Nullpointer at ArAIEval Shared Task:
Arabic Propagandist Technique Detection with Token-to-Word Mapping in Sequence Tagging

Abrar Abir
Computer Science Program,
Carnegie Mellon University in Qatar
Doha, Qatar
abir@cmu.edu

Kemal Oflazer[*]
Language Technologies Institute,
Carnegie Mellon University
Pittsburgh, PA, USA
ko@andrew.cmu.edu

Abstract

This paper investigates the optimization of propaganda technique detection in Arabic text, including tweets & news paragraphs, from ArAIEval shared task 1. Our approach involves fine-tuning the AraBERT v2 model with a neural network classifier for sequence tagging. Experimental results show relying on the first token of the word for technique prediction produces the best performance. In addition, incorporating genre information as a feature further enhances the model's performance. Our system achieved a score of 25.41, placing us 4th on the leaderboard. Subsequent post-submission improvements further raised our score to 26.68.

1 Introduction

The spread of propagandistic content on social media and news platforms poses a significant challenge to the integrity of information consumed by the public. Detecting and countering such content is crucial for maintaining a well-informed society. Identifying propaganda techniques in Arabic texts presents unique challenges due to the language's complex morphology and the use of diverse dialects. Our main contribution is developing an optimized system that leverages AraBERT for embeddings and a neural network classifier for sequence tagging to detect propagandistic techniques in Arabic texts. We propose a robust preprocessing pipeline for Arabic, addressing Unicode inconsistencies, misaligned annotations, and user-mention normalization. We also evaluate and compare different sequence tagging approaches.

The paper is structured as follows: Section 2 describes the related work. Section 3 describes the data preprocessing steps. Section 4 discusses the sequence tagging prediction approaches. Section 5 presents the details of our system. Section 6 reports on the experimental results and discussions. Section 7 concludes the paper.

2 Related Work

Detecting propaganda and persuasion techniques in Arabic text has gained significant attention in recent years. Hasanain *et al.* (2023b) have introduced the ArAIEval shared task, focusing on detecting propagandistic techniques in unimodal and multimodal Arabic content, highlighting the importance of this task in the Arabic language context. Alam *et al.* (2022) have provided an overview of the WANLP 2022 shared task on propaganda detection in Arabic, discussing the challenges and significance of this task in the Arabic language domain. Several studies have explored the use of large language models for propaganda detection. Hasanain *et al.* (2023a) investigated the application of large language models for propaganda span annotation, showcasing their potential for identifying propagandistic content. In a follow-up study, Hasanain *et al.* (2024a) have evaluated the capability of GPT-4 in identifying propaganda and annotating propaganda spans in news articles, demonstrating the effectiveness of advanced language models in this domain.

The ArAIEval shared task, organized by Hasanain *et al.* (2023b), focused on persuasion techniques and disinformation detection in Arabic text, emphasizing the need for language-specific approaches in tackling these issues. These studies demonstrate the growing interest in propaganda detection and the diverse approaches explored to address this challenge, particularly in Arabic. Our work builds upon these existing efforts, focusing on the role of token-to-word mapping in Arabic sequence tagging for propaganda detection.

---

[*]At the time of the research, Kemal Oflazer was also affiliated with Carnegie Mellon University, Qatar.

3 Preprocessing the Data

In this section, we describe our data and the pre-processing steps applied to ensure the annotation consistency and accuracy in the text snippets.

Dataset The dataset is taken from the ArAIEval 2024 task 1.[1] The dataset contains JSONL files with UTF-8 text encodings. Given a multigenre text snippet (a news paragraph or a tweet) from the dataset, the task is to detect the propaganda techniques used in the text together with the exact span(s) in which each propaganda technique appears. Table 1 presents basic statistics on the dataset.

Removal of Unicode Control Characters To handle emojis and other Unicode characters commonly found in tweets, we remove all Unicode characters classified as control ("Cf") or other ("Co") and replace them with spaces. Replacement with spaces instead of just removing them preserves the character positions and avoids disrupting the annotation spans in the text.

Handling Misaligned Span Annotations For some annotations in the dataset, the reported span of the propagandistic technique ("start" and "end") does not match with the actual start and character positions of the propagandistic technique substring in the main string; e.g.,

```
{
    "id": "1395982884532523010",
    "text": "المسجد_الأقصى #غزة_تنتصر# LINK مشعل: القضية الفلسطينية واحدة لا تتجزأ",
    "labels": [
        {"start": 78, "end": 88, "technique": "Slogans", "text": "#غزة_تنتصر"}
    ]
}
```

In this instance, the annotation span starts at character 78 and ends at character 88, but the text is only 69 characters long. To address this issue, we completely ignore the reported span ("start" and "end") and perform a substring search using the text technique provided in the annotation to determine the correct character spans.

