88	0.74	0.87

Table 8: Results of applying Twitter-AraBERT on Sentiment Analysis task

4.2. LSTM Baseline

We build a simple LSTM baseline and apply it to the corpora for the two tasks. We employ the AraVec (Twitter-CBOW 300) which are pre-trained Arabic word embeddings as a first layer (Soliman et al., 2017), followed by an LSTM layer with 70 nodes and a dropout of 0.25%. This is followed by a fully connected dense layer with 30 nodes. The last layer is also a fully connected dense layer where the output depends on the number of classes in each task. For Dialect Identification, there are 6, 6 and 4 output classes for PADIC, MADAR-6 and SHAMI respectively. For the Sentiment Analysis task, we use the binary classification task. Table 9 shows the LSTM baseline settings.

Max_length	130 (DI), 77 (SA)
Optimiser	Adam (DI), RMSprop(SA)
Word_embeddings	AraVec (Twitter-CBOW 300)
LSTM_nodes	70
Drop_out	0.25
Dense_nodes	30
Activation_function	Sigmoid
Loss	Categorial_crossentrapy (DI), Binary_crosentrapy (SA)
Batch_size	32
Epochs	up to 100, Early_stopping

Table 9: LSTM baseline network settings

We use the loss with a minimum value to monitor the model and to save the best performance weights. Tables 10 and 11 show the results of applying the baseline into the corpora in concern. It is clear that a baseline LSTM with Arabic pre-trained word embeddings is not able to perform well with dialectal Arabic NLP tasks. The accuracy does not exceed 0.6 in any corpus. Moreover, the MCC shows zero values through all the corpora which means that the classifier is not able to correctly classify the documents and it is no better than random prediction. For Shami corpus the accuracy is high (comparing to other datasets) while the F-score is equally low 0.18, which suggests that Shami is more imbalanced and the model is not doing well on recall on minority classes.

Data_set	Accuracy	MCC	F-score
PADIC	0.17	0	0.14
Shami	0.57	0.004	0.18
MADAR-6	0.17	0	0.29

Table 10: Results of applying LSTM baseline on Dialect Identification task

Data_set	Accuracy	MCC	F-score
ATSAD	0.52	0	0.34
40-k Tweets	0.49	0	0.33
ASTD	0.69	0	0.41

Table 11: Result of applying LSTM baseline on Sentiment Analysis task

4.3. Feature-Based Classification for Dialectal Arabic

In addition to BERT and LSTM experiments we also investigate the performance of traditional Machine Learning algorithms on Dialectal Arabic. An SVM machine learning model (linear SVC) was built and proposed in (Qwaider et al., 2019) for dialectal Arabic Sentiment Analysis. We employ the same approach for both tasks. The models apply various n-gram features as follows:

- Word-gram features with uni-gram, bi-grams and tri-grams, the transformation weight is 0.8.
- Character-gram features with word boundary consideration from bi-grams to 5-grams and the transformation weight of 0.5
- Character-gram features without word boundary consideration from bi-grams to 5-grams and the transformation weight of 0.4.

For the SVM ((linear SVC) classifier, we set a linear kernel and use the default squared_hinge loss function, we set tolerance to be 1e-5, other parameters are kept as default. The results after training and testing the model are presented in Table 12 and 13.

Data_set	Accuracy	MCC	F-score
PADIC	0.72	0.66	0.72
Shami	0.90	0.84	0.86
MADAR-6	0.89	0.87	0.89

Table 12: Results of applying the feature-based model (SVM) on the Dialect Identification task

Data_set	Accuracy	MCC	F-score
ATSAD	0.96	0.92	0.96
40-k Tweets	0.84	0.67	0.84
ASTD	0.80	0.45	0.71

Table 13: Results of applying the feature-based model (SVM) on the Sentiment Analysis task

We further investigate the effect of feature based approaches by placing a fully connected classification layer on the top of the language model rather than using a traditional machine learning algorithm such as SVM or NB. The model seems like BERT, but instead of the pre-trained language model layers we use the feature extraction language model discussed before, followed by a classification layer. Table 14 and 15 show the results of this experiment. Table 16 and 17 report the results for all the aforementioned experiments.

