AraBERT: Transformer-based Model for Arabic Language Understanding

Wissam Antoun*, Fady Baly*, Hazem Hajj

American University of Beirut

{wfa07, fgb06, hh63}@aub.edu.lb

Abstract

The Arabic language is a morphologically rich language with relatively few resources and a less explored syntax compared to English. Given these limitations, Arabic Natural Language Processing (NLP) tasks like Sentiment Analysis (SA), Named Entity Recognition (NER), and Question Answering (QA), have proven to be very challenging to tackle. Recently, with the surge of transformers based models, language-specific BERT based models have proven to be very efficient at language understanding, provided they are pre-trained on a very large corpus. Such models were able to set new standards and achieve state-of-the-art results for most NLP tasks. In this paper, we pre-trained BERT specifically for the Arabic language in the pursuit of achieving the same success that BERT did for the English language. The performance of AraBERT is compared to multilingual BERT from Google and other state-of-the-art approaches. The results showed that the newly developed AraBERT achieved state-of-the-art performance on most tested Arabic NLP tasks. The pretrained araBERT models are publicly available on github.com/aub-mind/araBERT hoping to encourage research and applications for Arabic NLP.

Keywords: Arabic, transformers, BERT, AraBERT, Language Models

1. Introduction

Pretrained contextualized text representation models have enabled massive advances in Natural Language Understanding (NLU) tasks, and achieved state-of-the-art performances in multiple NLP tasks (Howard and Ruder, 2018; Devlin et al., 2018). Early pretrained text representation models aimed at representing words by capturing their distributed syntactic and semantic properties using techniques like Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). However, these models did not incorporate the context in which a word appears into its embedding. This issue was addressed by generating contextualized representations using models like ELMO (Peters et al., 2018)).

Recently, there has been a focus on applying transfer learning by fine-tuning large pretrained language models for downstream NLP/NLU tasks with a relatively small number of examples, resulting in notable performance improvement for these tasks. This approach takes advantage of the language models that had been pre-trained in an unsupervised manner (or sometimes called self-supervised). However, this advantage comes with drawbacks, particularly the huge corpora needed for pre-training, in addition to the high computational cost of days needed for training (latest models required 500+ TPUs or GPUs running for weeks (Conneau et al., 2019; Raffel et al., 2019; Adiwardana et al., 2020)). These drawbacks restricted the availability of such models to English mainly and a handful of other languages. To remedy this gap, multilingual models have been trained to learn representations for +100 languages simultaneously, but still fall behind single-language models due to little data representation and small language-specific vocabulary. While languages with similar structure and vocabulary can benefit from the shared representations (Conneau et al., 2019), this is not the case for other languages, like Arabic, which differ in morphological and syntactic structure and share very little with other abundant Latin-based languages.

In this paper, we describe the process of pretraining the BERT transformer model (Devlin et al., 2018) for the Arabic language, and which we name ARABERT. We evaluate ARABERT on three Arabic NLU downstream tasks that are different in nature: (*i*) Sentiment Analysis (SA), (*ii*) Named Entity Recognition (NER), and (*iii*) Question Answering (QA). The experiments results show that ARABERT achieves state-of-the-art performances on most datasets, compared to several baselines including previous multilingual and single-language approaches. The datasets that we considered for the downstream tasks contained both Modern Standard Arabic (MSA) and Dialectal Arabic (DA).

Our contributions can be summarized as follows:

- A methodology to pretrain the BERT model on a large-scale Arabic corpus.

- Application of ARABERT to three NLU downstream tasks: Sentiment Analysis, Named Entity Recognition and Question Answering.

- Publicly releasing ARABERT on popular NLP libraries.

The rest of the paper is structured as follows. Section 2. provides a concise literature review of previous work on language representation for English and Arabic. Section 3. describes the methodology that was used to develop ARABERT. Section 4. describes the downstream tasks and benchmark datasets that are used for evaluation. Section 5. presents the experimental setup and discusses the results. Finally, section 6. concludes and points to possible directions for future work.

---

*Equal Contribution

2. Related Works

2.1. Evolution of Word Embeddings

The first meaningful representations for words started with the word2vec model developed by (Mikolov et al., 2013). Since then, research started moving towards variations of word2vec like of GloVe (Pennington et al., 2014) and fast-Text (Mikolov et al., 2017). While major advances were achieved with these early models, they still lacked contextualized information, which was tackled by ELMO (Peters et al., 2018). The performance over different tasks improved noticeably, leading to larger structures that had superior word and sentence representations. Ever since, more language understanding models have been developed such as ULMFit (Howard and Ruder, 2018), BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), ALBERT (Lan et al., 2019), and T5 (Raffel et al., 2019), which offered improved performance by exploring different pretraining methods, modified model architectures and larger training corpora.

