NGU_CNLP at WANLP 2022 Shared Task: Propaganda Detection in Arabic

Ahmed Samir
Information Technology Institute
ahmedsamirio95@gmail.com

Abu Bakr Soliman
Nile University
ab.soliman@nu.edu.eg

Mohamed Ibrahim
New Giza University
mohamed.shafik@ngu.edu.eg

Laila Hesham
New Giza University
laila.afify@kaust.edu.sa

Samhaa R. El-Beltagy
New Giza University/Optomatica
samhaa@computer.org

Abstract

This paper presents the system developed by the NGU_CNLP team for addressing the shared task on Propaganda Detection in Arabic at WANLP 2022. The team participated in the shared tasks' two sub-tasks which are: 1) Propaganda technique identification in text and 2) Propaganda technique span identification. In the first sub-task the goal is to detect all employed propaganda techniques in some given piece of text out of a possible 17 different techniques, or to detect that no propaganda technique is being used in that piece of text. As such, this first sub task is a multi-label classification problem with a pool of 18 possible labels. Subtask 2 extends subtask 1, by requiring the identification of the exact text span in which a propaganda technique was employed, making it a sequence labeling problem. For task 1, a combination of a data augmentation strategy coupled with an enabled transformer-based model, comprised our classification model. This classification model ranked first amongst the 14 systems participating in this subtask. For subtask two, a transfer learning model was adopted. The system ranked third among the 3 different models that participated in this subtask.

1 Introduction

The term propaganda was coined in the seventeenth century as a means of disseminating noble ideas among groups of individuals. Over time, it has become known for referring to the use of infused ideas, news or partial arguments to groups of people with the intention of manipulating their beliefs and behaviors, typically, towards deceptive agendas. In today's world, we can find various forms of propaganda in almost every newspaper article, social media post or mass media broadcasting. It is rarely the case that individuals are simply informed without being pervasively biased. Moreover, propaganda propagation is no longer exclusively dominated by religious, political or demographic entities, but even by individuals where a variety of agendas are involved. With such huge proliferation and the expected undesired influences, it is of utmost importance to be able to detect faulty or misleading information for the purpose of efficiently and promptly handling the widespread of fallacies. However, detecting computational propaganda is a significantly involved task especially with the ever growing efforts to make it go inconspicuous. By leveraging tools from machine learning (ML), it is possible to automate the necessary NLP tasks required to detect such malicious agendas. Computational propaganda detection has gained immense attention [1, 2, 3, 4] in the NLP community. For example, Google and Facebook, are currently testing ML-powered fact-checking tools to investigate the authenticity of shared information for the purpose of fighting potential "infodemics" [5, 2] which are a form of propaganda. Typically, NLP tasks utilize several modeling approaches such as n-gram models and neural networks models. In n-gram modeling, n-words are being paired and processed. Neural networks are essential to the success of numerous NLP tasks.

NLP models have been revolutionized by the introduction of contextualized embeddings such as the the BERT transformer-based language model, first introduced by Google [6]. The idea of the BERT-model is that it can read a sentence simultaneously in both directions. Applying this to the task of propaganda detection is made possible through two subtasks. The first of which is the identification of the propaganda technique in the sentence together with the corresponding text span. The second subtask is concerned with the classification of the deployed propaganda technique out of the 18 well-known propaganda techniques [7].

In [8], a logistic regression-based model was proposed to detect a propagandist text, along with features acquired from Linguistic Inquiry and Word Count (LIWC) text analysis software, to solve a binary classification problem. Fine-tuning of the BERT transformer model was performed in [9] where, prior to using the BERT architecture, the authors initially concentrated on the pre-processing phases to offer additional details about the language

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 545 - 550

Dataset	Train	dev	dev_test
No. of Tweets	504	52	52

Table 1: Data Distribution for Subtask 1

model and current propaganda strategies. However, the author later utilised the BERT architecture to frame the work as a problem of sequence labelling. In [10], some linguistic characteristics and global noncontextual word embeddings were exploited.

