# AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding

**Wissam Antoun** and **Fady Baly** and **Hazem Hajj**
American University of Beirut
{wfa07, fbg06, hh63}@aub.edu.lb

## Abstract

Advances in English language representation enabled a more sample-efficient pre-training task by Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA). Which, instead of training a model to recover masked tokens, it trains a discriminator model to distinguish true input tokens from corrupted tokens that were replaced by a generator network. On the other hand, current Arabic language representation approaches rely only on pretraining via masked language modeling. In this paper, we develop an Arabic language representation model, which we name AraELECTRA. Our model is pretrained using the replaced token detection objective on large Arabic text corpora. We evaluate our model on multiple Arabic NLP tasks, including reading comprehension, sentiment analysis, and named-entity recognition and we show that AraELECTRA outperforms current state-of-the-art Arabic language representation models, given the same pretraining data and with even a smaller model size.

## 1 Introduction

Recently, pre-trained language representation models have demonstrated state-of-the-art performance on multiple NLP tasks and in different languages. Pre-training is commonly done via Masked Language Modeling (MLM) (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2019), where an input sequence has some of its tokens randomly hidden and the model is tasked to recover the original masked tokens. While this approach has proven successful, recent works have shown that MLM is not sample-efficient (Clark et al., 2020b), since the network only learns from the small subset of masked tokens per sequence (15% of the tokens in BERT). Clark et al. (2020b) proposed an approach called Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA). The method uses a pre-training technique based on replaced token detection (RTD) task is more efficient than MLM, and thus achieved state-of-the-art results on English benchmarks. RTD is a pre-training task where a model is tasked to distinguish true input tokens from synthetically generated ones. RTD solves the issue of the mismatch created in MLM, where the model only sees the `[MASK]` token during pre-training but not during fine-tuning. In ELECTRA, a small masked language generator network **G** is used to generate used to generate the corrupted tokens, and BERT-based discriminator model **D** predicts for whether a token is an original or a replacement.

Current state-of-the-art language representation models for Arabic employ MLM as a pre-training objective (Antoun et al., 2020; Safaya et al., 2020; Lan et al., 2020; Abdul-Mageed et al., 2020b; Chowdhury et al., 2020; Abdul-Mageed et al., 2020a). In this paper, we describe the process of pre-training a transformer encoder model for Arabic language understanding using the RTD objective, which we call AraELECTRA. We also evaluate AraELECTRA on multiple Arabic NLP tasks and show empirically that AraELECTRA outperforms current state-of-the-art Arabic pre-trained models.

Our contributions can be summarized as follows:

- Pre-training the ELECTRA model on a large-scale Arabic corpus.
- Reaching a new state-of-the-art on multiple Arabic NLP tasks.
- Publicly releasing AraELECTRA on popular NLP libraries.

The rest of the paper is organized as follows. Section 2 provides a review of previous Arabic language representation literature. Section 3 details



*Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195
Kyiv, Ukraine (Virtual), April 19, 2021.

the methodology used in developing AraELECTRA. Section 4 describes the experimental setup, evaluation procedures, and experiment results. Finally, we conclude in Section 5.

## 2 Related Works

Recently, work on Arabic language representation have been on the rise due to the performance benefits that transfer learning approaches have brought. Early transfer learning approaches in Arabic relied on using pre-trained word embeddings i.e. AraVec (Soliman et al., 2017). Model-level transfer learning was shown to work on Arabic with hULMonA (ElJundi et al., 2019), a recurrent neural network-based language modeling approach. Antoun et al. (2020) and Safaya et al. (2020) improved on hULMonA, and pre-trained transformer-based models with MLM with large scale Arabic corpora. Other approaches addressed issues with the early BERT-based models such as training on code-switched English-Arabic corpora to improve performance on information retrieval tasks (Lan et al., 2020), and training on dialectal Arabic (DA) corpora to address the domain miss-match between MSA and DA during pre-training and fine-tuning (Abdul-Mageed et al., 2020b; Chowdhury et al., 2020).