2.2. Non-contextual Representations for Arabic

Following the success of the English word2vec (Mikolov et al., 2013), the same feat was sought by NLP researchers to create language specific embeddings. Arabic word2vec was first attempted by (Soliman et al., 2017), and then followed by a Fasttext model (Bojanowski et al., 2017) trained on Wikipedia data and showing better performance than word2vec. To tackle dialectal variations in Arabic (Erdmann et al., 2018) presented techniques for training multidialectal word embeddings on relatively small and noisy corpora, while (Abu Farha and Magdy, 2019; Abdul-Mageed et al., 2018) provided Arabic word embeddings trained on ∼250M tweets.

2.3. Contextualized Representations for Arabic

For non-English languages, Google released a multilingual BERT (Devlin et al., 2018) supporting 100+ languages with solid performance for most languages. However, pre-training monolingual BERT for non-English languages proved to provide better performance than the multilingual BERT such as Italian BERT Alberto (Polignano et al., 2019) and other publicly available BERTs (Martin et al., 2019; de Vries et al., 2019). Arabic specific contextualized representations models, such as hULMonA (ElJundi et al., 2019), used the ULMfit structure, which had a lower performance that BERT on English NLP Tasks.

3. ARABERT: Methodology

In this paper, we develop an Arabic language representation model to improve the state-of-the-art in several Arabic NLU tasks. We create ARABERT based on the BERT model, a stacked Bidirectional Transformer Encoder (Devlin et al., 2018). This model is widely considered as the basis for most state-of-the-art results in different NLP tasks in several languages. We use the BERT-base configuration that has 12 encoder blocks, 768 hidden dimensions, 12 attention heads, 512 maximum sequence length, and a total of ∼110M parameters[1]. We also introduced additional preprocessing prior to the model's pre-training, in order to better fit the Arabic language. Below, we describe the pre-training setup, the pre-training dataset for ARABERT, the proposed Arabic-specific preprocessing, and the fine-tuning process.

3.1. Pre-training Setup

Following the original BERT pre-training objective, we employ the *Masked Language Modeling* (MLM) task by adding whole-word masking where; 15% of the N input tokens are selected for replacement. Those tokens are replaced 80% of the times with the [MASK] token, 10% with a random token, and 10% with the original token. Whole-word masking improves the pre-training task by forcing the model to predict the whole word instead of getting hints from parts of the word. We also employ the *Next Sentence Prediction* (NSP) task that helps the model understand the relationship between two sentences, which can be useful for many language understanding tasks such as Question Answering.

3.2. Pre-training Dataset

The original BERT was trained on 3.3B words extracted from English Wikipedia and the Book Corpus (Zhu et al., 2015). Since the Arabic Wikipedia Dumps are small compared to the English ones, we manually scraped Arabic news websites for articles. In addition, we used two publicly available large Arabic corpora: (1) the 1.5 billion words Arabic Corpus (El-Khair, 2016), which is a contemporary corpus that includes more than 5 million articles extracted from ten major news sources covering 8 countries, and (2) OSIAN: the Open Source International Arabic News Corpus (Zeroual et al., 2019) that consists of 3.5 million articles (∼1B tokens) from 31 news sources in 24 Arab countries.

The final size of the pre-training dataset, after removing duplicate sentences, is 70 million sentences, corresponding to ∼24GB of text. This dataset covers news from different media in different Arab regions, and therefore can be representative of a wide range of topics discussed in the Arab world. It is worth mentioning that we preserved words that include Latin characters, since it is common to mention named entities, scientific or technical terms in their original language, to avoid information loss.

3.3. Sub-Word Units Segmentation

Arabic language is known for its lexical sparsity which is due to the complex concatenative system of Arabic (Al-Sallab et al., 2017). Words can have different forms and share the same meaning. For instance, while the definite article "‏ال‎ - *Al*", which is equivalent to "the" in English, is always prefixed to other words, it is not an intrinsic part of that word. Hence, when using a BERT-compatible tokenization, tokens will appear twice, once with "*Al-*" and

---

[1]Further details about the transformer architecture can be found in (Vaswani et al., 2017)

once without it. For instance, both "كتاب - *kitAb*" and "الكتاب-*AlkitAb*" need to be included in the vocabulary, leading to a significant amount of unnecessary redundancy.

