# Team Alexa at NADI Shared Task

**Mutaz Bni Younes  Nour Al-Khdour  Mohammad AL-Smadi**
mmbniyounes18 , naalkhdour17@cit.just.edu.jo  maalsmadi9@just.edu.jo

Department of Computer Science
Jordan University of Science and Technology
Irbid, Jordan

## Abstract

In this paper, we discuss our team's work on the NADI Shared Task. The task requires classifying Arabic tweets among 21 dialects. We tested out different approaches, and the best one was the simplest one. Our best submission was using Multinational Naive Bayes classifier with n-grams as features. Our best submitted score on the test phase was 17% F1-score and 35% accuracy. However, in the post-evaluation phase we used an ensemble model including BERT and Multinational Naive Bayes classifier and it outperformed the top submission on the task, this ensemble model achieved 27.73% F1-score and 40.90% accuracy.

## 1 Introduction

The interest of the research community concerning the Arabic natural language processing (NLP) currently is focused on dialect identification at several levels, region level, country level, and provinces. Most previous works focused on Modern Standard Arabic (MSA) (Elfardy and Diab, 2013) (Al-Sabbagh and Girju, 2012) because it is commonly used in formal writing between Arab countries. Many previous work on Arabic dialect classification used a combination of n-gram both on word and char level with Multinomial Naive Bayes such as (Salameh et al., 2018; Meftouh et al., 2019; Talafha et al., 2019). Eldesouki et al. (2016) successfully applied SVM. Zhang and Abdul-Mageed (2019) proposed a semi-supervised model with BERT and obtained the top rank for MADAR Twitter User Dialect Identification subtask in the MADAR Shared task (Bouamor et al., 2019). MADAR corpus was the first large-scale resource built for Arabic dialects (Bouamor et al., 2018).

The Nuanced Arabic Dialect Identification (NADI) (Abdul-Mageed et al., 2020) provided a labeled dataset consisting of Arabic tweets with two subtasks: the first subtask is based on country-level dialect identification and the second subtask is province-level dialect identification. The dataset was challenging and the same dataset was used for both subtasks, even the algorithms that achieved high results in similar tasks did not gain satisfying results. In this paper, we focused on the first subtask. Alexa model was built using a weighted ensemble model with n-grams features in the word and character levels. The ensemble model consists of the OneVsRest classifier with MNB, MNB, and Logistic Regression. Our model obtained a 17% F1-score and 35% accuracy; it ranks twelve based on F1 out of 18 participants.

The remainder of the paper is organized as follows: data analysis will be shown in section 2. Section 3 proposes a description of the Alexa model. The results for subtask 1 are discussed in Sections 4, and 5 respectively, followed by the conclusion in Section 6.

## 2 Data

The NADI shared task contains two subtasks; both use the same training and development data but differ in labels. The number of training, development, and testing datasets are shown in Table 1.





*Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 237–242
Barcelona, Spain (Online), December 12, 2020

Table 1: The distribution of the dataset

| Corpus | Train | Dev | Test |
|---|---|---|---|
| NADI | 21000 | 4957 | 5000 |

The first subtask contains country-level dialects as labels, where the second subtask contains province-level dialects as labels. The dataset has a total of 100 provinces, all of them are from 21 Arab countries.

The dataset is collected from tweets in different domains. It is highly imbalanced data; the number of classes for each country is shown in Table 2.

Table 2: Label Distribution

| Country | Number | Country | Number |
|---|---|---|---|
| Bahrain | 210 | Yemen | 851 |
| Djibouti | 210 | Syria | 1,070 |
| Sudan | 210 | Morocco | 1,070 |
| Mauritania | 210 | United_Arab_Emirates | 1,070 |
| Somalia | 210 | Libya | 1,070 |
| Qatar | 234 | Oman | 1,098 |
| Kuwait | 420 | Algeria | 1,491 |
| Palestine | 420 | Saudi_Arabia | 2,312 |
| Jordan | 426 | Iraq | 2,556 |
| Lebanon | 639 | Egypt | 4,473 |
| Tunisia | 750 | – | – |

### 2.1 Pre-processing

We experimented with different pre-processing settings, such as removing the links, @username, additional white spaces, punctuations, English alphabets, emojis, diacritics, repeated characters, and the non-Arabic tweets that use Arabic alphabets such as Pashto, Urdu, and Persian. However, preprocessing the data had a negative effect contrary to the expected; the results decreased, so we concluded that training the classifiers without preprocessing would be more effective.

### 2.2 Unlabeled Tweets

The organizers included 10 million unlabeled tweets. Some previous work used such data to generate more training samples. In Zhang and Abdul-Mageed (2019), the authors used self-learning method to augment the training dataset. This method increased their baseline's accuracy by 3% and their F1-score by 6%. However, we did not use this dataset since all our experiments exploiting it did not yield any improvements.