We hence propose an Arabic ELECTRA-based language representation model pre-trained using the RTD objective on large MSA corpora.

## 3 AraELECTRA: Methodology

In this paper, we develop an ELECTRA-based Arabic language representation model to improve the state-of-the-art in Arabic reading comprehension. We create AraELECTRA a bidirectional transformer encoder model with 12 encoder layers, 12 attention heads, 768 hidden size, and 512 maximum input sequence length for a total of 136M parameters. The pre-training setup and dataset of AraELECTRA are described in the following sections.

### 3.1 Pre-training Setup

While AraBERT was trained using the MLM objective, AraELECTRA is pre-trained using the RTD objective. The RTD approach trains two neural network models, a generator **G** and a discriminator **D** or AraELECTRA, as shown in Figure 1. **G** takes a corrupted input sequence, where random tokens are replaced with the `[MASK]` token, and learns to predict the original tokens that have been masked. The generator network **G** is in our case a small BERT model with 12 encoder layers, 4 attention heads, and 256 hidden size[1]. The discriminator network **D** then takes as input the recovered sequence from the output of **G** and tries to predict which tokens were replaced and which tokens are from the original text.

While this approach may look similar to a generative adversarial network (GAN) (Goodfellow et al., 2014), the generator network in ELECTRA is trained with maximum-likelihood instead of adversarial training to fool the discriminator and the input to the generator is not a random noise vector, but a corrupted sequence of tokens.

### 3.2 pre-training Dataset

We chose to pre-train on the same dataset as AraBERTv0.2 (Antoun et al., 2020), to make the comparison between models fair. The dataset is a collection of the Arabic corpora list below:

- The OSCAR corpus (Ortiz Suárez et al., 2020).
- The 1.5B words Arabic Corpus (El-Khair, 2016).
- The Arabic Wikipedia dump from September 2020.
- The OSIAN corpus (Zeroual et al., 2019).
- News articles provided by As-Safir newspaper.

The total size of the training dataset is 77GB or 8.8 billion words, and comprises mostly news articles. For validation, we use new Wikipedia articles that were published after the September 2020 dump.

The same wordpiece vocabulary from AraBERTv0.2 was used for tokenization.

### 3.3 Fine-tuning

Since the discriminator network has the same architecture and layers as a BERT model, we add a linear classification layer on top of ELECTRA's output, and fine-tune the whole model with the added layer on new tasks. AraELECTRA's performance is validated on three Arabic NLP tasks i.e. question answering (QA), sentiment analysis (SA) and named-entity recognition (NER).

[1] In the generator, the input embeddings of size 768 are first projected into the generator hidden size with the addition of a linear layer.