Data_set	Accuracy	MCC	F-score
PADIC	0.73	0.68	0.74
Shami	0.57	0	0.50
MADAR-6	0.89	0.87	0.89

Table 14: Results of the feature-based model with fully connected classification layers on the Dialect Identification task

Data_set	Accuracy	MCC	F-score
ATSAD	0.96	0.91	0.95
40-k Tweets	0.82	0.63	0.82
ASTD	0.77	0.49	0.73

Table 15: Results of the feature-based model with fully connected classification layers on the Sentiment Analysis task

5. Discussion

The LSTM model is the worst performing model among all the models with a huge evaluation gap between that and the other models. The low performance of the LSTM network might be due to the usage of the AraVec pre-trained word embeddings. The percentage of OOV words is high (from 30% to 70%). We try to overcome this problem by replacing the embedding vector for the missing word with the embedding vector for the least lexical-distance word. To compute the lexical distance, we apply the Levenshtein distance algorithm and set the distance to at most two characters.

	PADIC			SHAMI			MADAR-6		
	Acc	MCC	F	Acc	MCC	F	Acc	MCC	F
Multilingual-BERT	72	67	72	88	81	83	89	87	89
Arabic-BERT	71	66	72	87	78	81	80	76	80
Twitter-BERT	**77**	**73**	**77**	**91**	**86**	**86**	**91**	**90**	**91**
LSTM	17	0	14	57	0.4	18	17	0	29
TFIDF + SVM	72	66	72	90	84	86	89	87	89
TFIDF+ Dense	73	68	74	57	0	50	89	87	89

Table 16: Performance measurements for all the experiments on Dialect Identification.

	ATSAD			40K tweets			ASTD		
	Acc	MCC	F	Acc	MCC	F	Acc	MCC	F
Multilingual-BERT	80	60	80	83	66	83	81	51	75
Arabic-BERT	93	87	93	83	66	83	84	63	81
Twitter-BERT	**97**	**94**	**97**	**91**	**82**	**91**	**88**	**74**	**87**
LSTM	52	0	34	49	0	33	69	0	41
TFIDF + SVM	96	92	96	84	67	84	80	45	71
TFIDF+ Dense	96	91	95	82	63	82	77	49	73

Table 17: Performance measurements for all the experiments on Sentiment Analysis

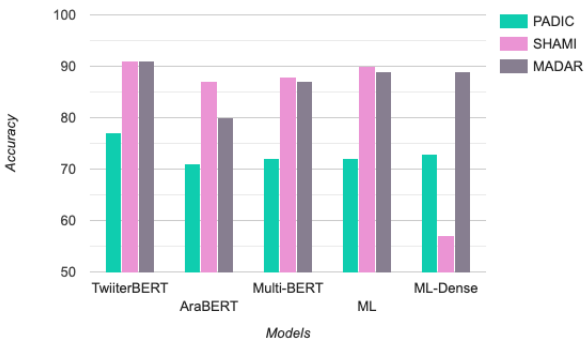

Figure 2: Accuracy of different Dialect Identification models

This makes the LSTM network not perform well even when the word embeddings layer is set to be trainable. The network is also biased towards the majority class. This is very clear in the case of SHAMI, which is the most unbalanced dataset of all.

As we see from the experiments, feature-based classification methods can compete with the pre-trained language models followed by a fully connected layer and sometimes even outperform them. Figure 2 plots the accuracy for the Dialect Identification models as well as the used corpora. The LSTM is not shown, as it is the worst of all and the MCC was 0. Although the results are close in some cases, the Twitter-AraBERT outperforms all the models on all corpora. The Twitter-AraBERT model and the ML models are close to each other in terms of accuracy especially for SHAMI, a non-parallel and unbalanced corpus. It is clear that the size of the corpus has an effect on the performance of the DI task. For example, ML with feature based and svm algorithm is doing better on SHAMI and MADAR than PADIC. However, for a corpus of reasonable size, even with unbalanced data like SHAMI, ML algorithms (SVM) have the ability to compete with pre-trained language models. On well structured and human annotated corpora like PADIC and MADAR both feature-based approaches do nearly the same regardless of whether they are using an SVM or a classification layer. Both corpora have handcrafted examples that increase the power of n-gram language models.

