SCUoL at CheckThat! 2021: An AraBERT Model for Check-Worthiness of Arabic Tweets

Saud Althabiti [1,2], Mohammad Alsalka [1] and Eric Atwell [1]

[1] *University of Leeds, Leeds, LS2 9JT, United Kingdom*
[2] *King Abdulaziz University, Jeddah, 21589, Kingdom of Saudi Arabia*

Abstract

Many people nowadays tend to explore social media to obtain news and find information about various events and activities. However, an abundance of misleading and false information is spreading every day for many purposes, dramatically impacting societies. Therefore, it is vitally important to identify false information on social media to help individuals distinguish the truth and protect communities from the harmful effects of false information. For this reason, determining which information has the priority to be scrutinized is a significant prior step that several studies have considered. In this paper, we have addressed Subtask-1A(Arabic) of CLEF2021 CheckThat! Lab. We have done that in two steps. The first involved pre-processing the provided dataset with text segmentation and tokenization. In the second step, we implemented different models on the Arabic tweets in order to binary classify them according to whether a specific tweet is worth being considered for fact-checking or not. We mainly compared two versions of the pre-trained AraBERT model with some of the traditional word encoding methods, including the Linear SVC model with TF-IDF. The results indicate that the AraBERTv2 version outperforms the other models. Consequently, we used it for our final submission, and we were ranked third among eight other participating teams.

Keywords

AraBERTv2, AraBERTv0.2, Check-worthiness, Fact-check, CheckThat Lab

1. Introduction

Social media has become a key tool for disseminating news as it allows news providers to distribute and propagate their messages to a broader range of users. The colossal number of people using social media every day has accelerated the rate at which fake news spreads [1]. This has been a common problem that affected many people, governments, and organizations. Consequently, researchers have been working on developing various methods for detecting misinformation in various languages. Because of the large amount of false information on social media, it is impossible for human moderators to examine every single piece of information that may involve false facts. Therefore, before fact-checking by humans, there is a need to flag potential posts that may contain false information. Thus, organizations and researchers have started to develop systems that check for the most notable claims [1]. This paper concentrates on Subtask 1A in Arabic from CheckThat, a lab contest with different tasks for competitors [2]. This year, the lab provided the following three main tasks: Detecting Check-Worthy Claims (Task1), Previously Fact-Checked Claims (Task2), and Fake News Detection (Task3). Each of the main tasks has two subtasks. Subtask 1A was provided in five different languages (Arabic, Bulgarian, English, Spanish, and Turkish).

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania
EMAIL: scssal@leeds.ac.uk (S. Althabiti); M.A.Alsalka@leeds.ac.uk (M. Alsalka); e.s.atwell@leeds.ac.uk (E. Atwell)

1.1. Task Definition:

Subtask 1A in the Arabic language aims at classifying real-world tweets into predefined categories. Each tweet has been labelled as to whether the tweet is worth checking or not. Hence, this is a supervised text classification problem that aims to classify upcoming tweets based on their content. We used a pre-trained model that can make more accurate predictions. The main objective of this task is to develop a system that would automatically identify whether a tweet is worth checking or not.

This subtask comes with three CSV files containing collections of tweets and divided into the following three categories:

- A training dataset that includes 3439 labelled tweets.
- A development dataset with 661 labelled tweets.
- A testing dataset with 600 un-labelled tweets.

The training and development datasets contain four features (topic_id, tweet_id, tweet_url, tweet_text) and two labelling features (claim, claim_worthiness) . The second labelling feature classifies the tweet's check-worthiness, which is defined as whether this tweet includes any concerning claims that may have a significantly detrimental impact.

The remaining sections of this paper are as follow. The next section gives a review of related work. Section 3 details the adopted approach in our experiment, including the pre-processing phase of the provided text and the fine-tuning stage of the used pre-trained model. Section 4 provides the results of the experiments, and Section 5 provides the conclusions. Finally, Section 6 contain some acknowledgements for this work.

2. Related Work

This section briefly describes other work which we considered while developing our experiment.

2.1. Check-Worthy Claims Detection using ClaimRank System.

ClaimRank is a multilingual system developed for the purpose of detecting claims that are worth checking [3]. The system supports two languages: English and Arabic. It is trained using real annotations from nine reliable organizations, such as CNN and PolitiFact. During the development of the ClaimRank system, the data underwent pre-processing, followed by the extraction of features. These features are then passed to the model. For Arabic claim detection, they used two datasets translated into English when training the model. By contrast, our experiment used a pre-trained model with a large Arabic dataset.

2.2. Automatic Identification and Verification of Claims

Atanasova et al [2]did some work in CLEF'2019 CheckThat Lab. It was related to the same task (Task1- Check-worthy claims), which we present in our paper. They discussed the dataset, evaluation tools, and evaluated results for eleven teams participating in this task [2]. In terms of data, they used a previous version of the dataset called "CT-CWC-18" [4]. The training data in the 2019-version was a combination of both the training and testing English datasets from the previous year. In addition, they added labelled data from 16 different resources for testing purposes. In the evaluation phase, they used Mean Average Precision measurement to calculate the rank of sentences of each participant. Best results made by participating teams used supervised classification models. Linear machine learning models such as SVM, naive Bayes, regression trees and neural networks models such as LSTM were utilized [2]. Despite the relatively great results, none of them used pre-trained models.