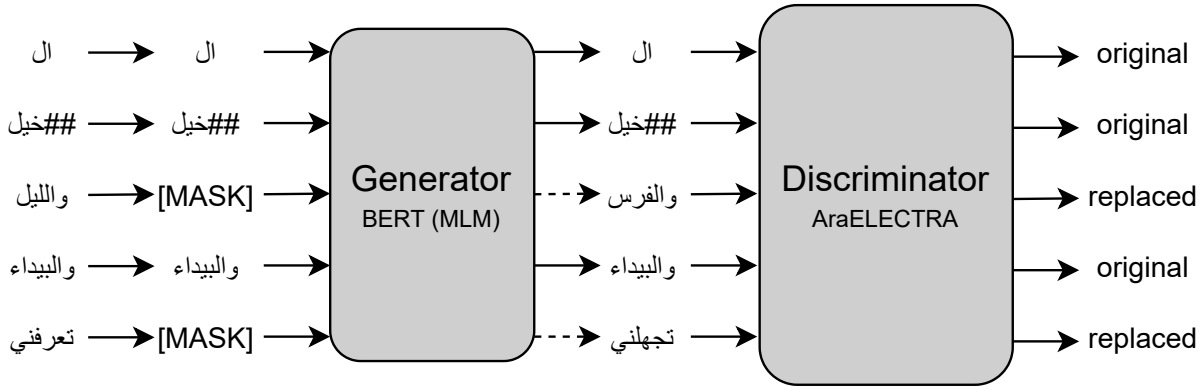


Figure 1: Replaced Token Detection pre-training task

## 4 Experiments and Evaluation

### 4.1 Experimental Setup

**pre-training** For pre-training, 15% of the 512 input tokens were masked. The model was pre-trained for 2 million steps with a batch size of 256. Pre-training took 24 days to finish on a TPUv3-8 slice. The learning rate was set to 2e-4, with 10000 warm-up steps.

**Fine-tuning** All the models were fine-tuned with batch size set to 32, maximum sequence length of 384, and a stride of 128 for QA, and a maximum sequence length of 256 for SA and NER. Experiments were only performed with the following learning rates [2e-5, 3e-5, 5e-5], since model specific hyper-parameter optimization is computationally expensive.

### 4.2 Datasets and other Models

#### 4.2.1 Question Answering

The question answering task examines the model's reading comprehension and language understanding capabilities. The datasets of choice are the Arabic Reading Comprehension Dataset (ARCD) (Mozannar et al., 2019) and the Typologically Diverse Question Answering dataset (TyDiQA) (Clark et al., 2020a). Both datasets follow the SQuAD (Rajpurkar et al., 2016) format where the model is required to extract the span of the answer, given a question and a context.

The ARCD (Mozannar et al., 2019) training set consists of 48344 machine-translated questions and answers from English, with 693 questions and answers from the ARCD set. The test was performed on the remaining 702 questions from the ARCD set. From the TyDiQA (Clark et al., 2020a), we chose the Arabic examples from the training and development sets of subtask 2, for a total of 14508 pairs for training and 921 pairs for testing.

#### 4.2.2 Sentiment Analysis

Arabic sentiment Analysis evaluation is done on the Arabic Sentiment Twitter Dataset for LEVantine (ArSenTD-Lev) (Baly et al., 2018). The dataset contains 4000 tweets written in the Levantine Arabic dialect and annotated for the sentiment (5 classes), topic, and sentiment target. The data was split 80-20 for training and testing.

#### 4.2.3 Named-Entity Recognition

For Arabic NER recognition, the model is evaluated on the ANERcorp dataset (Benajiba et al., 2007), with the data split from CAMeL Lab (Obeid et al., 2020). The train split has 125,102 words and the test split has 25,008 words, labeled for organization (ORG), person (PER), location (LOC), and miscellaneous (MISC).

#### 4.2.4 Reference Models

We evaluate our model against a collection of Arabic transformer models.

- AraBERTv0.1 (Antoun et al., 2020).
- AraBERTv0.2 base, large (Antoun et al., 2020).
- Arabic-BERT base, medium, large (Safaya et al., 2020).
- Arabic ALBERT base, large, xlarge[2].
- ARBERT (Abdul-Mageed et al., 2020a).

### 4.3 Results

Experimental results for the different datasets and models are shown in Table 1.