To avoid this issue, we first segment the words using Farasa (Abdelali et al., 2016) into stems, prefixes and suffixes. For instance, "اللّغة - *Alloga*" becomes "ال + لغ + ة - *Al+ log +a*". Then, we trained a SentencePiece (an unsupervised text tokenizer and detokenizer (Kudo, 2018)), in unigram mode, on the segmented pre-training dataset to produce a subword vocabulary of ∼60K tokens. To evaluate the impact of the proposed tokenization, we also trained SentencePiece on non-segmented text to create a second version of ARABERT (AraBERTv0.1) that does not require any segmentation. The final size of vocabulary was 64k tokens, which included nearly 4K unused tokens to allow further pre-training, if needed.

3.4. Fine-tuning

Sequence Classification To fine-tune AraBERT for sequence classification, we take the final hidden state of the first token, which corresponds to the word embedding of the special "[CLS]" token prepended to the start of each sentence. We then add a simple feed-forward layer with standard Softmax to get the probability distribution over the predicted output classes. During fine-tuning, the classifier and the pre-trained model weights are trained jointly to maximize the log-probability of the correct class.

Named Entity Recognition For the NER task, each token in the sentence is labeled with the IOB2 format (Ratnaparkhi, 1998), where the "B" tag corresponds to the first word of the entity, the "I" tag corresponds to the rest of the words of the same entity, and the "O" tag indicates that the tagged word is not a desired named entity. Hence, we treat the system as a multi-class classification process, which allows us to use some text classification methods to label the tokens. Furthermore, after using the AraBERT tokenizer, we only input the first sub-token of each word to the model.

Question Answering In the QA, given a question and a passage containing the answer, the model needs to select a span of text that contains the answers. This is done by predicting a "start" token and an "end" token on condition that the "end" token should appear after the "start" token. During training, the final embedding of every token in the passage is fed into two classifiers, each with a single set of weights, which are applied to every token. The dot product of the output embeddings and the classifier is then fed into a softmax layer to produce a probability distribution over all the tokens. The token with the highest probability of being a "start" toke is then selected, and the same process is repeated for the "end" token.

4. Evaluation

We evaluated ARABERT on three Arabic language understanding downstream tasks: Sentiment Analysis, Named Entity Recognition, and Question Answering. As a baseline, we compared ARABERT to the multilingual version of BERT, and to other state-of-art results on each task.

4.1. Sentiment Analysis

We evaluated ARABERT on the following Arabic sentiment datasets that cover different genres, domains and dialects.

- **HARD:** The Hotel Arabic Reviews Dataset (Elnagar et al., 2018) contains 93,700 hotel reviews written in both Modern Standard Arabic (MSA) and in dialectal Arabic. Reviews are split into positive and negative reviews, where a negative review has a rating of 1 or 2, a positive review has a rating of 4 or 5, and neutral reviews with rating of 3 were ignored.

- **ASTD:** The Arabic Sentiment Twitter Dataset (Nabil et al., 2015) contains 10,000 tweets written in both MSA and Egyptian dialect. We tested on the balanced version of the dataset, referred to as ASTD-B.

- **ArSenTD-Lev:** The Arabic Sentiment Twitter Dataset for LEVantine (Baly et al., 2018) contains 4,000 tweets written in Levantine dialect with annotations for sentiment, topic and sentiment target. This is a challenging dataset as the collected tweets are from multiple domains and discuss different topics.

- **LABR:** The Large-scale Arabic Book Reviews dataset (Aly and Atiya, 2013) contains 63,000 book reviews written in Arabic. The reviews are rated between 1 and 5. We benchmarked our model on the unbalanced two-class dataset, where reviews with ratings of 1 or 2 are considered negative, while those with ratings of 4 or 5 are considered positive.

- **AJGT:** The Arabic Jordanian General Tweets dataset (Alomari et al., 2017) contains 1,800 tweets written in Jordanian dialect. The tweets were manually annotated as either positive or negative.