Figure 3 plots the accuracy for the Sentiment Analysis task. Also here Twitter-AraBERT is the best over all the corpora. Sentiment Analysis is a task that does not depend on the structural properties of language as much but rather on the context where emotions are expressed. On the ATSAD corpus where emojis were used as weak labels for annotation Twitter-AraBERT performs very well. Twitter-AraBERT is also able to deal efficiently with the problem of imbalanced datasets like ATSD which is of small size. When it comes to the composition of dialects in the datasets, the 40k-tweets dataset as well as ASTD include Egyptian data. In this case, Twitter-BERT performs better than ML methods. In contrast, in a multi-dialect corpus like ATSAD, feature-based approaches are also a good choice, achieving results very close to those obtained with Twitter-BERT.

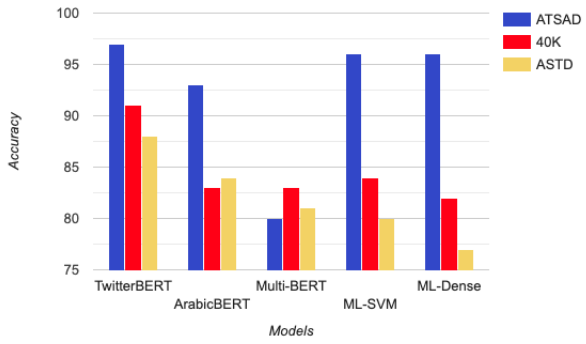

Figure 3: Accuracy of different Sentiment Analysis models

In general, applying pre-trained language models on dialectal Arabic NLP tasks leads to reasonable results. Many factors play a role on the decision of which model to choose for an NLP task: The size of the dataset, the sources and the quality of the data, the data balance, whether the corpus contains MSA or multi-dialectal Arabic data, as well as the number of classes. For under-resourced languages we show that traditional ML approaches perform well and that they are still a highly competitive choice over more complicated and time and resource intensive deep neural models.

6. Conclusion

In this study, we discuss the issue of choosing the best methods for Dialectal Arabic NLP tasks, taking into consideration the differences among the resources. We implemented various approaches from traditional ML to the most recent approaches using BERT, and using different corpora. We wanted to measure the performance of pre-trained models compared to feature-based ML methods. Firstly, we proposed the use of a pre-trained language model like BERT into Dialectal Arabic. Two DA-NLP tasks were used in this study (Dialect Identification and Sentiment Analysis) on six different corpora (3 for each task). Fine-tuning BERT for DA can produce acceptable results on all corpora. Using BERT that supports Arabic saves effort and time to build deep learning models for dialectal Arabic from scratch.

The second part of the study investigates other classification approaches and compares them to the BERT models. We build an LSTM baseline with the support of the pre-trained AraVec word embeddings which does not perform well. The usage of AraVec with a large OOV dialectal words does not facilitate the model in being retrained and fine-tuned for DA. We also built feature-based models either using the SVM or using a fully connected neural network layer. The usage of a tailor-made feature extractor can compete end-to-end feature training in BERT. In summary, after investigating feature-based and feature-pre-trained machine learning approaches, we can say that training DL models such as LSTMs directly from data is not a good solution for the specific tasks and datasets for DA. BERT-pre-trained models appear to be a good solution for dialectal Arabic tasks but feature-pre-training is nearly matched by traditional feature-based models. However, not all BERT-pre-trained models perform equally well. There is a preference for the model that was trained on social media which contains dialectal linguistic variation. However, the use of pre-trained models does not necessarily mean getting better results all the time. In some experiments the use of the SVM algorithm with feature-based classification does surprisingly well and produces very competitive results. In the future, we intend to consider performing error analysis to have a deep look into the proposed approaches, especially on the LSTM approach, since the performance was surprisingly too low. In addition, make more effort to implement an effective LSTM model that can compete with the aforementioned models. Moreover, we want to compare more between BERT models and Feature-based engineering models, for example, in terms of running or inference time., so researchers can decide on the chosen model based on their preferences criteria.

