Multi-Dialect Arabic BERT for Country-Level Dialect Identification

Bashar Talafha* **Mohammad Ali*** **Muhy Eddin Za'ter** **Haitham Seelawi**
Ibraheem Tuffaha **Mostafa Samir** **Wael Farhan** **Hussein T. Al-Natsheh**

Mawdoo3 Ltd, Amman, Jordan
{bashar.talafha,mohammad.ali,muhy.zater,haitham.selawi,
ibraheem.tuffaha,mostafa.samir,wael.farhan,h.natsheh}
@mawdoo3.com

Abstract

Arabic dialect identification is a complex problem for a number of inherent properties of the language itself. In this paper, we present the experiments conducted, and the models developed by our competing team, Mawdoo3 AI, along the way to achieving our winning solution to subtask 1 of the Nuanced Arabic Dialect Identification (NADI) shared task. The dialect identification subtask provides 21,000 country-level labeled tweets covering all 21 Arab countries. An unlabeled corpus of 10M tweets from the same domain is also presented by the competition organizers for optional use. Our winning solution itself came in the form of an ensemble of different training iterations of our pre-trained BERT model, which achieved a micro-averaged F1-score of 26.78% on the subtask at hand. We publicly release the pre-trained language model component of our winning solution under the name of Multi-dialect-Arabic-BERT model, for any interested researcher out there.

1 Introduction

The term Arabic language is better thought of as an umbrella term, under which it is possible to list hundreds of varieties of the language, some of which are not even mutually comprehensible. Nonetheless, such varieties can be grouped together with varying levels of granularity, all of which correspond to the various ways the geographical extent of the Arab world can be divided, albeit loosely. Despite such diversity, up until recently, such varieties were strictly confined to the spoken domains, with Modern Standard Arabic (MSA) dominating the written forms of communication all over the Arab world. However, with the advent of social media, an explosion of written content in said varieties have flooded the internet, attracting the attention and interest of the wide Arabic NLP research community in the process. This is evident in the number of held workshops dedicated to the topic in the last few years.

In this paper we present and discuss the strategies and experiments we conducted to achieve the first place in the Nuanced Arabic Dialect Identification (NADI) Shared Task 1 (Abdul-Mageed et al., 2020), which is dedicated to dialect identification at the country level. In section 2 we discuss related work. This is followed by section 3 in which we discuss the data used to develop our model. Section 4 discusses the most significant models we tested and tried in our experiments. The details and results of said experiments can be found in section 5. The analysis and discussion of the results can be obtained in section 6 followed by our conclusions in section 7.

2 Related Work

The task of Arabic dialect identification is challenging. This can be attributed to a number of reasons, including: a paucity of corpora dedicated to the topic, the lack of a standard orthography between and across the various dialects, and the nature of the language itself (e.g. its morphological richness among other peculiarities). To tackle these challenges, the Arabic NLP community has come up with a number of responses. One response was the development of annotated corpora that focus primarily on dialectical

*Equal contribution.

Proceedings of the Fifth Arabic Natural Language Processing Workshop, pages 111–118
Barcelona, Spain (Online), December 12, 2020

data, such as the Arabic On-line Commentary dataset (Zaidan and Callison-Burch, 2014), the MADAR Arabic dialect corpus and lexicon (Bouamor et al., 2018), the Arap-Tweet corpus (Zaghouani and Charfi, 2018), in addition to a city-level dataset of Arabic dialects that was curated by (Abdul-Mageed et al., 2018). Another popular form of response is the organization of NLP workshops and shared tasks, which are solely dedicated to developing approaches and models that can detect and classify the use of Arabic dialects in written text. One example is the MADAR shared task (Bouamor et al., 2019), which focuses on dialect detection at the level of Arab countries and cities.

The aforementioned efforts by the Arabic NLP community, have resulted in a number of publications that explore the application of a variety of Machine Learning (ML) tools to the problem of dialect identification, with varying emphasis on feature engineering, ensemble methods, and the level of supervision involved (Salameh et al., 2018; Elfardy and Diab, 2013; Huang, 2015; Talafha et al., 2019b).

The past few years have also witnessed a number of published papers that explore the potential of Deep Learning (DL) models for dialect detection, starting with (Elaraby and Abdul-Mageed, 2018; Ali, 2018), who show the enhanced performance that can be brought about through the use of LSTMs and CNNs, all the way to (Zhang and Abdul-Mageed, 2019), who highlight the potential of pre-trained language models to achieve state of the art performance on the task of dialect detection.