This paper presents the system developed by the NGU_CNLP team for addressing the shared task on Propaganda Detection in Arabic at WANLP 2022 [11]. The team participated in the shared tasks' two sub-tasks which are: 1) Propaganda technique identification in text and 2) Propaganda technique span identification. In the first sub-task the goal is to detect all employed propaganda techniques in some given piece of text out of a possible 17 different techniques, or to detect that no propaganda technique is being used in that piece of text. As such, this first sub task is a multi-label classification problem with a pool of 18 possible labels. Subtask 2 extends subtask 1, by requiring the identification of the exact text span in which a propaganda technique was employed, making it a sequence labeling problem.

2 Subtask 1: Propaganda Classification

The first sub-task our team participated in, was a multi-label classification problem. In this subtask, the input is a single piece of text (a tweet), and the required output, is the set of the propaganda techniques used in it. The evaluation metric for this subtask was the micro-average F_1 score.

2.1 Initial Experimentation

Three labeled datasets were provided by the task organizers for training (train), validation (dev), and testing (dev_test) during the model development phase. The distribution of this data is shown in Table 1.

The total number of labels in this dataset was 18. Some of the tweets had as many as 5 labels, with the average number of labels/tweets being 1.75. Eight out of the 18 different labels/classes had less than 10 instances in the training dataset, i.e., they were very under-represented. Furthermore, careful analysis of the labels' distribution in the provided 3 datasets revealed that there is a discrepancy in the distribution of labels among them as shown in Figure 1.

To better understand the problem, gain insights into its challenges, as well as to establish a good baseline, we decided to apply traditional ML algorithms on the training data, and to test on the aggregated set of dev and dev_test data. Simple text pre-processing was carried out in this step which included normalization, diacritic removal, url removal and number removal. Text was then tokenized and represented as a bag of words. Classifiers that were used in this step included, Support Vector Machines ((with a multitude of k values), Naïve Bayes, Stochastic Gradient Descent, Logistic regression, Random Forests and simple K-nearest Neighbor. After experimenting with various configurations, the best result obtained was from the Linear Support Vector Machine with a micro average F_1 score of 0.44. However, looking at the F_1 scores of individual classes revealed that the majority of classes had zero as a score, highlighting the class imbalance problem seen in the training data.

2.2 Data Redistribution and Augmentation

To address the discrepancy in the distribution of labels in the provided datasets and to avoid the negative impact of this discrepancy on the quality of the prediction model, all three sets were merged and the re-split using multi-label stratification to ensure more uniform label distribution. The results of carrying out this step, are shown in Figure 2. Unfortunately, under-represented labels remained under represented even after the merge and re-split steps.

To overcome the fact that the training data size was quite small (only 504 instances) and in an attempt to provide more examples for under-represented labels thus addressing the class imbalance problem, means for expanding the training dataset were sought. The one that was adopted, was the translation of a similar dataset which is available in English to Arabic. The used dataset was taken from SemEval-2020 Task 11 [12] which targeted the detection of propaganda techniques in news articles. Data from the SemEval-2020 Task was translated to Arabic using the RapidAPI Translation tool [1].

Since the SemEval-2020 training data labels did not directly map to the labels used in the WANLP

