R00 at NLP4IF-2021: Fighting COVID-19 Infodemic with Transformers and More Transformers

Ahmed Qarqaz **Dia Abujaber** **Malak A. Abdullah**
Jordan University of Science and Technology
Irbid, Jordan
afalqarqaz17, daabujaber17@cit.just.edu.jo
mabdullah@just.edu.jo

Abstract

This paper describes the winning model in the Arabic NLP4IF shared task for fighting the COVID-19 infodemic. The goal of the shared task is to check disinformation about COVID-19 in Arabic tweets. Our proposed model has been ranked 1st with an F1-Score of 0.780 and an Accuracy score of 0.762. A variety of transformer-based pre-trained language models have been experimented with through this study. The best-scored model is an ensemble of AraBERT-Base, Asafya-BERT, and ARBERT models. One of the study's key findings is showing the effect the pre-processing can have on every model's score. In addition to describing the winning model, the current study shows the error analysis.

1 Introduction

Social media platforms are highly used for expressing and delivering ideas. Most people on social media platforms tend to spread and share posts without fact-checking the story or the source. Consequently, the propaganda is posted to promote a particular ideology to create further confusion in understanding an event. Of course, it does not apply to all posts. However, there is a line between propaganda and factual news, blurred for people engaged in these platforms (Abedalla et al., 2019). And thus, social media can act as a distortion for critical and severe events. The COVID-19 pandemic is one such event.

Several previous works were published for using language models and machine learning techniques for detecting misinformation. Authors in Haouari et al. (2020b) presented a twitter data set for COVID-19 misinformation detection called "ArCOV19-Rumors". It is an extension of the "ArCOV-19" (Haouari et al., 2020a), which is a data set of Twitter posts with "Propagation Networks". Propagation networks refer to a post's retweets and conversational threads. Other authors in Shahi et al. (2021) performed an exploratory study of COVID-19 misinformation on Twitter. They collected data from Twitter and identified misinformation, rumors on Twitter, and misinformation propagation. Authors in Müller et al. (2020) presented CT-BERT, a transformer-based model pre-trained on English Twitter data. Other works that used Deep Learning models to detect propaganda in news articles (Al-Omari et al., 2019; Altiti et al., 2020).

The NLP4IF (Shaar et al., 2021) shared-task offers an annotated data set of tweets to check disinformation about COVID-19 in each tweet. The task asked the participants to propose models that can predict the disinformation in these tweets. This paper describes the winning model in the shared task, an ensemble of AraBERT-Base, Asafya-BERT, and ARBERT pre-trained language models. The team R00's model outperformed the other teams and baseline models with an F1-Score of 0.780 and an Accuracy score of 0.762. This paper describes the Dataset and the shared task in section 2. The Data Preprocessing step is presented in section 3. The experiments with the pre-trained language models are provided in section 4. Finally, the proposed winning model and methodology are discussed in section 5.

2 Dataset

The Data provided by the organizers Shaar et al., 2021 comprised of tweets, which are posts from the Twitter social media platform "twitter.com". The posts are related to the COVID-19 pandemic and have been annotated in a "Yes or No" question style annotation. The annotator was asked to read the post/tweet and go to an affiliated weblink (if the tweet contains one). For each tweet, the seven main questions that were asked are:

1. Verifiable Factual Claim: *Does the tweet contain a verifiable factual claim?*

Proceedings of the 4th NLP4IF Workshop on NLP for Internet Freedom, pages 104–109
June 6, 2021. ©2021 Association for Computational Linguistics

2. False Information: *To what extent does the tweet appear to contain false information?*

3. Interest to General Public:*Will the tweet affect or be of interest to the general public?*

4. Harmfulness: *To what extent is the tweet harmful to the society/person(s)/company(s)/product(s)?*

5. Need of Verification: *Do you think that a professional fact-checker should verify the claim in the tweet?*

6. Harmful to Society: *Is the tweet harmful the society and why?*

7. Require attention: *Do you think that this tweet should get the attention of government entities?*

For each question, the answer can be "Yes" or "No". However the questions two through five **depend** on the first question. If the first question (Verifiable Factual Claim) is answered "No", questions two through five will be labeled as "NaN". "NaN" is interpreted as there's no need to ask the question. For example, for the following tweet:

> *"maybe if i develop feelings for covid-19 it will leave".*

This tweet is not a verifiable factual claim. Therefore asking whether it's False Information or is in Need of Verification is unnecessary. Moreover, our model modified the values to be " No" for all text samples with labels annotated as "NaN".

