Arabic Dialect Identification for Travel and Twitter Text

Pruthwik Mishra and **Vandan Mujadia**
IIIT, Hyderabad
{pruthwik.mishra,vandan.mu}@research.iiit.ac.in

Abstract

This paper presents the results of the experiments done as a part of MADAR Shared Task in WANLP 2019 on Arabic Fine-Grained Dialect Identification. Dialect Identification is one of the prominent tasks in the field of Natural language processing where the subsequent language modules can be improved based on it. We explored the use of different features like char, word n-gram, language model probabilities, etc on different classifiers. Results show that these features help to improve dialect classification accuracy. Results also show that traditional machine learning classifier tends to perform better when compared to neural network models on this task in a low resource setting.

1 Introduction

In general, Arabic (language), refers to a wide spectrum of native languages used in Middle East and North Africa. As mentioned in Zaidan and Callison-Burch (2014), native languages of Arabic speakers differ with each other and with Modern Standard Arabic (MSA). These native languages or dialects can be categorized based on their common linguistic features and geographical locations (Elaraby and Abdul-Mageed, 2018). This categorization is described in detail in Bouamor et al. (2019). In the technological expansion of communication era, automatic identification of these dialects becomes an essential task for major natural language applications. These applications can be Machine Translation (Ling et al., 2013), Speech Recognition (Bouamor et al., 2018), Tourist Guide (Alshutayri and Atwell, 2017), Real-time Disaster Management (Elaraby and Abdul-Mageed, 2018; Alkhatib et al., 2019) and in health care. The task at hand was to identify a natural language dialect given a sequence of text for Arabic (Salameh and Bouamor, 2018). As per the shared tasks, these texts were either tourist help guide (subtask1) or the social media text (subtask2).

2 Related Work

Dialect identification is well known task in the Natural Language processing community. We can find work on different languages like English, German, Chinese, etc (Jauhiainen et al., 2018) for natural language dialect processing. Mostly it can be categorized into spoken and text level tasks. These categorization also includes work on resource creation for dialects (Zaidan and Callison-Burch, 2014; Zampieri et al., 2018) as well as the building a robust system for Dialect Identification. In Arabic, it is prerequisite for most NLP tasks, where many subsequent tasks depend on it. We can find spoken dialect identification work in Biadsy et al. (2009); Najafian et al. (2018); Shon et al. (2017), etc. For text, one can find recent work in Elaraby and Abdul-Mageed (2018); Salameh and Bouamor (2018); Abdul-Mageed et al. (2018); Butnaru and Ionescu (2018); Guellil et al. (2019).

MADAR shared task (Bouamor et al., 2019) consists of two sub-tasks which are

- MADAR Travel Domain Dialect Identification - this subtask requires identification of the dialect of a sentence, the dialect can be of any one of the pre-defined 26 arabic dialects as described in Bouamor et al. (2019)
- MADAR Twitter User Dialect Identification - this subtask requires the origin country of a tweet for a given user. We consider this classification task as a pipeline of 2 tasks. First we classify each tweet according to its country. Each user can tweet several times. The user to country mapping is decided based on frequency of the previous classification task. Each user is mapped to the most likely country predicted by the tweets s/he posts.

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 234–238
Florence, Italy, August 1, 2019.

We utilized features and model described in Salameh and Bouamor (2018) as baselines for Arabic dialect identification on Corpus-26 (Bouamor et al., 2018). We wanted to replicate their model which used multinomial naive bayes classifier (Pedregosa et al., 2011) on character and word n-gram with language model score as features to get state of the art accuracy.

3 Data

The details of the datasets used for training, development and test, in different subtasks are given in the tables 1 and 2. In table 1, the training data was distributed into 26 classes named as MADAR-Corpus-26 where each class had 1600 samples. Each class had a representation of 200 samples in the dev data.

Type	#Sentences
train	41600
dev	5200
test	5200

Table 1: Corpus Details for subtask1

Type	#Users	#Tweets
train	2180	217592
dev	300	29869
test	500	49962

Table 2: Corpus Details for subtask2

4 Experimental Setup

4.1 Prepossessing

Preprocessing is a necessary step while handing textual data. The preprocessing steps involved in the subtasks are detailed below:-

- **Tokenization and Normalization :** We did not use any off-the-shelf tokenizer for the tweets. We used the standard technique of tokenizing the text on white spaces for both the tasks.
- **Text cleaning (Tweets) :** Unlike standard texts, tweets can contain different spelling variations of words, special characters, twitter handles, urls due to limited space. We tried different experiments to observe the impact of removal of the twitter handles and urls on the overall classification accuracy. We observed that removal of these terms adversely affects the classification score. So we chose to keep the tweets as they were.