## 3 System

In this section, we propose our system (Alexa model) that consists of multiple steps. In feature extraction level, we extract features from the tweets such as language models (n-grams) (Brown et al., 1992) that assigns probabilities to a specific number of sequences of words or characters. There are many experiments done by combining different sets of language models on word level and character level (Talafha et al., 2019) and (Meftouh et al., 2019). On the word level, unigrams and bigrams (1,2) were the best, and on character level we end up with n_grams range from 1 to 5 character with two types of tfidf vectorizers as shown in the Figure 1 "char" and "char_wb." "char_wb" characters with the word boundaries. The features were weighted as follows: word-level unigram and bigram features weighted as 0.8, character-level Tfidf Vectorizer "char_wb" weighted as 1.1 and Tfidf Vectorizer "char" weighted as 1.0.

Then, the extracted features were concatenated to train the ensemble model; the ensemble model is composed of One Vs Rest strategy with MNB (Small and Hsiao, 1985), MNB, and Logistic regression (Kleinbaum et al., 2002). The prediction form the three classifiers are summed to produce the one label for each tweet. The parameters used to train each classifier:

1. One Vs Rest strategy with MNB parameters: alpha equal 0.05 and the default values for the rest parameters.
2. MNB parameters: alpha equal 0.02 and the default values for the rest parameters.
3. Logistic regression parameters: multi_class='ovr', and the default values for the rest parameters. When the assigned parameter in multi_class is 'ovr', then the algorithm uses one-vs-rest (OvR) scheme.

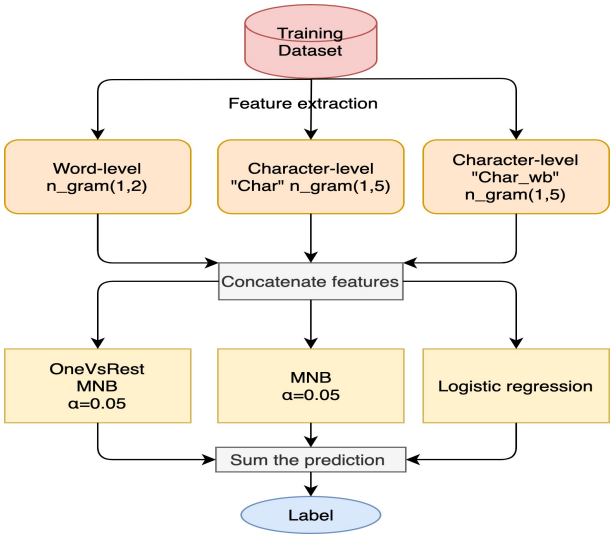


Figure 1: Alexa model architecture

## 4 Results

It is worth mentioning that using different pre-trained embedding models with this dataset did not perform well. We used BERT (Devlin et al., 2018) multilingual model and Aravec model (Soliman et al., 2017) to generate embeddings for the dataset. Both of them achieved low results when using them as main features or as extra features. On the other hand, using these models for training outperformed the best submission on the task. We will talk about the model and the results in the post-evaluation section.

### 4.1 Post-Evaluation

We tested out different ideas in the post-evaluation phase. We concatenated similar dialects into one label and used it to get extra features for the data we have. We ended up with four main dialects. "GULF", AFRICA", "LEVANT", and "MAGRIB". Stacking the probability of each dialect with the n-grams features did not boost the results by noticeable differences.

Another approach we tested is adding hand-written rules, adding a few rules boosted the F1-score by 2%. For example, we increase the probability of a tweet to be labeled as "Jordan" if we see the word "jordan" in it. However, some rules could lead to miss classifying some tweets.

#### 4.1.1 Ensemble model

Figure 2 shows our ensemble model that concatenates weighted predicted probabilities from bert-large-arabic model (Safaya et al., 2020) and MNB classifier. BERT-large-arabic model is one of ArabicBERT models, ArabicBERT released on four different sizes (Large, Base, Medium, and Mini), as well it was

trained on nearly 95 GB of Arabic text from Open Super-large Crawled and Wikipedia. Furthermore, training data was in Modern Standard Arabic and dialectical Arabic, in our opinion, this is the main reason to enhance the performance of dialect classification, as for this task because it produces a meaningful embedding representation.

The predicted probabilities were multiplied by weights in order to get the highest F1-score possible for the development dataset, the final weights were determined for both classifiers based on multiple experiments as follows: 0.35 for MNB probabilities and 1.4 for BERT probabilities. This model outperformed the best submission on the task. It achieved 27.73% F1-score and 40.90% accuracy. The parameters used to train each classifier:

1. One Vs Rest strategy with MNB parameters: alpha equal 0.01 and the default values for the rest of the parameters. The features were weighted as follows: word-level unigram and bigram features weighted as 0.8, character-level Tfidf Vectorizer "char_wb" weighted as 1.1 and Tfidf Vectorizer "char" weighted as 1.0.

2. BERT-large-arabic parameters: num_train_epochs equal 2, learning_rate equal 2e-5, and the default values for the rest of the parameters.