Task Our team participated in the Arabic text shared task. The Arabic data set consists of 2,536 tweets for the training data, 520 tweets for the development (validation) data, and 1,000 tweets for the test data. It has been observed that the label distribution in the training data is unbalanced, as shown in Figure 1.

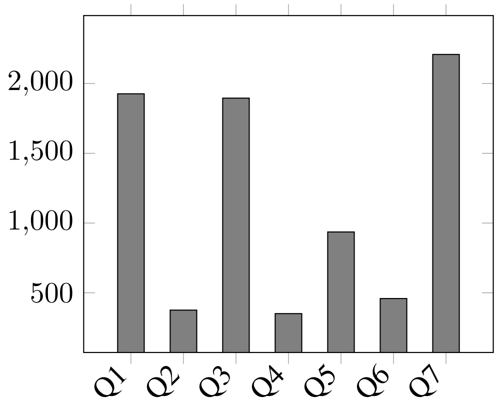


Figure 1: label distribution in data. Unbalance labels for questions.

3 Data Pre-Processing

Social media posts can contain noisy features, particularly the special characters (#, @, emojis, weblinks, etc..). Many elements within Arabic text can act as distortions for the model. We Tokenize the Arabic text [1], and for each sequence of tokens, we remove stop-words, numbers, and punctuation from the text. We also remove any non-Arabic terms in the text. Stemming and Segmentation are two common pre-processing operations done in Arabic Natural Language Processing. However, we do not apply them here, except in the case of AraBERT, where segmentation was applied.

4 Fine-tuning Pre-Trained Language Models

We approach the problem as a multi-label classification problem. For each label in a text sample, the label's value can be one (yes) or zero (no). In the training phase, we load the pre-trained language model (along with its corresponding tokenizer) and stack a linear classifier on top of the model.

This section describes the pre-trained Arabic language models that have been used in the study. The hyperparameters' fine-tuning is also detailed in this section in addition to the experiments' results.

4.1 Pre-trained Arabic Language Models

This section goes over the pre-trained language models experimented with through the study: AraBERT, Asafaya-BERT, ARBERT, and MARBERT.

- **AraBERT** (Antoun et al.) follows the original BERT pre-training (Devlin et al., 2018), employing the Masked Language Modelling task. It was pre-trained on roughly 70-million sentences amounting to 24GB of text data. There are four variations of the model: *AraBERTv0.2-base, AraBERTv0.2-large, AraBERTv2-base, AraBERTv2-large*. The difference is that the *v2* variants were trained on the pre-segmented text where prefixes and suffixes were split, whereas the *v0.2* were not. The models we used are the *v0.2* variants. the Authors recommended using the Arabert-Preprocessor powered by the farasapy[2] python package for the *v2* versions. Although the *v0.2* models don't require it, we

[1] Preprocessing was done using the NLTK Library
[2] farasapy

have found that the Arabert-Preprocessor improves the performance significantly for some experiments. So, we have used it with all the AraBERT models only.

- **Asafaya-BERT** (Safaya et al., 2020) is a model also based on the BERT architecture. This model was pre-trained on 8.2B words, with a vocabulary of 32,000 word-pieces. The corpus the model was pre-trained on was not restricted to Modern Standard Arabic, as they contain some dialectal Arabic, and as such Safaya et al. (2020) argue that this boosts the model's performance on data gathered from social media platforms. There are four variants of the model: Large, Base, Medium, and Mini. We only used Large and Base.
- **ARBERT** (Abdul-Mageed et al., 2020) is a pre-trained model focused on Modern Standard Arabic (MSA). It was trained on 61GB of text data, with a vocabulary of 100K Word-Pieces. There is only one variation of this model, which follows the BERT-Base architecture. It uses 12-attention layers (each with 12-attention heads) and 768 hidden-dimension. We use this model to possibly write some tweets (such as news updates) formally following MSA.
- **MARBERT** (Abdul-Mageed et al., 2020) argues that since AraBERT and ARBERT are trained on MSA text, these models are not well suited for tasks involving dialectal Arabic, which is what social media posts often are. MARBERT was trained on a large Twitter data set comprised of 6B tweets, making up about 128GB of text data. MARBERT follows the BERT-Base architecture but without sentence prediction. MARBERT uses the same vocabulary as ARBERT (100K Word-Pieces).