4.2 Feature Engineering

The features used for subtask1 were similar to those used in Salameh and Bouamor (2018). 3 different machine learning models were explored. All the below mentioned models were implemented using scikit-learn (Pedregosa et al., 2011) machine learning library.

- Linear SVM
- Multinomial Naive Bayes
- Logistic Regression

The individual features used in different subtasks are explained in detail here.

- **Subtask1**
  - **TF-IDF:** We used different combinations of word and character level n-grams for the tasks. We observed that combining word and character level n-gram TF-IDF vectors performed significantly better than individual word or character TF-IDF vectors. For our final submissions, combinations of word unigrams and character level n-grams were considered where n lies in $\{2, 3, 4, 5\}$.
  - **Language Modeling:** We trained different language models (LM) for the two types of corpora available to us. We trained the language model on sentences specific to a particular class for both MADAR-Corpus-6 (6 LMs) and MADAR-Corpus-26 (26 LMs). 2 features were included for these language models while developing machine learning models for subtask1. The coarse probabilities mentioned in table 3 came from the scores of the language model trained on MADAR-6 corpus. The final language model score was arrived at by adding the scores of the word and character 5-gram LMs for both the corpora.
- **Subtask2** For the first classification task in subtask2, we used the same word, character TF-IDF features and the same classifiers as

Model	N-gram Features		Other Features	P	R	F1	Acc
	Word	Char					
Baseline			Word 5-gram + Char 5-gram LM	67.7	67.4	67.4	67.4
mNB	1	1+2+3		64.9	63.9	63.7	63.9
mNB	1	2+3+4+5		66.3	65.0	64.9	65.0
mNB	1	1+2+3	Word&Char-5gram LM+Corpus 6 probs	**67.7**	**67.5**	**67.5**	**67.5**
mNB	1	2+3+4+5	Word&Char-5gram LM+Corpus 6 probs	**67.7**	**67.5**	67.4	**67.5**
SVM	1	1+2+3		64.3	63.9	63.9	63.9
SVM	1	2+3+4+5		64.8	64.4	64.4	64.4
SVM	1	1+2+3	Word&Char-5gram LM+Corpus 6 probs	67.7	67.4	67.4	67.4
SVM	1	2+3+4+5	Word&Char-5gram LM+Corpus 6 probs	67.7	67.4	67.4	67.4
logreg	1	1+2+3		64.4	64.0	63.9	64.0
logreg	1	2+3+4+5		65.3	65.0	65.0	65.0
logreg	1	1+2+3	Word&Char-5gram LM+Corpus 6 probs	67.7	67.4	67.4	67.4
logreg	1	2+3+4+5	Word&Char-5gram LM+Corpus 6 probs	67.7	67.4	67.4	67.4
MLP	1+2	1+2+3+4+5	50 neurons	65.12	64.17	64.37	64.17
MLP	1+2	1+2+3+4+5	100 neurons	66.68	65.9	66.0	65.9
MLP	1+2	1+2+3+4+5	200 neurons	67.39	66.63	66.78	66.63
MLP	1+2	1+2+3+4+5	50 neurons + Char LM	67.05	66.85	66.82	66.85
MLP	1+2	1+2+3+4+5	100 neurons + Char LM	66.83	66.67	66.62	66.67
MLP	1+2	1+2+3+4+5	200 neurons + Char LM	67.76	66.60	66.55	66.60

Table 3: Results On Dev Set for subtask1

mentioned in subtask1. We used the dialect probabilities as an additional feature which were present in the column 4 in the provided data. These dialect probabilities were obtained by the best model in Salameh and Bouamor (2018). We followed an ensemble approach for the classification task. Some of the tweets were unavailable in the training set. Some tweets consisted of only english tokens, so the arabic dialect probabilities were missing for those tweets. So we used two separate classifiers with the following features to handle data of different types

- Word Unigram, Character 2-5 gram TF-IDF vectors, dialect probabilities for the tweets which contained arabic text
- Word Unigram, Character 2-5 gram TF-IDF vectors for the tweets which contained no arabic text or contained only urls or twitter handles

During testing, different classifiers were used for inferencing with appropriate feature. We marked 'Saudi_Arabia' as the country of origin for a tweet which was unavailable because most of the tweets in the training set were from the users of Saudi Arab.