3 Dataset

The novel dataset of NADI shared task consists of around 31,000 labeled tweets covering the entirety of the 21 Arab countries. Additionally, the task presents an unlabeled corpus of 10M tweets. The labeled dataset is split into 21,000 examples for training, with the rest of the tweets, i.e., 10,000, distributed equally between the development and test sets. Each tweet is annotated with a single country only. In Figure 1 we can see the distribution of tweets per country in which *Egypt* and *Bahrain* has the highest and lowest tweet frequencies, respectively. We also note that the ratio of the development to train examples is generally similar across the various dialects, except for the ones with lowest frequencies.

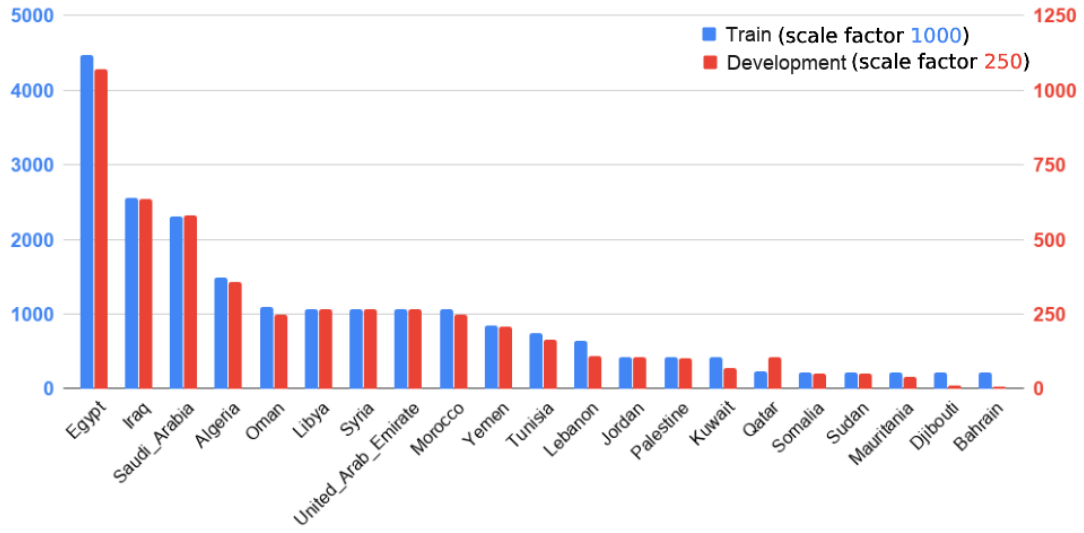

Figure 1: Classes distribution for both Train and Development sets

The unlabeled dataset is provided in the form a twitter crawling script, and the IDs of 10M tweets, which in combination can be used to retrieve the text of these tweets. We are able to retrieve 97.7% of them, as the rest seem to be unavailable (possibly deleted since then or made private). This dataset can be beneficial in multiple ways, including building embedding models (e.g., Word2Vec, FastText) (Mikolov et al., 2013; Bojanowski et al., 2017), pre-training language models, or in semi-supervised learning and data augmentation techniques. This dataset can also be used to further pre-train an already existing language model, which can positively affect its performance on tasks derived from a domain similar to that of the 10M tweets, as we show in our results in Section 6.

4 System Description

In this section, we present the various approaches employed in our experiments, starting with the winning approach of our Multi-dialect-Arabic-BERT model, followed by the rest of them. The results of our experiments are presented in Table 1.

4.1 Multi-dialect-Arabic-BERT

Our top performing approach, which achieved the number one place on the NADI task 1, is based on a Bidirectional Encoder Representations from Transformers (BERT) architecture (Devlin et al., 2018). BERT uses the encoder part of a Transformer (Vaswani et al., 2017), and is trained using a masked language model (MLM) objective. This involves training the model to predict corrupted tokens, which is achieved using a special mask token that replaces the original ones. This is typically done on a huge corpus of unlabeled text. The resultant model can then produce contextual vector representations for tokens that capture various linguistics signals, which in turn can be beneficial for downstream tasks.