4.2 Fine-Tuning

Each model has been trained for 20 epochs. We found that after the 10th epoch, most of the model scores start to plateau. This is, of course, highly dependent on the learning rate used for each model. We have not tuned the models' learning rates, and rather we chose the learning rate we found best after doing multiple experiments with each model. We use a **Training Batch-Size** of 32 and a **Validation Batch-Size** of 16 for all the models. For each model's tokenizer we choose a **Max Sequence-length** of 100.

Each model has been trained on two versions of the data set, one that has not been pre-processed (We refer to it as "Raw") and one that has been pre-processed (we refer to it as "Cleaned"). A model that has been trained on cleaned data in training time will also receive cleaned text at validation and testing time. We apply the post-processing step, where for the labels Question-2, 3, 4, and Question-5, if a model predicts that Question-1 is "No" then the values of the mentioned Questions (Q2 through Q5) will be "NaN" Unconditionally. This, of course, assumes that the model can perform well on the first question. We report the results in Table 1.

Note: It is worth noting that, initially, we save the model on the first epoch along with its score as the "best-score". After each epoch, we compare the score of the model on that epoch with the best score. If the model's current score is higher than the best score, the model will be saved, and the model's best score will be overwritten as the current model's score. And as such, saying we train a model for 20 epochs is **not an accurate description** of the model's training. The score we used as criteria for saving was the Weighted F1-Score.

4.3 Results

We see (in Table 1) that generally, training on cleaned data either gave slightly better scores or no significant improvement, with ARBERT 4.1 being the exception. This is because ARBERT was specifically trained on Arabic text that followed the Modern Standard Arabic. Cleaning has normalized text for the model and removed features in the text that may otherwise act as noise. Furthermore, we conclude that Asafya-BERT 4.1 has a better performance when trained on Raw data, proving that a model pre-trained on Twitter data would perform better. Lastly, we observe that using a larger model (deeper network) does provide a slight improvement over using the Base version. [3]

5 Ensemble Pre-trained language Models

To maximize the scores, we resort to ensembling some of the models we fine-tuned on the data set. Ensemble models are known to improve accuracy

[3] Results and scores were generated using the Scikit-learn library

ID	Model	Data	Learning Rate	F1-Weighted	F1-Micro	Accuracy
(1)	AraBERT-Base	Raw	$3e^{-6}$	0.703	0.727	0.338
(2)	AraBERT-Base	Cleaned	$3e^{-5}$	0.735	0.725	0.394
(3)	AraBERT-Large	Raw	$3e^{-5}$	0.733	0.737	0.390
(4)	AraBERT-Large	Cleaned	$3e^{-5}$	0.747	0.749	0.425
(5)	MARBERT	Raw	$4e^{-5}$	0.737	0.741	0.382
(6)	MARBERT	Cleaned	$4e^{-6}$	0.735	0.735	0.413
(7)	ARBERT	Raw	$8e^{-6}$	0.715	0.728	0.407
(8)	ARBERT	Cleaned	$3e^{-5}$	0.734	0.745	0.398
(9)	Asafaya-Base	Raw	$5e^{-6}$	0.750	0.749	0.413
(10)	Asafaya-Base	Cleaned	$3e^{-5}$	0.707	0.743	0.382
(11)	**Asafaya-Large**	Raw	$5e^{-6}$	**0.750**	**0.752**	**0.436**
(12)	Asafaya-Large	Cleaned	$5e^{-6}$	0.737	0.743	0.373

Table 1: Shows model scores on the validation data set. The Weighted F1-Score and the Micro F1-Score are the average F1-Scores of the labels.