4.3 Deep Models

For subtask1, We have also tried out deep learning based classifier, where we used character and word level TF-IDF features as described above as input to the multi-layer perception (MLP). Here we used

Model	N-gram Features		Other Features	P	R	F1	Acc
	Word	Char					
SVM	1	1+2+3		87.8	49.8	60.0	66.3
SVM	1	2+3+4+5		88.0	50.0	60.07	66.7
SVM	1	2+3+4+5	Dialect Probabilities	87.9	49.6	59.8	66.3

Table 4: Results on Dev Set for subtask2

Subtask-Model	N-gram Features		Other Features	P	R	F1	Acc
	Word	Char					
subtask1-mNB	1	1+2+3	Word&Char-5gram LM+Corpus 6 probs	66.56	66.31	66.21	66.31
subtask2-SVM	1	1+2+3	Dialect Probabilities	83.37	47.73	57.90	67.20

Table 5: Results On Test Set for subtask1 and subtask2

sequential pipeline of keras[1] which contains one dense layer (with ReLU (Li and Yuan, 2017) activation) and output layer with softmax activation with categorical_crossentropy as loss function and Adam as optimizer. We trained this classifier for 30 epochs with early stopping criteria on GeForce GTX 1060 GPU. In result section, we show and discuss results in detail.

5 Observations

We could observe that all the classifiers performed similarly when all the features were used. Combination of character and word level TF-IDF vectors performed better than character or word level TF-IDF vectors in isolation. We could see that the language models trained at word and character level were the biggest contributor to the system's performance for subtask1. TF-IDF features and coarse probabilities did not add much to the overall accuracy. Logistic Regression and multinomial naive bayes techniques performed significantly poor for subtask2, so we did not report the results in this paper. Machine learning approaches performed marginally better than the multi-layer perceptrons. This could be due to the higher number of parameters that deep learning approaches try to learn compared to traditional approaches. One of the main reasons for lower classification accuracy in subtask2 is our assumption to assign country of origin for unavailable tweets as 'Saudi_Arabia'. There were 5992 unavailable tweets in the test corpus.

[1] https://keras.io

6 Conclusion and Future Work

We presented our experiments on supervised dialect identification task (MADAR) in Arabic. Our experiments demonstrate that for relatively low resource task such as MADAR, traditional machine learning algorithms with feature engineering show their potentials compared to the deep learning approaches. Unlabelled Arabic corpora can be used to learn character and word embeddings in Arabic. It would be an interesting area to explore how recurrent neural networks perform on this task.

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)*.

Manar Alkhatib, May El Barachi, and Khaled Shaalan. 2019. An arabic social media based framework for incidents and events monitoring in smart cities. *Journal of Cleaner Production*, 220:771–785.

Areej Alshutayri and Eric Atwell. 2017. Exploring twitter as a source of an arabic dialect corpus. *International Journal of Computational Linguistics (IJCL)*, 8(2):37–44.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the eacl 2009*

workshop on computational approaches to semitic languages, pages 53–61. Association for Computational Linguistics.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

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 (WANLP19)*, Florence, Italy.

Andrei M Butnaru and Radu Tudor Ionescu. 2018. Unibuckernel reloaded: First place in arabic dialect identification for the second year in a row. *arXiv preprint arXiv:1805.04876*.

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2019. Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic language identification in texts: A survey. *arXiv preprint arXiv:1804.08186*.

Yuanzhi Li and Yang Yuan. 2017. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607.

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.

Maryam Najafian, Sameer Khurana, Suwon Shan, Ahmed Ali, and James Glass. 2018. Exploiting convolutional neural networks for phonotactic based dialect identification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5174–5178. IEEE.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Mohammad Salameh and Houda Bouamor. 2018. Fine-grained arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344.

Suwon Shon, Ahmed Ali, and James Glass. 2017. Mitqcri arabic dialect identification system for the 2017 multi-genre broadcast challenge. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 374–380. IEEE.

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

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, et al. 2018. Language identification and morphosyntactic tagging: The second vardial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects*. Association for Computational Linguistics.