Large Language Models for Propaganda Span Annotation

Maram Hasanain, Fatema Ahmad, Firoj Alam
Qatar Computing Research Institute, HBKU, Qatar
{mhasanain,fakter,fialam}@hbku.edu.qa

Abstract

The use of propagandistic techniques in online content has increased in recent years aiming to manipulate online audiences. Fine-grained propaganda detection and extraction of textual spans where propaganda techniques are used, are essential for more informed content consumption. Automatic systems targeting the task over lower resourced languages are limited, usually obstructed by lack of large scale training datasets. Our study investigates whether Large Language Models (LLMs), such as GPT-4, can effectively extract propagandistic spans. We further study the potential of employing the model to collect more cost-effective annotations. Finally, we examine the effectiveness of labels provided by GPT-4 in training smaller language models for the task. The experiments are performed over a large-scale in-house manually annotated dataset. The results suggest that providing more annotation context to GPT-4 within prompts improves its performance compared to human annotators. Moreover, when serving as an expert annotator (consolidator), the model provides labels that have higher agreement with expert annotators, and lead to specialized models that achieve state-of-the-art over an unseen Arabic testing set. Finally, our work is the *first* to show the potential of utilizing LLMs to develop annotated datasets for propagandistic spans detection task prompting it with annotations from human annotators with limited expertise. All scripts and annotations will be shared with the community.[1]

1 Introduction

Malicious actors are actively exploiting online platforms to disseminate misleading content for political, social, and economic agendas (Perrin, 2015; Alam et al., 2022a; Sharma et al., 2022). The objective of using propaganda is to generate distorted and often misleading information, which can result in heightened polarization on specific issues and division among communities. Hence, it is important to automatically detect and debunk propagandistic content. The majority of relevant research has focused on either binary or multiclass and multi-label classification scenarios of the task (Barrón-Cedeno et al., 2019; Rashkin et al., 2017; Piskorski et al., 2023b). More recently, interest has shifted to finer-grained propaganda detection at the text span level, which is a multilabel sequence tagging task, where more than one propaganda technique can be used within the same text span (Da San Martino et al., 2019, 2020; Alam et al., 2022b; Przybyła and Kaczyński, 2023; Hasanain et al., 2024b). Such fine-grained analysis is necessary for system explainability and improved digital media literacy among news readers. The task in its nature is complex (Martino et al., 2020), and the complexity is magnified by the large number of propaganda techniques that might be present (18 (Da San Martino et al., 2019) *vs.* 23 (Piskorski et al., 2023b) techniques for example). The subjective nature of the task also results in added challenges.

LLMs showed remarkable capabilities on versatile downstream NLP tasks, and on a plethora of languages, including Arabic (Bang et al., 2023; Ahuja et al., 2023; Abdelali et al., 2024; Liang et al., 2022). However, the utility of LLMs in span-level propaganda detection and categorization remains under-explored. Therefore, we aim to leverage LLMs selecting the highly effective, GPT-4 (OpenAI, 2023), for the task. Moreover, LLMs have shown to be effective aids in creating annotated datasets to train or evaluate other models in a variety of tasks (Alizadeh et al., 2023). Since there are many propaganda techniques to label and a need to create large and diverse datasets to train specialized models, LLMs might benefit the process of developing new datasets for propaganda span detection. Recruiting humans to carry such large-scale annotations has been a very tedious and costly pro-

[1] https://github.com/MaramHasanain/llm_prop_annot

Findings of the Association for Computational Linguistics: EACL 2024, pages 14522–14532

cedure. Our study also aims to investigate whether we could use a LLM, such as GPT-4, to reduce human annotation cost and effort by either reducing the number of annotators, or hiring annotators with less expertise. Finally, to further understand the value of automatic propaganda labeling with LLMs, we employ labels generated by the model under different setups to train specialized language models for the task.

Specifically, we study the following research questions: *(i)* Is GPT-4 capable of annotating propagandistic spans effectively? *(ii)* Can GPT-4 serve both as a general and as an expert annotator of propaganda spans?[2] *(iii)* Which propaganda techniques can GPT-4 annotate best? *(iv)* Can we effectively train specialized models for the task using GPT-4's annotations? Our study makes the following contributions:

- We explore the use of GPT-4 as an annotator for detecting and labeling spans with propagandistic techniques, which is the *first attempt* at such a task. Results reveal the great potential of the model to replace more expert annotators for some propaganda techniques, including those that are highly prevalent in the experimental dataset, such as "Loaded Language". We also provide an in-depth analysis of the model performance at different annotation stages, for more informed adoption of such annotation approach.