[1] https://rapidapi.com/gofitech/api/nlp-translation/

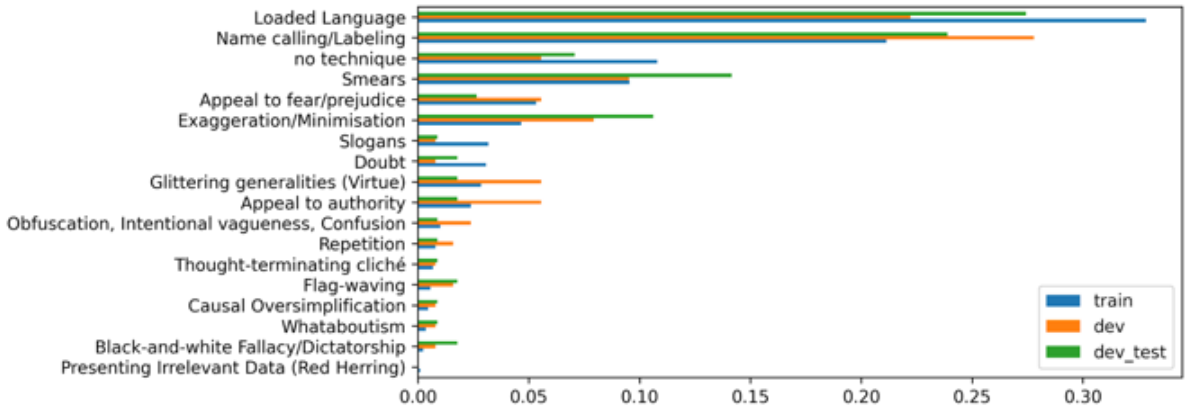

Figure 1: Label Distribution for Data of Subtask 1

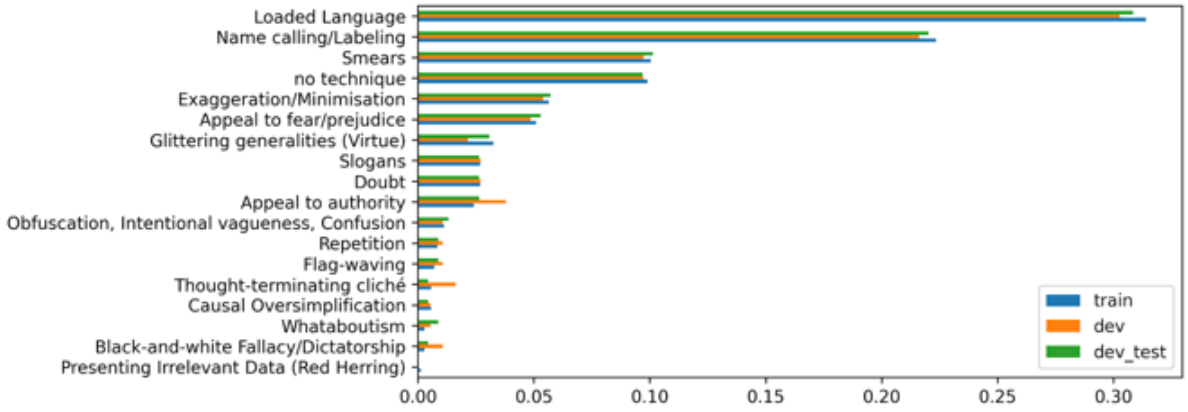

Figure 2: Data ReDistribution for Subtask 1

2022 shared task, a mapping function was created to convert the names of the SemEval labels to WANLP ones. In the end, 3,938 sentences were added to the original WANLP dataset bringing up the number of training instances from 504 to 4, 442.

2.3 Overview of the Adopted Model

Following the step described in the previous subsection, we decided to experiment with a more powerful prediction model. Since transformer based models have shown superior results in text classification tasks, the model we used was AraBERT [13]. Consequently, text preprocessing was done using the AraBERT preprocessor with the default configuration. Hyperparameters were tuned and optimized through the use of randomized grid search. The final used configuration was as follows:

- Proportion of extra data sampled to be in train dataset: 0.7
- Max. length of tokenization: 128
- Batch size: 8
- Number of epochs: 50 with early stopping
- Learning rate: 0.0001
- Learning rate scheduler: Linear
- Warm-up ratio: 0.1

The metric used for evaluation was the F_1 Micro score. The best configuration evaluation loss on the dev set can be seen in Figure 3.

This final configuration was used to train 5 models on 5-fold splits of the labeled data (train, dev and dev_test). The prediction probabilities of each model were averaged into the final prediction probabilities, and then the threshold for prediction was set (empirically) to be 0.4.