We started with the *ArabicBERT* (Safaya et al., 2020), which is a publicly released BERT model trained on around 93 GB of Arabic content crawled from around the internet. This model is then fine-tuned on the NADI task 1, by retrieving the output of a special token [CLS], placed at the beginning of a given tweet. The retrieved vector is in turn fed into a shallow feed-forward neural classifier that consists of a dropout layer, a dense layer, and a softmax activation output function, which produces the final predicated class out of the original 21. It is worth mentioning that during the fine-tuning process, the loss is propagated back across the entire network, including the BERT encoder.

We then were able to significantly improve the results obtained from the model above, by further pre-training *ArabicBERT* on the 10M tweets released by the NADI organizers, for 3 epochs. We refer to this final resultant model as the *Multi-dialect-Arabic-BERT*.

In order to squeeze out more performance from our model, we ended up using ensemble techniques. The best ensemble results came from the voting of 4 models that were trained with different maximum sequence lengths (i.e., 80, 90, 100 and 250). The voting step was accomplished by taking the element-wise average of the predicted probabilities per class for each of these models. The class with the highest value is then outputed as the predicted label.

All of our models were trained using an Adam optimizer (Kingma and Ba, 2014) with a learning rate of 3.75×10^{-5} and a batch size of 16 for 3 epochs. No preprocessing was applied to the data except for the processing done by ArabicBERT tokenizer. The vocabulary size for ArabicBert is 32000 and sentencepiece (Kudo and Richardson, 2018) is used as a tokenizer.

We publicly release the Multi-dialect-Arabic-BERT[1] on GitHub to make it available for use by all researchers for any task including reproducing this paper's results.

4.2 Other Traditional Machine and Deep Learning Models

In addition to our winning solution, we experimented with a number of other approaches, none of which has exceeded an F1-score of 21, but which we list here for the sake of completeness anyway.

- *MADAR-Mawdoo3 Model*
  Originally proposed by Ragab et al. (2019), three models (i.e., a Multinomial Naive Bayes (MNB), logistic regression, and weak dummy classifier) are trained separately on the data to obtain their dialect probability distributions, which then, in conjunction with TF–IDF vectors, make up the feature space. These features are then fed into an ensamble of five other models (i.e., MNB with one-vs-rest strategy, a Support Vector Machine model (SVM), a Bernoulli Naive Bayes classifier, a K-nearest-neighbours classifier with one-vs rest strategy, and finally a weak dummy classifier). The final predicted classes are obtained using a hard voting approach.

- *MADAR-Safina Model*
  This model follows Safina model proposed in (Bouamor et al., 2019). The model is an ensemble

[1] https://github.com/mawdoo3/Multi-dialect-Arabic-BERT

Model	Dev Set Results		Test Set Results	
	Accuracy	F1-Score	Accuracy	F1-Score
MADAR-Safina	33.35	10.1	-	-
Logistic-Regression	35.65	16.57	-	-
MADAR-1 Mawdoo3	33.45	12.24	-	-
MADAR-1 JUST	30.3	17.07	-	-
FastText-embeddings	34.28	19.74	-	-
Aravec fully connected	35.67	20.86	-	-
Arabic-BERT-Single	40.85	24.45	-	-
Arabic-BERT-Ensemble-Diff-Len	41.48	24.92	-	-
Multi-dialect-Arabic-BERT	43.7	26	-	-
Multi-dialect-Arabic-BERT-Ensemble-Diff-Len	44.95	27.58	**42.86**	**26.78**
Multi-dialect-Arabic-BERT-Ensemble-Diff-Len with rules	**45.07**	**29.03**	42.55	26.77

Table 1: Final results on NADI development and testing set

of 3 classifiers: Language model classifier based on 5-char n-gram features (Heafield et al., 2013), Naive Bayes classifier based on 4-to-6 char n-gram features, Naive Bayes classifier based on 1-word n-gram features. The only pre-processing step used is to duplicate every single word for the language model and the char n-gram classifiers. The purpose of duplicating every single word is to detect circumfix n-gram patterns.

- *MADAR-JUST Model*
  In this model, we applied the approach proposed by Talafha et al. (2019a). In order to balance the training data, a data augmentation technique based on random shuffling was performed to enlarge and balance the training data. After that, for each sentence, a vector of size 21 that represents a language model probability for each country was extracted and concatenated to a word and character level TF–IDF vectors. An MNB classifier is then applied with the One-vs-the-rest strategy.

- *FastText Model*
  FastText (Bojanowski et al., 2017) was originally implemented to help obtain enhanced word representations over simpler methods such as Word2Vec (Mikolov et al., 2013). In our experiments, we pool the fastText vectors of each token in a given sentence, to obtain a fixed-size dense representation of the sentence at hand. This is in turn fed into a multinomial logistic regression for classification (Zolotov and Kung, 2017; Joulin et al., 2016).