Normalizing User Mentions In the dataset, user mentions in the text were replaced with a generic placeholder "@USER". However, this replacement was not reflected in the technique annotations. Consider the following example:

Table 1: Data Distribution by Type

Type	Train	Dev	Test	**Total**
Tweet	995	249	260	1504
Paragraph	6002	672	786	7476

```
{
    "id": "1274014147739516935",
    "text": "الشعب يموت جوع @USER @USER @USER وين فلوس النفط يا LINK",
    "labels": [
        {
            "start": 0,
            "end": 78,
            "technique": "Questioning_the_Reputation",
            "text": "الشعب يموت جوع @HadiPresident @DrMaeenSaeed @alimohsnalahmar وين فلوس النفط يا"
        }
    ]
}
```

We replaced all specific user mentions in the technique annotations to resolve this discrepancy with the generic "@USER" placeholder. This normalization ensured consistency between the text and its annotations.

Manual Handling of Failed Substring Searches Despite the automated preprocessing steps, there were cases where the substring search failed due to discrepancies in Unicode handling. For instance:

```
{
    "id": "3Fa3SRBNPLP4ND9qWHQY77_002",
    "text": "جلب وحاز المادة المخدرة (الهيروين) والمؤثر العقلي الميثامفيتامين بقصد الاتجار في غير الأحوال المرخص بها ....",
    "labels": [
        {
            "technique": "Questioning_the_Reputation",
            "text": "جلب وحاز المادة المخدرة (الهيروين) والمؤثر العقلي الميثامفيتامين بقصد الاتجار في غير الأحوال المرخص بها",
            "start": 48,
            "end": 158
        }
    ],
    "type": "paragraph"
}
```

Here, the parentheses around *[alhirwin - Heroin]* were reversed in the main text compared to the annotation, likely due to improper Unicode handling of [U+202C] and [U+202D]. Such cases were manually corrected to ensure the annotations matched the text accurately.

These preprocessing steps ensured the dataset was clean and the annotations aligned correctly with the text.

4 Sequence Tagging Prediction Approaches

This section discusses various approaches to sequence tagging for detecting propagandistic techniques in text.

4.1 Token-Level Training and Prediction

In this approach, the model is trained on labels assigned to each token, and predictions are made for each token. The predicted spans of propagandistic techniques are then computed based on these token-level predictions. One limitation of this method is that the predicted spans might start or end in the middle of a word since words are often broken into multiple tokens.

4.2 Token-Level Training with Word-Level Prediction

To address the issue of token-level span inaccuracies, we explore two methods where the model is still trained on token-level labels but predictions are aggregated at the word level.

Majority Label Assignment In this approach, each word, when broken into multiple tokens, receives multiple labels. We assign the word the label that appears most frequently among its tokens. For instance, if a word is split into three tokens and one token is predicted to have label 1 while the other two tokens are predicted to have label 2, we assign the word label 2.

First Token Label Assignment In this approach, for each word broken into multiple tokens, we only consider the label of the first token to assign the label to the entire word.

4.3 Word-Level Training and Prediction

In this approach, the model is trained directly on labels assigned to each word rather than each token. To achieve this, we determine the embedding of each word by max-pooling the embeddings of its constituent tokens. This aggregated word embedding is then used for both training and prediction at the word level.

5 The Propagandistic Technique Detection System

In this section, we describe the system used for the task of detecting propagandistic techniques in Arabic text snippets. Our system is based on the AraBERT version 2 pre-trained model, followed by a neural network classifier for sequence tagging as shown in Figure 1. Figure 1 (a) shows the structure of one of our classifiers and Figure 1 (b) shows the overall structure. There is one classifier for each token in the input.

Embedding Generation & Classification For generating embeddings, we utilized the AraBERT version 2 base pre-trained model,[2] a BERT-based model specifically designed for Arabic language processing. For each token, we concatenate the embedding of the `[CLS]` token with the token's contextual embeddings. Then, input is fed into a classifier to predict the labels corresponding to different propagandistic techniques. The architecture
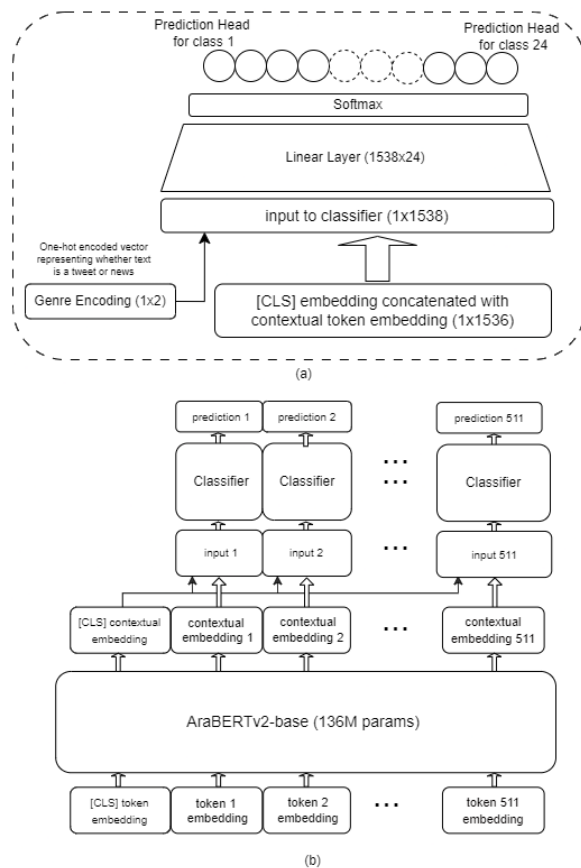
Figure 1: Diagram of the (a) Classifier, (b) Sequence Tagging System

of the neural network is quite simple; it has no hidden layers, and has a total of $1536 \cdot 24 = 36,864$ parameters to train.