2.4 Final Results

For this shared task, the task organizers provided 440 unlabeled tweets. The model described in the previous section was used to predict various labels

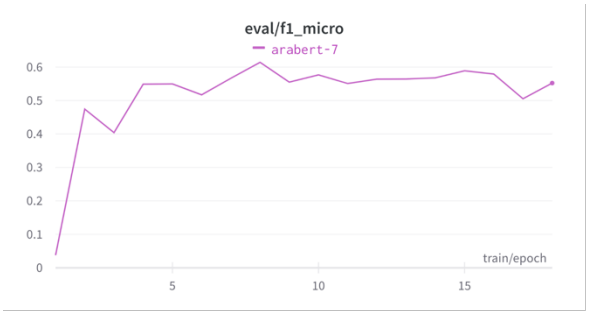

Figure 3: Model Performance for Subtask 1

for each tweet. The final results released by the task organizers have shown that the model that we have developed ranked at number one with an average micro F_1 score of 0.649. The next best performing system achieved a score of 0.609.

3 Subtask 2: Propaganda Span Detection

In the second task, the goal was not only to identify propaganda techniques used in a piece of text, but the exact text span that represents each technique. This can be thought of as a sequence tagging task or as a token classification [14].

3.1 Exploratory Data Analysis

The provided data distribution for this task was identical to subtask 1, except that here exact spans for each of the labels were provided. So similar to classes in subtask 1, some propaganda techniques (labels) such as Whataboutism and Causal Oversimplification were under represented while others such as Name calling/Labeling and Loaded Language were over represented. An example of a tweet taken from this dataset is shown in Figure 4. In this example, spans which are given in numbers representing their start and end positions, are mapped to their equivalent text fragments.

As can be seen in this example, it is possible to assign the same label more than once in the same tweet as well as to assign more than one label to the same span. So in the shown example, 'Loaded Language' appears twice, and the span (مقامر ومجنون) is labeled as both 'Loaded Language' as well as 'Name Calling/Labeling'. This pattern of multiple label assignment for the same span, appeared in 53 locations in the train dataset, 18 in the dev dataset and 16 in the dev_test dataset. The dataset used for training, was the augmented translated one described in Subtask 1.

ID	1399057217349881860		
Tweet	"مقامر ومجنون".. الصحافة البريطانية تشن هجوماً حاداً على "غوارديولا" بعد ضياع دوري الأبطال		
Labels			
	Span	مقامر ومجنون	مقامر ومجنون	هجوماً حاداً
	Label	Name calling/Labeling	Loaded Language	Loaded Language

Figure 4: An Example of a Tweet taken from Subtask 2 Dataset

3.2 Preprocessing

One of the challenges of preprocessing the texts for this task, is that the final output must a span denoted in terms of the position of its first character and the positions of its last character in the text, which would then be compared with spans represented in a similar way in the test data provided. Preprocessing had to be handled carefully so that it does not change the order of the letters contained in the texts. This was done by applying only simple operations such as normalization. To get the data ready for the next step, we transformed the data into the widely used style of data representation BIO so that each span in the tweet is accompanied by a distinctive tag that indicates if it is outside the classification (O), the beginning of a classification (B), or within the classification span (I). When a single span was assigned to more than one technique, we neglected the technique that is most representative in the train dataset.

3.3 Overview of the Adopted Model

Using modern transformers and neural networks techniques, the proposed solution relied on employing a previously developed model that addresses a similar task in Arabic in order to transfer its experience for solving this particular problem. Specifically, we used the Marefa-NER model, which is one of the pre-trained templates available on the HuggingFace platform and which targets Named Entity Recognition (NER). The model was pretrained to identify 9 different types of entities within any news text or Wikipedia article.

After preparing the training data using the BIO format, a neural network was setup for a token classification problem, In other words, the network was responsible for assigning an appropriate class for each token in the text. Tokenizing the text was based on the originally followed strategy in Marefa-NER which was XLM-RoBERTa [15].

Hyperparameter tuning was performed through a series of experiments with the most important of these values being:

- Max. length of tokenization: 512
- Batch size: 8
- Number of epochs: 14 with early stopping
- Learning rate: 0.00001
- Learning rate scheduler: Linear
- Optimizer: Adam
- No. Hidden Layers: 24
- No. Attention heads: 16

Using the 'dev_test' dataset with the aim of optimizing the F_1-scores, Figure 5 shows the progress made with the F_1-scores during the training process while Figure 6 shows the training loss decreasing gradually.