- *AraVec fully connected Model*
  AraVec is an Arabic based Word2Vec model, trained and published by (Soliman et al., 2017). Similar to our FastText model above, we pool the AraVec vectors of the constituent tokens of a sentence to obtain its fixed-size vector representation. However, instead of a conventional ML algorithm, we feed these representations into a feed forward classifier, which is trained to obtain the final predictions (Ashi et al., 2018).

5 Experiments and Results

As mentioned above, multiple approaches have been investigated in the experiments we conducted, starting with traditional ML techniques then moving to DL approaches, before finally settling on our winning BERT based model. For our traditional ML experiments, we tried various models such as SVM, Logistic Regression (LR) and Naive Bayes (NB), along with features such as TF–IDF. We also tried other ML models that performed well on previous similar tasks such as MADAR Mawdoo3-AI and MADAR Safina models. However, all of these models came short when compared to the BERT models as can be seen in Table 1, with the best Macro-Averaged F1-score achieved using traditional ML approaches being

Model	Results	
	Accuracy	Macro-Averaged F1-Score
ArabicProcessors	38.34	23.26
BERT-NGRAMS	39.66	25.99
Mawdoo3-ai	**42.86**	**26.78**

Table 2: Final results on NADI testing dataset for the 3 top performing participating teams

17.06%. We then experimented with a number DL models, along with pre-trained word embedding features, such as fastText and Word2Vec. These models easily surpassed the performance of their traditional ML counterparts, with a maximum macro-averaged F1 score of 20.86%.

As alluded to above, the best results were achieved by our BERT models. Using the standalone ArabicBERT (Safaya et al., 2020) we were able to achieve 24.45% Macro-Averaged F1-score on the development dataset. This score was further increased to 26.46% using ensemble techniques. This motivated us to further pre-train it on the 10 million unlabelled tweets to form the Multi-dialect-Arabic-BERT model. Using this setup, we were able to achieve a Macro-Averaged Macro-Averaged F1 score of 26%. Here again, we used the ensemble trick to obtain a 27.58% Macro-Averaged F1-score on the development set and 26.78% on the test set, thus winning the competition. We note that applying lexicon-based prediction rules to the best model mentioned above boosted the results of development set to 29.03 F1-score. However, these rules slightly decreased the test set results to 26.77 F1-score, concluding that such rules cause the system to suffer from over-fitting the development set.