ID	Model	Q1	Q2	Q3	Q4	Q5	Q6	Q7
(1)	AraBERT-Base	0.73	0.11	0.71	0.22	0.37	0.43	0.84
(2)	AraBERT-Base	0.76	**0.26**	0.75	0.38	0.42	**0.55**	0.83
(4)	AraBERT-Large	**0.81**	0.16	**0.79**	0.32	0.42	0.50	**0.85**
(5)	MARBERT	0.78	0.12	0.78	0.36	0.43	0.44	0.84
(6)	MARBERT	0.75	0.10	0.74	**0.52**	**0.48**	0.54	0.84
(8)	ARBERT	0.78	0.19	0.78	0.36	0.44	0.53	0.83
(10)	Asafya-Base	0.78	0.11	0.77	0.30	0.22	0.39	0.84
(12)	Asafya-Large	0.79	0.18	0.78	0.40	0.35	0.48	0.84

Table 2: Shows models F1-Scores for the labels on the validation data set.

under the right conditions. If two models can detect different data patterns, then ensembling these two models would perhaps (in theory) give a better prediction. Of course, the process of finding a good ensemble is an empirical one. It involves a process of trial-and-error of combining different models and choosing the best one. However, as we show in Table 1 various combinations can be done, and as a result, trying all combinations would perhaps be impractical. We mention in Section-2 that the label distribution in the data set is unbalanced, and hence for labels like Question-2 (False Information), the model can give poor predictions for the answer to that label. However, suppose we were to acquire a model (through experimentation) that tends to perform well in predicting that label. In that case, we could ensemble this model with one that generally performs well to get a better overall score.

Strategy Through experimentation and for each label, train a model that performs well on that label and save it for an ensemble. Then, train a model that generally performs well on all labels (relative to the models at hand) and save it for an ensemble. After collecting several models, ensemble these models through various combinations. And for each ensemble, record the combination and its score (performance on validation data). Choose the best performing ensemble.

Weighted-Average Our approach for an ensemble is to take the weighted average of each's model predictions for each sample. Each model produces a vector of probabilities (whose length is equal to the number of labels) for each tweet. We take the weighted average point-wise and then apply a 0.5-threshold to decide if a label is one (yes) or zero (no). We suggest using the weighted average rather than a normal average with equal weights to

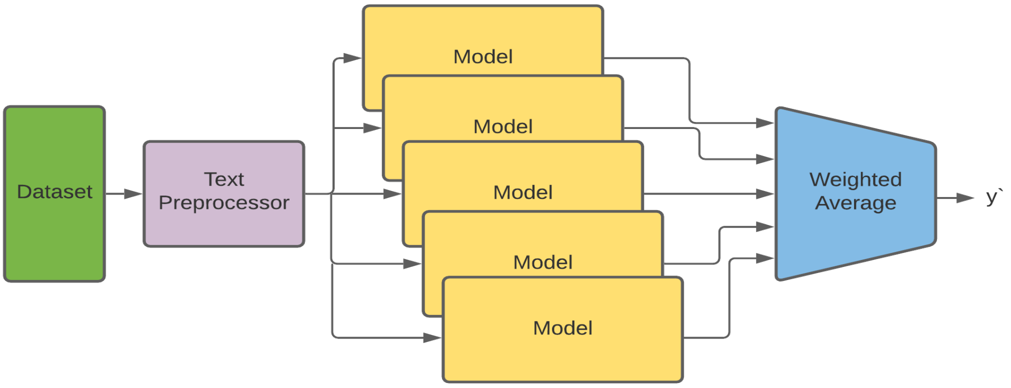


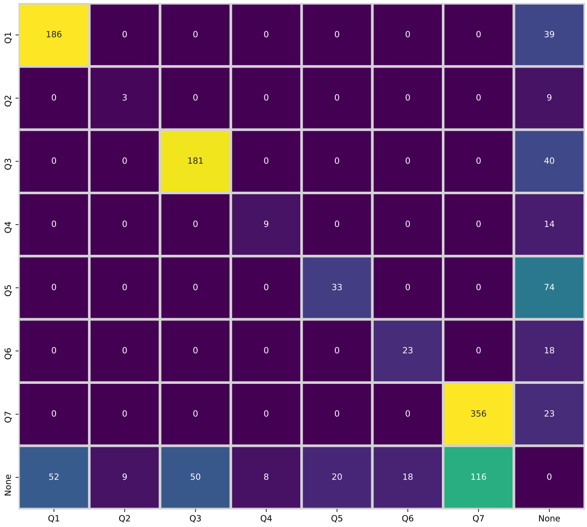
Figure 2: Shows Ensemble architecture. Each model has its classifier stacked on top. The models receive the text pre-processed and produce logits. Logits are then inserted into a Sigmoid layer making predictions. Prediction vectors are multiplied with a scalar (the weight), and the weighted average is calculated point-wise.

give higher confidence to the generally performing model as opposed to the less generally performing one. The intuition is that you would want the model to be the deciding factor in predicting better overall performance. The models with the lesser weights are merely there to increase the models' confidence in predicting some labels. The optimal weights for an ensemble are obtainable through experimentation. As a hyperparameter, they can be tuned.