7. Acknowledgements

The research reported in this paper was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

8. Bibliographical References

Abdaoui, A., Berrimi, M., Oussalah, M., and Moussaoui, A. (2021). Dziribert: a pre-trained language model for the algerian dialect. *arXiv preprint arXiv:2109.12346*.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Computer Speech & Language*, 28(1):20–37.

Abdullah, M., Hadzikadicy, M., and Shaikhz, S. (2018). SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 835–840. IEEE.

Abu Kwaik, K. and Saad, M. K. (2019). ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification. *ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification*, (Proceedings of the Fourth Arabic Natural Language Processing Workshop).

Al-Saqqa, S., Obeid, N., and Awajan, A. (2018). Sentiment Analysis for Arabic Text using Ensem-

ble Learning. In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. IEEE.

Ali, M. (2018). Character level convolutional neural network for Arabic dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 122–127.

Aly, M. and Atiya, A. (2013). Labr: A large scale arabic book reviews dataset. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 494–498.

Antoun, W., Baly, F., and Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Baert, G., Gahbiche, S., Gadek, G., and Pauchet, A. (2020). Arabizi language models for sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 592–603, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Baly, R., El-Khoury, G., Moukalled, R., Aoun, R., Hajj, H., Shaban, K. B., and El-Hajj, W. (2017). Comparative evaluation of sentiment analysis methods across arabic dialects. *Procedia Computer Science*, 117:266–273.

Beltagy, A., Wael, A., and ElSherief, O. (2020). Arabic dialect identification using bert-based domain adaptation. *arXiv preprint arXiv:2011.06977*.

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., Obeid, O., Khalifa, S., Eryani, F., Erdmann, A., et al. (2018). The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

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

Boughorbel, S., Jarray, F., and El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one*, 12(6):e0177678.

Chiang, D., Diab, M., Habash, N., Rambow, O., and Shareef, S. (2006). Parsing Arabic dialects. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

de Francony, G., Guichard, V., Joshi, P., Afli, H., and Bouchekif, A. (2019). Hierarchical Deep Learning for Arabic Dialect Identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 249–253.

Derczynski, L. (2016). Complementarity, F-score, and NLP Evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 261–266.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diab, M. and Habash, N. (2014). Natural language processing of Arabic and its dialects. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP). Doha, Qatar*, page 10. Citeseer.

Duwairi, R. M., Marji, R., Sha'ban, N., and Rushaidat, S. (2014). Sentiment analysis in Arabic tweets. In *2014 5th International Conference on Information and Communication Systems (ICICS)*, pages 1–6. IEEE.

Elawady, R. M., Barakat, S., and Elrashidy, N. M. (2014). Different feature selection for sentiment classification. *International Journal of Information Science and Intelligent System*, 3(1):137–150.

Fares, Y., El-Zanaty, Z., Abdel-Salam, K., Ezzeldin, M., Mohamed, A., El-Awaad, K., and Torki, M. (2019). Arabic Dialect Identification with Deep Learning and Hybrid Frequency Based Features. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 224–228.

Guellil, I., Saâdane, H., Azouaou, F., Gueni, B., and Nouvel, D. (2019). Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*.

Habash, N., Rambow, O., and Kiraz, G. A. (2005). Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 17–24.

Heikal, M., Torki, M., and El-Makky, N. (2018). Sentiment analysis of Arabic Tweets using deep learning. *Procedia Computer Science*, 142:114–122.

Kwaik, K. A., Saad, M., Chatzikyriakidis, S., and Dobnik, S. (2018). Shami: A corpus of levantine arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Kwaik, K. A., Saad, M., Chatzikyriakidis, S., and Dobnik, S. (2019). LSTM-CNN Deep Learning Model for Sentiment Analysis of Dialectal Arabic. In *International Conference on Arabic Language Processing*, pages 108–121. Springer.