Baselines: Sentiment Analysis is a popular Arabic NLP task. Previous approaches relied on sentiment lexicons such as ArSenL (Badaro et al., 2014), which is a large-scale lexicon of MSA words that is developed using the Arabic WordNet in combination with the English SentiWordNet. Recurrent and recursive neural networks were explored with different choices of Arabic-specific processing (Al Sallab et al., 2015; Al-Sallab et al., 2017; Baly et al., 2017). Convolutional Neural Networks (CNN) were trained with pre-trained word embeddings (Dahou et al., 2019a). A hybrid model was proposed by (Abu Farha and Magdy, 2019), where CNNs were used for feature extraction, and LSTMs were used for sequence and context understanding.

Current state-of-the-art results are achieved by the hUL-MonA model (ElJundi et al., 2019), which is an Arabic language model that is based on the ULMfit architecture (Howard and Ruder, 2018). We compare the results of ARABERT to those of hULMonA.

4.2. Named Entity Recognition

This task aims to extract and detect named entities in the text. It is framed as a word-level classification (or tagging) task, where the classes correspond to pre-defined categories such as names, locations, organizations, events and time expressions. For evaluation, we use the Arabic NER corpus (ANERcorp) (Benajiba and Rosso, 2007). This dataset contains 16.5K entity mentions distributed among 4 entities categories, *person* (39%), *organization*: (30.4%), *location*: (20.6%), and *miscellaneous*: (10%).

Baselines: Advances in the NER task have been focusing on English, namely on the CoNLL 2003 (Sang and De Meulder, 2003) dataset. Initially, NER was tackled with Conditional Random Fields (CRF) (Lafferty et al., 2001). Later on, CRFs were used on top of Bi-LSTM models (Huang et al., 2015; Lample et al., 2016) presenting significant improvements over standalone CRFs. Bi-LSTM-CRF structures were then used with contextualized embeddings that displayed further improvements (Peters et al., 2018). Lastly, large pre-trained transformers showed slight improvement, setting the current state-of-the-art performance (Devlin et al., 2018). As for Arabic, We compare ARABERT performance with Bi-LSTM-CRF baseline that set the previous state-of-the-art performance (El Bazi and Laachfoubi, 2019), and with BERT multilingual.

4.3. Question Answering

Open-domain Question Answering (QA) is one of the goals of artificial intelligence, this goal can be achieved by leveraging natural language understanding and knowledge gathering (Kwiatkowski et al., 2019). English QA research has been fueled by the release of large datasets such as Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016). On the other hand, research in Arabic QA has been hindered by the lack of such massive datasets, and by the fact that Arabic presents its own challenges such as:

- Inconsistent name spelling (ex: Syria in Arabic can be written as "سوريا - *sOriyA*" and "سورية - *sOriyT*")

- Name de-spacing (ex: The name is written as "عبدالعزيز - *AbdulAzIz*" in the question, and "عبد العزيز - *Abdul AzIz*" in the answer)

- Dual form "المثنى", which can have multiple forms (ex:

"قلمان" - "*qalamAn*" or "قلمين" - "*qalamyn*" meaning "*two pencils*")

- Grammatical gender variation: all nouns, animate and inanimate objects are classified under two genders either masculine or feminine (ex: "كبير" - "*kabIr*" and "كبيرة" - "*kabIrT*"

We evaluate ARABERT on the Arabic Reading Comprehension Dataset (ARCD) (Mozannar et al., 2019) , where the task is to find the span of the answer in a document for a given question. ARCD contains 1395 questions on Wikipedia articles along with 2966 machine translated questions and answers from the SQuAD dubbed (Arabic-SQuAD). We train on the whole Arabic-SQuAD and on 50% of ARCD and test on the remaining 50% of ARCD.

Baselines Multilingual BERT had previously achieved state of the art results on ARCD.

5. Experiments

5.1. Experimental Setup

Pretraining In our experiments, the original implementation of BERT on TensorFlow was used. The data for pre-training was sharded, transformed into TFRecords, and then stored on Google Cloud Storage. Duplication factor was set to 10, a random seed of 34, and a masking probability of 15%. The model was pre-trained on a TPUv2-8 pod for 1,250,000 steps. To speed up the training time, the first 900K steps were trained on sequences of 128 tokens, and the remaining steps were trained on sequences of 512 tokens. The decision of stopping the pre-training was based on the performance of downstream tasks. We follow the same approach taken by the open-sourced German BERT (DeepsetAI,). Adam optimizer was used, with a learning rate of 1e-4, batch size of 512 and 128 for sequence length of 128 and 512 respectively. Training took 4 days, for 27 epochs over all the tokens.