Proposed Model We ensemble five models as shown in Figure 2, all of them were trained on cleaned data. And so, the models were tested on cleaned data. The models are:

1. Model (2): AraBERT-Base, with a weight of 3.
2. Model (4): Asafya-BERT-Large, with a weight of 3
3. Model (10): Asafya-BERT-Base, with a weight of 1.
4. Model (12): AraBERT-Large, with a weight of 1.
5. Model (8): ARBERT, with a weight of 3.

Our model achieved an F1-Weighted Score of 0.749, an F1-Micro Score of 0.763, and an Accuracy of 0.405 on validation data. It also earned an F1-Weighted Score of 0.781 and an Accuracy of 0.763 on the Test data. These results made the model ranked the first mode since it is the top-performing model in the shared task. Figure 3 presents the confusion matrix for the Ensemble-model predictions on the labels.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	None
Q1	186	0	0	0	0	0	0	39
Q2	0	3	0	0	0	0	0	9
Q3	0	0	181	0	0	0	0	40
Q4	0	0	0	9	0	0	0	14
Q5	0	0	0	0	33	0	0	74
Q6	0	0	0	0	0	23	0	18
Q7	0	0	0	0	0	0	356	23
None	52	9	50	8	20	18	116	0

Figure 3: Shows confusion matrix for the Ensemble-model predictions on the labels. The Y-axis represents the *True*-Label while the X-axis represents the *Predicted*-label.

6 Conclusion

This paper described the winning model in the NLP4IF 2021 shared task. The task aimed to check disinformation about COVID-19 in Arabic tweets. We have ensembled five pre-trained language models to obtain the highest F1-score of 0.780 and an Accuracy score of 0.762. We have shown the performances of every pre-trained language model on the data set. We also have shown some of the models' performances on each label. Moreover, we have demonstrated the confusion matrix for the ensemble model. We have illustrated that a pre-trained model on Twitter data (Asafya-Bert in Section 4.1) will perform better relative to a model that hasn't.

References

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

Ayat Abedalla, Aisha Al-Sadi, and Malak Abdullah. 2019. A closer look at fake news detection: A deep learning perspective. In *Proceedings of the 2019 3rd International Conference on Advances in Artificial Intelligence*, pages 24–28.

Hani Al-Omari, Malak Abdullah, Ola AlTiti, and Samira Shaikh. 2019. JUSTDeep at NLP4IF 2019 task 1: Propaganda detection using ensemble deep learning models. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 113–118, Hong Kong, China. Association for Computational Linguistics.

Ola Altiti, Malak Abdullah, and Rasha Obiedat. 2020. JUST at SemEval-2020 task 11: Detecting propaganda techniques using BERT pre-trained model. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1749–1755, Barcelona (online). International Committee for Computational Linguistics.

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

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*.

Fatima Haouari, Maram Hasanain, Reem Suwaileh, and Tamer Elsayed. 2020a. Arcov-19: The first arabic covid-19 twitter dataset with propagation networks. *arXiv preprint arXiv:2004.05861*, 3(1).

Fatima Haouari, Maram Hasanain, Reem Suwaileh, and Tamer Elsayed. 2020b. Arcov19-rumors: Arabic covid-19 twitter dataset for misinformation detection. *arXiv preprint arXiv:2010.08768*.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*.

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

Shaden Shaar, Firoj Alam, Giovanni Da San Martino, Alex Nikolov, Wajdi Zaghouani, Preslav Nakov, and Anna Feldman. 2021. Findings of the NLP4IF-2021 shared task on fighting the COVID-19 infodemic and censorship detection. In *Proceedings of the Fourth Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF@NAACL' 21, Online. Association for Computational Linguistics.

Gautam Kishore Shahi, Anne Dirkson, and Tim A Majchrzak. 2021. An exploratory study of covid-19 misinformation on twitter. *Online Social Networks and Media*, 22:100104.