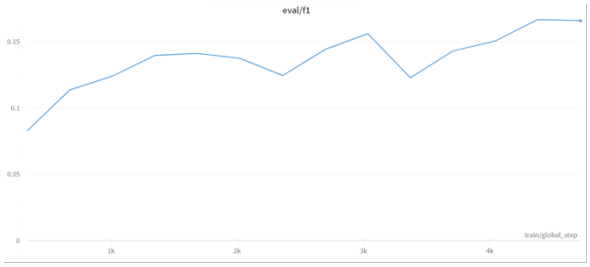

Figure 5: F_1-score of our Model during training for Subtask 2

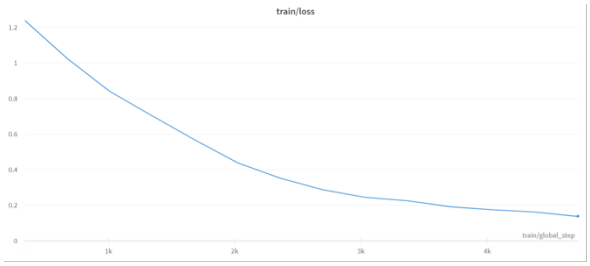

Figure 6: Training Loss of our Model for Subtask 2

After completing the whole training process, the model with the highest F_1-Score was retrieved and adopted. The best model results are shown in Table 2.

Training	Validation	F_1-score	Accuracy
0.1637	1.2753	0.1669	0.7815

Table 2: Model's validation results for subtask 2

4 Summary

The winning system for the propaganda classification task and the third-placed system for the propaganda span identification task has been described. Both of the developed solutions used transformer models. For subtask 1, the classification task was approached with the AraBert architecture and data augmentation. Final predictions were obtained based on an ensemble of 5 models. For subtask 2, the Marefa-NER model together with the XLM-RoBERTa as a tokenizer, were used to tackle the sequence tagging task with same translated data from subtask 1 to overcome the small and imbalanced dataset provided. An interesting future research direction would be to perform error analysis and conduct ablation studies to get more insights from the reported results and improve the models accordingly.

References

[1] Bolsover Gillian and Philip Howard. Computational propaganda and political big data: Moving toward a more critical research agenda. In *Big Data*, pages 273–376, April 2017.

[2] Akshay Jain and Amey Kasbe. Fake news detection. In *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–5, 2018.

[3] Hani Al-Omari, Malak Abdullah, Ola Altiti, and Samira Shaikh. Justdeep at nlp4if 2019 task 1: Propaganda detection using ensemble deep learning models. *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, 2019.

[4] Shaina Raza and Chen Ding. Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics*, page 335–362, 2022.

[5] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206, 02 2022.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[7] Miller C. R. The techniques of propaganda. from "how to detect and analyze propaganda,". In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *an address given at Town Hall*. The Center for learning, 1939.

[8] Jinfen Li, Zhihao Ye, and Lu Xiao. Detection of propaganda using logistic regression. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 119–124, Hong Kong, China, November 2019. Association for Computational Linguistics.

[9] Shehel Yoosuf and Yin Yang. Fine-grained propaganda detection with fine-tuned BERT. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 87–91, Hong Kong, China, November 2019. Association for Computational Linguistics.

[10] Tariq Alhindi, Jonas Pfeiffer, and Smaranda Muresan. Fine-tuned neural models for propaganda detection at the sentence and fragment levels. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 98–102, Hong Kong, China, November 2019. Association for Computational Linguistics.

[11] Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Preslav Nakov, and Giovanni Da San Martino. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop*, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics.

[12] Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online), December 2020. International Committee for Computational Linguistics.

[13] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May 2020. European Language Resource Association.

[14] Zhiyong He, Zanbo Wang, Wei Wei, Shanshan Feng, Xianling Mao, and Sheng Jiang. A survey on recent advances in sequence labeling from deep learning models, 2020.

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