2.3. Accenture at CheckThat! 2020

The winner team from the previous year contest CheckThat! Lab proposed a system that introduces a claims fact-checking approach employing two models: BERT and RoBERTa [5]. This team scored 1st position in both the Arabic and the English track. The models were used to classify social media claims that require an expert fact-checker. They examined four models and used AraBERTv0.1 Upsampled model for the official submission, which acquired a P@30 of 0.7000 compared to AraBERTv1.0 Upsampled, AraBERTv0.1 Unmodified, and ArabicBERT-Base Upsampled that achieved a P@30 of 0.675, 0.669, and 0.664 respectively [6].

2.4. AraBERT Model Study

One of the best models that delivered state-of-the-art outcomes in various NLP tasks is the AraBERT model [7]. This study used the BERT model for the Arabic language in order to accomplish what the English BERT model did. They discussed performing several tasks using this model as well as the mBERT, which is a multilingual BERT. The results showed that AraBERTv0.1 outperformed mBERT in sentiment analysis, named entity recognition, and question answering tasks [7].

3. Methodology

Various models can be applied for solving a binary classification problem. For example, linear machine learning models such as SVM have been used by many researchers and showed excellent results. Additionally, deep learning is considered one of the most effective domains in tackling such problems. It has demonstrated satisfactory performance in solving diverse tasks in natural language processing (NLP), such as text classification. However, the scarcity of big labelled textual datasets may lower the accuracy of the results. A transformer model such as BERT was introduced in the NLP domain to support various state-of-the-art performance tasks. Since we are dealing with Arabic tweets, we pre-trained a BERT-based model designed explicitly for Arabic called AraBERT. The following subsections cover text pre-processing and explain the model used in this experiment.

3.1. Text Pre-processing

In the first stage of this phase, all sentences were split using a pre-processing function called ArabertPreprocessor.preprocess. The main idea here is to divide words from punctuation symbols and numbers by a space, as in the following example:

#صفقة_القرن ← # صفقة _ القرن

3.1.1. Segmentation and Tokenization

Because Arabic has a complicated lexical structure and same-meaning vocabularies may have various forms, we considered pre-processing steps such as segmentation. To explain this, take the Arabic word "AlElm - العلم" which consists of the word "علم" (that means science) and the prefix "ال" (that means "the" in English). This word with and without the definite article can be duplicated and counted as two different words [7]. Therefore, all tweets, including the training, development, and testing datasets, were segmented before AraBERT tokenization.

There are a number of options for tokenization when working with a pre-trained BERT model. We used the base model 'aubmindlab/bert-base-arabertv2', which is more common in practice. Also, we utilized a helpful tool, called AutoTokenizer, which is essential for splitting inputs into tokens with each token encoded into a corresponding id. Then, we built a batch to feed the model using this tokenizer by setting the maximum length to 128 and the batch size to 32, padding inputs that include fewer tokens than the full length, and truncating longer inputs to the specified length.

Table 1: The output of the tokenization process with an example of a max_length of 16

صفقة _ القرن # القدس _ عاصمة _ فلسطين _ الأبدية #																
input_ids	2	10	5987	68	2890	10	3306	68	6552	68	2036	68	54632	3	0	0
token_type_ids	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
attention_mask	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
decoded text	**[CLS]**	#	صفقة	_	القرن	#	القدس	_	عاصمة	_	فلسطين	_	الأبدية	**[SEP]**		

The table above illustrates how the tokenizer function converts a given input to produce a dictionary. Due to the narrowed space of this table, we changed the max_length to 16 to exhibit this example. The keys of the returned dictionary are the strings: input_id, token_type_id, and attention_mask. Each token has a unique input id. For instance, the character hashtag (#) corresponds to the id 10. The attention_mask key determines what tokens should get noticed by the model. If the attention mask is zero, there is no input, and the padding argument fills the batch to complete the maximum length.

3.2. Fine Tuning

We used a pre-trained model called AraBERT as reported by Antoun et al. [7]. AraBERT is a model derived from the BERT model, which stands for Bidirectional Encoder Representations from Transformers [8]. The BERT model has been used in various Natural Language processing tasks and showed significant results [7].

In this experiment, we used BERT for sequence classification to fine-tune the model with the help of the transformer's library. The used classifier includes 12 BERT layers and fed into a hyperbolic tangent activation function to return a probability distribution across all the tokens. Training smaller datasets from scratch can cause overfitting. Therefore, it is good to use models previously trained on a vast dataset and fine-tune the model using a smaller dataset [7].