Kwaik, K. A., Chatzikyriakidis, S., Dobnik, S., Saad, M., and Johansson, R. (2020). An arabic tweets sentiment analysis dataset (atsad) using distant supervision and self training. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 1–8.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage

lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Meftouh, K., Harrat, S., Jamoussi, S., Abbas, M., and Smaili, K. (2015). Machine translation experiments on PADIC: A parallel Arabic dialect corpus. In *The 29th Pacific Asia conference on language, information and computation*.

Meftouh, K., Abidi, K., Harrat, S., and Smaili, K. (2019). The SMarT Classifier for Arabic Fine-Grained Dialect Identification.

Mishra, P. and Mujadia, V. (2019). Arabic Dialect Identification for Travel and Twitter Text. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 234–238.

Mohammed, A. and Kora, R. (2019). Deep learning approaches for Arabic sentiment analysis. *Social Network Analysis and Mining*, 9(1):52.

Mountassir, A., Benbrahim, H., and Berrada, I. (2012). An empirical study to address the problem of unbalanced data sets in sentiment classification. In *2012 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 3298–3303. IEEE.

Nabil, M., Aly, M., and Atiya, A. (2015). Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2515–2519.

Obeid, O., Salameh, M., Bouamor, H., and Habash, N. (2019). ADIDA: Automatic dialect identification for Arabic. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 6–11, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Omar, N., Albared, M., Al-Shabi, A. Q., and Al-Moslmi, T. (2013). Ensemble of classification algorithms for subjectivity and sentiment analysis of Arabic customers' reviews. *International Journal of Advancements in Computing Technology*, 5(14):77.

Qwaider, C., Chatzikyriakidis, S., and Dobnik, S. (2019). Can Modern Standard Arabic Approaches be used for Arabic Dialects? Sentiment Analysis as a Case Study. In *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics*, pages 40–50.

Ragab, A., Seelawi, H., Samir, M., Mattar, A., Al-Bataineh, H., Zaghloul, M., Mustafa, A., Talafha, B., Freihat, A. A., and Al-Natsheh, H. (2019). Mawdoo3 AI at MADAR Shared Task: Arabic Fine-Grained Dialect Identification with Ensemble Learning. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 244–248.

Safaya, A., Abdullatif, M., and Yuret, D. (2020). Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059.

Shoukry, A. and Rafea, A. (2012). Sentence-level Arabic sentiment analysis. In *2012 International Conference on Collaboration Technologies and Systems (CTS)*, pages 546–550. IEEE.

Soliman, A. B., Eissa, K., and El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Soumeur, A., Mokdadi, M., Guessoum, A., and Daoud, A. (2018). Sentiment Analysis of Users on Social Networks: Overcoming the challenge of the Loose Usages of the Algerian Dialect. *Procedia computer science*, 142:26–37.

Suárez, P. J. O., Romary, L., and Sagot, B. (2020). A monolingual approach to contextualized word embeddings for mid-resource languages. *arXiv preprint arXiv:2006.06202*.

Tachicart, R., Bouzoubaa, K., Aouragh, S. L., and Jaafa, H. (2017). Automatic identification of Moroccan colloquial Arabic. In *International Conference on Arabic Language Processing*, pages 201–214. Springer.

Talafha, B., Ali, M., Za'ter, M. E., Seelawi, H., Tuffaha, I., Samir, M., Farhan, W., and Al-Natsheh, H. T. (2020). Multi-Dialect Arabic BERT for Country-Level Dialect Identification. *arXiv preprint arXiv:2007.05612*.

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

Zbib, R., Malchiodi, E., Devlin, J., Stallard, D., Matsoukas, S., Schwartz, R., Makhoul, J., Zaidan, O., and Callison-Burch, C. (2012). Machine translation of Arabic dialects. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 49–59.

Zhang, C. and Abdul-Mageed, M. (2019). No Army, No Navy: Bert Semi-supervised Learning of Arabic Dialects. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 279–284.