6 Discussion and Analysis

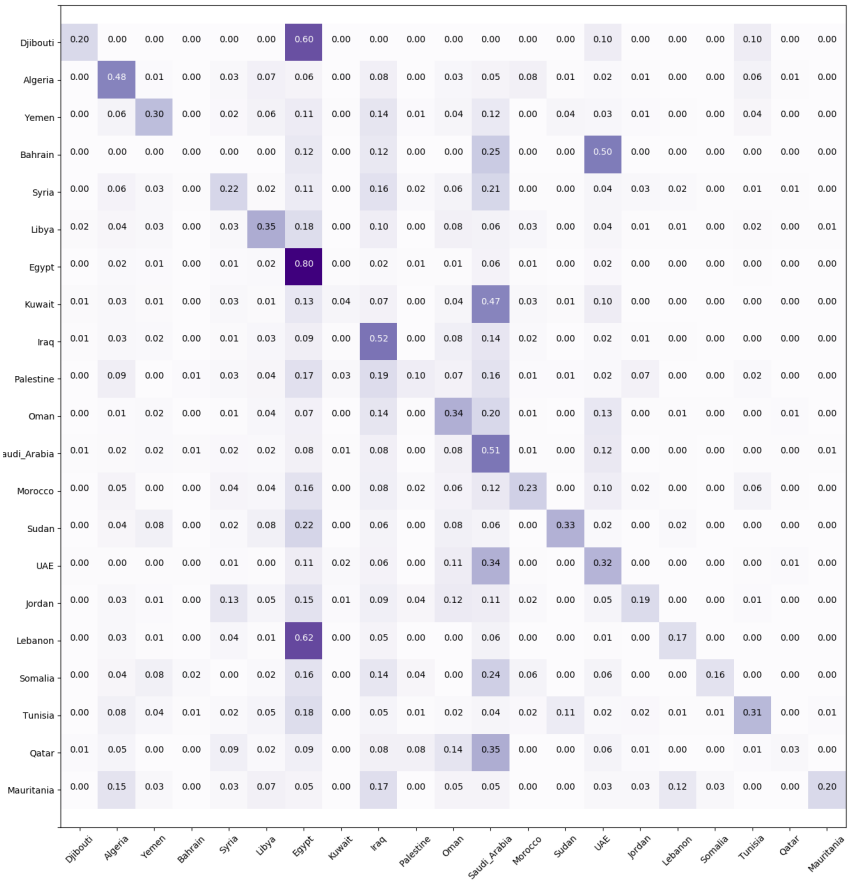

Figure 2: The confusion matrix of our best model on NADI development set

To aid us with the analysis of the strengths and weaknesses of our winning model, we provide the confusion matrix for its performance on the NADI development set in Figure 2. The matrix highlights a number of issues stemming from the training dataset itself. For instance, it can be clearly seen that the model is biased to the countries with more training data such as Egypt, Iraq and Saudi Arabia; for these countries, the model achieves better results, while achieving much worse F1-scores for the ones with the least training data available. It can also be seen that the model suffers when trying to differentiate between geographically nearby countries. For example, 50% of the development samples from Bahrain are labeled as UAE and 22% from Sudan are labeled as Egypt. This is expected, given the similarities in dialects between neighbouring countries. Some of the results shown in the confusion matrix have also led us to further investigate the datasets themselves. This resulted in finding that our model does in fact predict the correct class for certain tweets, which were somehow originally mislabeled. Some of these examples can be seen in Table 3.

Tweet	Label	Predicton	Actual
اللهم ارحمه واغفرله	Mauritania	Egypt	MSA
احنا ناس تافهة ملناش في الكلام ده	Lebanon	Egypt	Egypt
راسلني على الخاص	Algeria	Saudi_Arabia	MSA
وش اسويلك طيب؟؟؟	Syria	Saudi_Arabia	Saudi_Arabia
اللهم صل على محمد وال محمد	Palestine	Iraq	MSA
إي ديربالچ من قراراتي أبدًا مو ثابتة	Morocco	Iraq	Iraq

Table 3: Examples of mislabeled and confusing tweets.

7 Conclusion

In this paper we describe our first place solution for the NADI competition, task 1. This was achieved via three stages: firstly, we further pre-trained a publicly released BERT model (i.e., Arabic-BERT) on the 10 millions tweets supplied by the NADI competition organizers. Secondly, we trained the resultant model on the NADI labelled data for task 1, multiple times, independently, with each of these iterations using a different mixture of maximum sentence length and learning rate. Thirdly, we selected the 4 best performing iterations (based on their performance on the development dataset), and aggregated their softmax predictions via a simple element wise averaging function, to produce the final prediction for a given tweet. For future work, we would like to investigate other advanced pre-training methods, such as XLNET (Yang et al., 2019), and ELECTRA (Clark et al., 2020), which we believe might hold the key to better performance 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)*.

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.

Mohamed Ali. 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.

Mohammed Matuq Ashi, Muazzam Ahmed Siddiqui, and Farrukh Nadeem. 2018. Pre-trained word embeddings for arabic aspect-based sentiment analysis of airline tweets. In *International Conference on Advanced Intelligent Systems and Informatics*, pages 241–251. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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*, pages 199–207.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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 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.

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.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria, August. Association for Computational Linguistics.

Fei Huang. 2015. Improved arabic dialect classification with social media data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2118–2126.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ahmad Ragab, Haitham Seelawi, Mostafa Samir, Abdelrahman Mattar, Hesham Al-Bataineh, Mohammad Zaghloul, Ahmad Mustafa, Bashar Talafha, Abed Alhakim Freihat, and Hussein Al-Natsheh. 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.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

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.

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. 2019a. 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.

Bashar Talafha, Wael Farhan, Ahmed Altakrouri, and Hussein Al-Natsheh. 2019b. Mawdoo3 ai at madar shared task: Arabic tweet dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 239–243.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Wajdi Zaghouani and Anis Charfi. 2018. Arap-tweet: A large multi-dialect twitter corpus for gender, age and language variety identification. *arXiv preprint arXiv:1808.07674*.

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

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.

Vladimir Zolotov and David Kung. 2017. Analysis and optimization of fasttext linear text classifier. *arXiv preprint arXiv:1702.05531*.