We tried setting a high number of epochs during the fine-tuning step to look out for overfitting and take appropriate measures. Each time we trained the model using the training and validation datasets, there was overfitting in the fourth epochs. Since there is no accurate way to determine the number of epochs, we decided to set it to 3. The table below demonstrates the average training loss and the accuracy for the validation run.

Table 2: The Training loss average and validation accuracy in the first three epochs.

	Epoch 1	Epoch 2	Epoch 3
Average training loss	0.32	0.19	0.12
Accuracy	0.81	0.83	0.84

4. Results and Discussion

Since AraBERT has many versions, we explored different practices to determine which version we should use for our submission. Although a study compared different AraBERT models and found that some larger models performed better than others in terms of accuracy [9], we only dealt with base models due to the limitation of the used hardware. We first used the validation datasets to determine which model we should use for the final submission. Then, we used this dataset for development and compared the predicted results to the gold-labels dataset.

4.1. Validation Datasets Results Analysis

We examined the results of both versions on the development set. The results show that using the AraBERTv2-base model may increase the accuracy and the weighted average of precision, recall, and f1-score. In another attempt, we executed a baseline model that can be used for classification, particularly the Linear Support Vector Classification model (SVC) with TF-IDF. The experiment of the validation datasets indicates that this traditional word encodings method yields a weighted average F1-score of 0.67 compared to the AraBERTv0.2 and AraBERTv2 with 0.83 and 0.84, respectively. These results suggest that these methods might be insufficient to model the complicated Arabic tweets.

4.2. Testing Datasets Results Analysis

After releasing the test gold labels dataset, we compared our predictions to the correct values. Table 3 shows a comparison between AraBERTv0.2-base and AraBERTv2-base for both labels using a classification report function. The main observation from Table 3 is that the two versions approximately performed the same. There is a slight improvement in general using AraBERTv2-base, which we decided to use for the official submission.

Table 3: Comparison between AraBERTv0.2-base and AraBERTv2-base results

	AraBERTv0.2-base		AraBERTv2-base	
Criterion	0	1	0	1
Precision	0.74	0.60	0.72	0.63
Recall	0.71	0.64	0.78	0.56
F1-score	0.72	0.62	0.75	0.59
Accuracy	0.68		0.69	

Finally, the submitted predictions for Subtask1A on Check-worthiness of tweets in Arabic has achieved a 0.612 Mean-Average-Precision, and our team SCUoL ranked third among eight participating teams. Although the results were encouraging, the main concern is the insufficient improvement when examining the two versions of the AraBERT model. Another contributing factor is contrasting base models with larger-size models to provide more comprehensive selections.

5. Conclusion and Future Work

It is essential to gain information from reliable sources. Therefore, a considerable number of models are used to solve this problem. Due to the massive number of tweets published online, moderators want to concentrate on tweets worth checking only. This experiment examined an Arabic dataset that is labelled as to whether a tweet is worth checking. We took two steps toward our goal, including pre-processing and fine-tuning the datasets. We first approach the task by pre-processing the dataset and then applied two versions of the transformer-based model AraBERT as well as the traditional machine learning method Linear SVC model with TF-IDF. The comparison indicated that AraBERT beats the regular ML model. However, there was not a significant improvement when comparing the base versions of this model.

In the future, investigating additional features should be considered. Some related highlights for this task can be tackled not only by transformer-based embeddings but also by word-vector based models. In addition, the larger versions of the AraBERT may give a better performance.

6. Acknowledgements

The research behind this paper would not have been possible without the extraordinary assistance of the team and supervisors, Mohammad Ammar Alsalka and Eric Atwell. Their knowledge and consideration kept my work on track. Moreover, my colleague Abdullah Alsaleh has helped me outline the work and added a significant enhancement. Similarly, we appreciate the insightful comments offered by the reviewers.

7. References

1. Wright, D. and I. Augenstein. *Claim Check-Worthiness Detection as Positive Unlabelled Learning*. in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. 2020.
2. Atanasova, P., et al. *Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. Task 1: Check-Worthiness*. in *CLEF (Working Notes)*. 2019.
3. Jaradat, I., et al., *ClaimRank: Detecting check-worthy claims in Arabic and English.* arXiv preprint arXiv:1804.07587, 2018.
4. Atanasova, P., et al., *Overview of the CLEF-2018 CheckThat! lab on automatic identification and verification of political claims. Task 1: Check-worthiness.* arXiv preprint arXiv:1808.05542, 2018.
5. Williams, E., P. Rodrigues, and V. Novak, *Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models.* arXiv preprint arXiv:2009.02431, 2020.
6. Novak, V., *Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models.* 2020.
7. Antoun, W., F. Baly, and H. Hajj, *Arabert: Transformer-based model for arabic language understanding.* arXiv preprint arXiv:2003.00104, 2020.
8. Devlin, J., et al., *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018.
9. Wadhawan, A., *Arabert and farasa segmentation based approach for sarcasm and sentiment detection in arabic tweets.* arXiv preprint arXiv:2103.01679, 2021.