Sequence Tagging Approaches We experimented with four different sequence tagging prediction approaches, as described in Section 3.

Encoding Genre as a Feature We incorporated genre information as an additional feature in our model. Given the dataset comprises two distinct genres—tweets and news paragraphs—we encoded this information using a one-hot vector representation. This two-dimensional genre vector was then concatenated with the $(768 \cdot 2 = 1536)$ two-dimensional embedding vector as the input to our neural network classifier.

Hyperparameter Tuning We experimented with various learning rates, batch sizes, number of epochs, and the inclusion of hidden layers. The hyperparameter tuning was performed using grid search and cross-validation to identify the optimal set of parameters for our final model.

Final Model Training The final model was trained on concatenating the train and development dataset using the best sequence tagging approach and the optimal hyperparameters determined from our tuning experiments. This model was then used to predict the spans of propagandistic techniques in the test dataset, providing the results for our evaluation.

6 Results and Discussions

Table 2 summarizes our results for different techniques and whether genre information is used or not.

Table 2: Performance Metrics (Micro F1 score) on Test File for Different Sequence Tagging Approaches - with & without genre encoding

Approach	M1 score **with** Genre	M1 score **without** Genre
Token-to-Token	24.34	22.62
Token-to-Word (Majority-Label)	20.73	16.57
Token-to-Word (First-Label)	**26.68**	**24.37**
Word-to-Word	12.94	13.22

`[CLS]` Token Embedding The `[CLS]` token serves as a representation of the entire input sequence. By concatenating this sequence representation with individual token embeddings, the model gains access to global and local contextual cues, enabling better predictions of sequence labels. Simply adding the `[CLS]` embedding vector to the token embeddings produced slightly better outcomes compared to ignoring the `[CLS]` token altogether. However, this approach still fell short of the performance achieved through concatenation.

Different sequence tagging approaches The word-level-first sequence tagging technique emerged as the most effective method. This result is consistent with the nature of the Arabic language, where the essence or primary meaning of a word is often captured by the first token, especially for verbs. Conversely, the word-level-majority sequence tagging technique performed poorly compared to other methods. Despite aggregating token-level predictions at the word level, this approach did not yield significant improvements in accuracy. This suggests that the majority label assignment strategy may dilute the relevance of the primary propagandistic technique within each word, particularly in Arabic.

Genre Encoding Incorporating genre information into our model significantly improved performance across all sequence tagging approaches - except for the word-to-word approach. By providing the model with contextual cues about whether a text snippet was a tweet or a news paragraph, the model could better differentiate and identify the propagandistic techniques prevalent in each genre. This enhancement in performance, however, came at the cost of increased convergence time. Specifically, the inclusion of genre information extended the convergence period to around 10 epochs (as opposed to around 3 epochs).

Classifier Architecture In our experimentation, we explored various neural network architectures, ranging from models with 1 to 2 hidden layers containing 32 to 128 neurons on each layer to more complex configurations. Surprisingly, the best-performing model emerged as one with no hidden layers at all.

7 Conclusion

The study underscores the importance of methodological choices in improving the accuracy of propaganda technique detection models. Concatenating the `[CLS]` token embedding and leveraging the word-level-first sequence tagging approach, particularly when supplemented with genre information, proves to be a robust strategy. Future research should continue to explore the integration of additional contextual information and refine these techniques to further enhance the detection and understanding of propagandistic content across various languages and genres.

References

Firoj Alam, Abul Hasnat, Fatema Ahmed, Md Arid Hasan, and Maram Hasanain. 2024. ArMeme: Propagandistic content in Arabic memes. *arXiv preprint arXiv:2406.03916*.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 108–118, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Maram Hasanain, Fatema Ahmed, and Firoj Alam. 2023a. Large language models for propaganda span annotation. *arXiv preprint arXiv:2311.09812*.

Maram Hasanain, Fatema Ahmed, and Firoj Alam. 2024a. Can GPT-4 identify propaganda? annotation and detection of propaganda spans in news articles. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources And Evaluation, LREC-COLING 2024*, Torino, Italy.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023b. ArAIEval Shared Task: Persuasion techniques and disinformation detection in Arabic text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Maram Hasanain, Md. Arid Hasan, Fatema Ahmed, Reem Suwaileh, Md. Rafiul Biswas, Wajdi Zaghouani, and Firoj Alam. 2024b. ArAIEval Shared Task: Propagandistic techniques detection in unimodal and multimodal Arabic content. In *Proceedings of the Second Arabic Natural Language Processing Conference (ArabicNLP 2024)*, Bangkok. Association for Computational Linguistics.