Fine-tuning Fine-tuning was done independently using the same configuration for all tasks. We do not run extensive grid search for the best hyper-parameters due to computational and time constraints. We use the splits provided by the dataset's authors when available. and the standard 80% and 20% when not[2].

5.2. Results

Table 1 illustrates the experimental results of applying AraBERT to multiple Arabic NLU downstream tasks, compared to state-of-the-art results and the multilingual BERT model (mBERT).

---

[2]The scripts used to create the datasets are available on our Github repo https://github.com/aub-mind/arabert

Sentiment Analysis For Arabic sentiment analysis, the results in Table 1 show that both versions of AraBERT outperform mBERT and other state-of-the-art approaches on most tested datasets. Even though AraBERT was trained on MSA, the model was able to preform well on dialects that were never seen before.

Table 1: Performance of AraBERT on Arabic downstream tasks compared to mBERT and previous state of the art systems

Task	metric	prev. SOTA	mBERT	AraBERTv0.1/ v1
SA (HARD)	Acc.	95.7*	95.7	**96.2** / 96.1
SA (ASTD)	Acc.	86.5*	80.1	92.2 / **92.6**
SA (ArsenTD-Lev)	Acc.	52.4*	51.0	58.9 / **59.4**
SA (AJGT)	Acc.	92.6**	83.6	93.1 / **93.8**
SA (LABR)	Acc.	87.5†	83.0	85.9 / 86.7
NER (ANERcorp)	macro-F1	81.7††	78.4	**84.2** / 81.9
QA (ARCD)	Exact Match		34.2	51.1 / **54.8**
	macro-F1	mBERT	61.3	82.1 / **82.2**
	Sent. Match		90.0	95.5 / **95.6**

* (ElJundi et al., 2019)
** (Dahou et al., 2019b)
† (Dahou et al., 2019b)
†† Previous state of the art performance by BiLSTM-CRF model

Named Entity Recognition Results in Table 1 show that AraBERTv0.1 improved results by 2.53 points in F1 score scoring 84.2 compared with the Bi-LSTM-CRF model, making AraBERT the new state-of-the-art for NER on ANERcorp. Testing AraBERT with tokenized suffixes and prefixes showed results similar to that of the Bi-LSTM-CRF model. We believe that the reason this happened is that the start token (B-label) is referenced to the suffixes most of the time. An example of this, "الجامعة" with a label B-ORG becomes "جامعة", "ال" with labels B-ORG, I-ORG respectively, providing misleading starting cues to the model. Testing multilingual BERT, it proved inefficient as we got results lower than the baseline model.

Question Answering While the results in Table 1 show an improvement in F1-score, the exact match scores were significantly lower. Upon further examination of the results, the majority of the erroneous answers differed from the true answer by one or two words with no significant impact on the semantics of the answer. Examples are shown in Tables 2 and 3. We also report a 2% absolute increase in the sentence match score over mBERT, which is the previous state-of-the-art. Sentence Match (SM) measures the percentage of predictions that are within the same sentence as the ground truth answer.

Table 2: Example of an erroneous results from the ARCD test set: the only difference is the preposition "في - *In*".

Question	أين تأسست منظمة الأمم المتحدة؟
	where was the united nations established?
Ground Truth	*In San Francisco* – في سان فرنسيسكو
Predicted Answer	*San Francisco* – سان فرنسيسكو

Table 3: Another example of an erroneous results from the ARCD test set: the predicted answer does not include "introductory" words.

Question	ما هو النظام الخاص بدولة النمسا؟
	What is the type of government in Austria?
Ground Truth	*Austria is a federal republic* – النمسا هي جمهورية فيدرالية
Predicted Answer	*A federal republic* – جمهورية فيدرالية

5.3. Discussion

AraBERT achieved state-of-the-art performance on sentiment analysis, named entity recognition, and the question answering tasks. This adds truth to the assumption that pretrained language models on a single language only surpass the performance of a multilingual model. This jump in performance has many explanations. First, data size is a clear factor for the boost in performance. AraBERT used around 24GB of data in comparison with the 4.3G Wikipedia used for the multilingual BERT. Second, the vocab size used in the multilingual BERT is 2k tokens in comparison with 64k vocab size used for developing AraBERT. Third, with the large data size, the pre-training distribution has more diversity. As for the fourth point, the pre-segmentation applied before BERT tokenization improved performance on SA and QA tasks but reduced it on the NER task. It is also noted that the pre-processing applied to the pre-training data took into consideration the complexities of the Arabic language. Hence, increased the effective vocabulary by excluding unnecessary redundant tokens that come with certain common prefixes, and help the model learn better by reducing the language complexity. We believe these factors helped to reach state-of-the-art results on 3 different tasks and 8 different datasets. Obtained results indicate that the advantage we got in the datasets considered are better understood in a monolingual model than of a general language model trained on Wikipedia crawls such as multilingual BERT.