[2]https://github.com/KUIS-AI-Lab/Arabic-ALBERT/

| Model | TyDiQA | | ARCD | | ArSenTD-LEV | ANERcorp |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | F1 | F1 |
| AraBERTv0.1 | 68.51 | 82.86 | 31.62 | 67.45 | 53.56 | 83.14 |
| AraBERTv0.2-base | 73.07 | 85.41 | 32.76 | 66.53 | 55.71 | 83.70 |
| AraBERTv0.2-large | 73.72 | 86.03 | 36.89 | **71.32** | 56.94 | 83.08 |
| Arabic-BERT-base | 67.42 | 81.24 | 30.48 | 62.24 | 54.21 | 81.05 |
| Arabic-BERT-large | 70.03 | 84.12 | 33.33 | 67.28 | 55.32 | 82.15 |
| Arabic-ALBERT-base | 67.10 | 80.98 | 30.91 | 61.33 | 51.70 | 76.89 |
| Arabic-ALBERT-large | 68.07 | 81.59 | 34.19 | 65.41 | 54.62 | 79.61 |
| Arabic-ALBERT-xlarge | 71.12 | 84.59 | **37.75** | 68.03 | 54.15 | 81.13 |
| ARBERT | 71.55 | 83.69 | 31.62 | 65.93 | 53.52 | 83.33 |
| AraELECTRA | **74.91** | **86.68** | 37.03 | 71.22 | **57.20** | **83.95** |

Table 1: Performance of all tested model on the various Arabic downstream tasks. Overall best scores are highlighted in bold, while the best score within base-sized models is underlined.

The results show that AraELECTRA achieved the highest performance on all tested datasets when compared to the other base models, and only fell short on ARCD to Arabic-ALBERT-xlarge, a model 4 times its size, in exact match scores, and to AraBERTv0.2-large in F1-score.

The performance difference between both QA datasets is due to the poor quality of the ARCD training examples, which are translated from English SQuAD. ARCD training examples also contained text in languages other than Arabic and English, which further reduced performance due to the occurrence of unknowns subwords and characters. It is also to be noted, that some training examples in Arabic TyDiQA contained HTML artifacts which appeared in the training context and answer.

As for the ArSenTD-LEV scores, all test Arabic models still struggle with fine-grained labelling of ArSenTD-Lev. Mainly because the dataset only contains 4K examples distributed between 5 sentiment classes and on 6 diverse topics, with high class-imbalance.

These results clearly demonstrate that ELECTRA's RTD objective achieves higher performance especially on QA tasks and improved semantic representation compared to MLM on Arabic text.

## 5 Conclusion

In this paper, we showed that pre-training using the RTD objective on Arabic text is more efficient and produces pre-trained language representation models better than the MLM objective. Our AraELECTRA model improves the state-of-the-art for Arabic Question Answering, sentiment analysis and named-entity recognition, and achieves higher performance compared to other models pre-trained with the same dataset and with larger model sizes. Our model will be publicly available, along with our pre-training and fine-tuning code, in our repository github.com/aub-mind/arabert/tree/master/araelectra

## Acknowledgments

The author would like to thank Tarek Naous for the constructive criticism of the manuscript. This research was supported by the University Research Board (URB) at the American University of Beirut (AUB), and by the TFRC program, which we thank for the free access to cloud TPUs. We also thank As-Safir newspaper for the data access.

## References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020a. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, and Lyle Ungar. 2020b. Toward micro-dialect identification in diaglossic and code-switched environments. *arXiv preprint arXiv:2010.04900*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Ramy Baly, Alaa Khaddaj, Hazem Hajj, Wassim El-Hajj, and Khaled Bashir Shaban. 2018. Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets. In *OSACT 3: The 3rd*

*Workshop on Open-Source Arabic Corpora and Processing Tools*, page 37.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Shammur Absar Chowdhury, Ahmed Abdelali, Kareem Darwish, Jung Soon-Gyo, Joni Salminen, and Bernard J. Jansen. 2020. Improving Arabic text categorization using transformer training diversification. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 226–236, Barcelona, Spain (Online). Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020a. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*.

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

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.

Obeida ElJundi, Wissam Antoun, Nour El Droubi, Hazem Hajj, Wassim El-Hajj, and Khaled Shaban. 2019. hulmona: The universal language model in arabic. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 68–77.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. Gigabert: Zero-shot transfer learning from english to arabic. In *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural Arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy. Association for Computational Linguistics.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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 Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

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.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. Osian: Open source international arabic news corpus-preparation and integration into the clarin-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182.