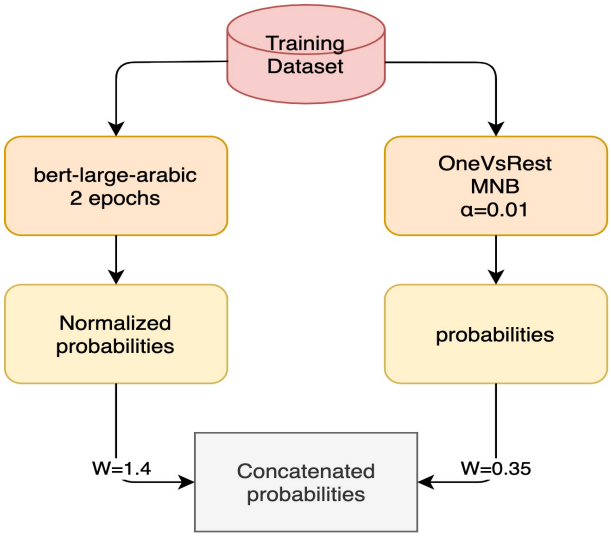


Figure 2: The ensemble model architecture

## 5 Discussion

The provided dataset did not include Modern Standard Arabic (MSA) label in the training and development dataset, see Figure 3 which contains wrongly classified tweets, the tweets in Figure 3 are written in MSA.

| Tweet | Label |
|---|---|
| اللهم طهر قلبي من النفاق ، وعملي من الرياء ، ولساني من الكذب ، وعيني من الخيانة<br>May Allah clean my heart and my work from hypocrisy, and my tongue from lying, and my eyes from betrayal | Egypt |
| انا لله و انا اليه راجعون الله يرحمه و يصبر ذويه<br>We belong to Allah and will return to him, may Allah have mercy on him and help his parents to be patient about their lost. | Morocco |
| ولن يخلف الله وعده<br>God will never renege his promise | Egypt |

Figure 3: examples of wrongly classified tweets

Our team investigated this problem; we used a model that was trained on detecting MSA text to extract the wrongly labeled tweets in our dataset. To train this model, we used MADAR corpus (Bouamor et al., 2018) because it contains MSA tweets and other dialects. Our used model achieves an accuracy score higher than the top submitted score on the MADAR Shared task. Such that, we assume that this model can accurately classify MSA tweets. We also used different Arabic datasets to see if our claim is true or not and we got similar results. Based on our system, more than 20% of the training and development tweets were MSA tweets. Table 3 shows our numbers.

Table 3: Number of MSA Tweets in Each Set

| | Train | Dev | Test |
|---|---|---|---|
| Total | 21,000 | 4,957 | 5,000 |
| Estimated MSA | 4,930 | 1,074 | 1,074 |

To test our findings, we re-labeled MSA tweets to ”MSA” and then applied multinomial naive bayes classifier on the development dataset, the results were as shown on table 4.

Table 4: Results after adding the MSA label

| | F1-score | Accuracy |
|---|---|---|
| With MSA | 14.9 | 43.837 |
| Without MSA | 14.9 | 35.203 |

In Table 4, we noticed that the F1-score decreased when re-labeling the dataset, this because some tweets are written mostly in MSA but may contain some dialect words, so without having the MSA label, these tweets will be classified correctly to their labeled (provided label) class. But when adding the MSA label, such tweets will be classified as MSA tweets, which decreases the overall F1-score for many classes and increases it for the MSA class.

## 6 Conclusion

In this paper, we tested different approaches in an Arabic dialect classification task, NADI shared task. The best submitted score on the test phase was using an ensemble model that contains Multinational naive Bayes, OneVsRest MNB, and logistic regression. For features, we used both CountVectorizer and TfidfVectorizer features. Our best submission achieved 17% F1-score. However, in the post-evaluation phase, we used an ensemble model which outperformed the best submitted score on the task. Our ensemble model contained weighted concatenation between BERT's probabilities and MNB probabilities. This model achieved 27.73% F1-score and 40.90% accuracy. We also noticed that many tweets were wrongly labeled because there were tweets written in MSA, and there was no MSA label. Also, there are six countries with less than 240 tweets; this leads the model to never predict these dialects. For future work, we would like to overcome these issues by finding a way to deal with the imbalanced dataset and a way to overcome the MSA problem.

## References


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 (WANLP 2020)*, Barcelona, Spain.

Rania Al-Sabbagh and Roxana Girju. 2012. Yadac: Yet another dialectal arabic corpus. In *LREC*, pages 2882–2889.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

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.

Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mohamed Eldesouki, Fahim Dalvi, Hassan Sajjad, and Kareem Darwish. 2016. Qcri@ dsl 2016: Spoken arabic dialect identification using textual features. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 221–226.

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)*, pages 456–461.

David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. 2002. *Logistic regression*. Springer.

Karima Meftouh, Karima Abidi, Salima Harrat, and Kamel Smaili. 2019. The smart classifier for arabic fine-grained dialect identification.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media.

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

Kenneth A Small and Cheng Hsiao. 1985. Multinomial logit specification tests. *International economic review*, pages 619–627.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Bashar Talafha, Ali Fadel, Mahmoud Al-Ayyoub, Yaser Jararweh, AL-Smadi Mohammad, and Patrick Juola. 2019. Team just at the madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 285–289.

Chiyu Zhang and Muhammad Abdul-Mageed. 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.