6. Conclusion

AraBERT sets a new state-of-the-art for several downstream tasks for Arabic language. It is also 300MB smaller than multilingual BERT. By publicly releasing our AraBERT models, we hope that it will be used to serve as the new baseline for the various Arabic NLP tasks, and hope that this work will act as a footing stone to building and improving future Arabic language understanding models. We are currently working on publishing an AraBERT version that won't depend on external tokenizers. We are also in the process of training models with a better understanding of the various dialects that the Arabic language has across different Arabic countries.

7. Acknowledgments

8. References

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16.

Abdul-Mageed, M., Alhuzali, H., and Elaraby, M. (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)*.

Abu Farha, I. and Magdy, W. (2019). Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy, August. Association for Computational Linguistics.

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., and Le, Q. V. (2020). Towards a human-like open-domain chatbot.

Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El-Hajj, W., and Shaban, K. (2015). Deep learning models for sentiment analysis in arabic. In *Proceedings of the second workshop on Arabic natural language processing*, pages 9–17.

Al-Sallab, A., Baly, R., Hajj, H., Shaban, K. B., El-Hajj, W., and Badaro, G. (2017). Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):1–20.

Alomari, K. M., ElSherif, H. M., and Shaalan, K. (2017). Arabic tweets sentimental analysis using machine learning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 602–610. Springer.

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)*, pages 494–498, Sofia, Bulgaria, August. Association for Computational Linguistics.

Badaro, G., Baly, R., Hajj, H., Habash, N., and El-Hajj, W. (2014). A large scale arabic sentiment lexicon for arabic opinion mining. In *Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP)*, pages 165–173.

Baly, R., Hajj, H., Habash, N., Shaban, K. B., and El-Hajj, W. (2017). A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):1–21.

Baly, R., Khaddaj, A., Hajj, H., El-Hajj, W., and Shaban, K. B. (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.

Benajiba, Y. and Rosso, P. (2007). Anersys 2.0: Conquering the ner task for the arabic language by combining the maximum entropy with pos-tag information. In *IICAI*, pages 1814–1823.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale.

Dahou, A., Elaziz, M. A., Zhou, J., and Xiong, S. (2019a). Arabic sentiment classification using convolutional neural network and differential evolution algorithm. *Computational intelligence and neuroscience*, 2019.

Dahou, A., Xiong, S., Zhou, J., and Elaziz, M. A. (2019b). Multi-channel embedding convolutional neural network model for arabic sentiment classification. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(4):1–23.

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., and Nissim, M. (2019). Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

DeepsetAI.). Open sourcing german bert.

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*.

El Bazi, I. and Laachfoubi, N. (2019). Arabic named entity recognition using deep learning approach. *International Journal of Electrical & Computer Engineering (2088-8708)*, 9(3).

El-Khair, I. A. (2016). 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.

ElJundi, O., Antoun, W., El Droubi, N., Hajj, H., El-Hajj, W., and Shaban, K. (2019). hulmona: The universal language model in arabic. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 68–77.

Elnagar, A., Khalifa, Y. S., and Einea, A. (2018). Hotel arabic-reviews dataset construction for sentiment analysis applications. In *Intelligent Natural Language Processing: Trends and Applications*, pages 35–52. Springer.

Erdmann, A., Zalmout, N., and Habash, N. (2018). Addressing noise in multidialectal word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 558–565.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M.,

Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., and Petrov, S. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Lafferty, J. D., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., Éric Villemonte de la Clergerie, Seddah, D., and Sagot, B. (2019). Camembert: a tasty french language model.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2017). Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Mozannar, H., Maamary, E., El Hajal, K., and Hajj, H. (2019). Neural arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118.

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, Lisbon, Portugal, September. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Polignano, M., Basile, P., de Gemmis, M., Semeraro, G., and Basile, V. (2019). AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, volume 2481. CEUR.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ratnaparkhi, A. (1998). Maximum entropy models for natural language ambiguity resolution.

Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding.

Zeroual, I., Goldhahn, D., Eckart, T., and Lakhouaja, A. (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, Florence, Italy, August. Association for Computational Linguistics.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.