- We show that when serving as a consolidator, GPT-4 provides labels that can be effectively used to train a specialized model for the task, achieving state-of-the-art performance on a recently released Arabic dataset from the ArAIEval shared task (Hasanain et al., 2024b). When testing that specialized model on the testing subset from our in-house dataset, it degraded performance by only 13% compared to training the model with the gold labels from the training subset.

- We are releasing all scripts, and annotations from human annotators and GPT-4 to benefit the community.[3]

[2]For this task, the manual annotation process followed generally has two phases: *(i)* annotation done by three *general* annotators, who are less experienced but trained annotators *(ii)* annotations reviewed and disagreements resolved by two expert annotators, referred to as consolidators.

[3]https://github.com/MaramHasanain/llm_prop_annot

2 Related Work

Propaganda Detection. Relevant research has employed diverse methods to identify propagandistic text, ranging from analyzing content based on writing style and readability features in articles (Rashkin et al., 2017; Barrón-Cedeno et al., 2019) to using transformer based models for classification at the binary, multiclass, and multilabel settings (Dimitrov et al., 2021). Recent efforts stress the importance of fine-grained identification of specific propagandistic techniques (Da San Martino et al., 2020). Da San Martino et al. (2019) identified 18 distinct techniques and created a dataset by manually annotating English news articles based on them. Next, they designed a multi-granular deep neural network that extracts propagandistic spans from sentences with a limited F_1=22.58, showing how complex the task is. Piskorski et al. (2023b) extended the 18 techniques into 23 and introduced a dataset in multiple languages. With these efforts, fine-grained propaganda detection in general, and over Arabic content and other lower-resourced languages specifically, still requires further exploration. Existing Arabic datasets are limited in size and number of targeted techniques (Alam et al., 2022b; Hasanain et al., 2023).

LLMs as Annotators. Constructing high-quality annotated datasets, essential for model training and evaluation, usually requires manual annotation by humans (Khurana et al., 2023). There has been efforts in utilizing LLMs for data annotation to overcome the challenges of human annotations, which include bias, time-overhead, and cost (Ding et al., 2023; Alizadeh et al., 2023; Thomas et al., 2023).

Sprenkamp et al. (2023) investigated the effectiveness of LLMs in annotating propaganda by utilizing five variations of GPT-3 and GPT-4. They tackled the task as a multi-label classification problem, using the SemEval-2020 Task 11 dataset. Their findings indicate that GPT-4 achieves results comparable to the current state of the art. Our work is closely related to theirs, however, they approached the problem as a multi-label text classification task of 14 techniques at the article level. In contrast, we focus on fine-grained propaganda detection at the span level including both multilabel and sequence tagging tasks, covering 23 techniques, which is much more challenging.

3 Dataset

Existing Arabic datasets for span-level propaganda detection either lack text span-level annotations (e.g., ArAIEval 2023 shared task dataset (Hasanain et al., 2023)), or cover a more limited set of propaganda techniques (e.g., (Alam et al., 2022b)). [4] Furthermore, to explore the potential of using LLMs as propagandistic spans annotators, a comprehensive dataset with complete human annotations is required as a gold standard.

For this study, an in-house developed dataset is utilized, referred to as ***ArPro*** across this work. It includes a total of **8,000** annotated paragraphs among which, 63% contain at least one propagandistic span. The paragraphs were selected from $2.8K$ news articles, with approximately $10K$ sentences, and around $277K$ words. It covers 14 different topics, with 'news' and 'politics' accounting for over 50% of paragraphs. We split the dataset in a stratified manner (Sechidis et al., 2011), allocating 75%, 8.5%, and 16.5% for training, development, and testing, respectively. We briefly discuss the dataset development process. A complete detail of that process is provided in our recent work (Hasanain et al., 2024a).

The dataset construction started from a large in-house collection of Arabic news articles sourced from over 300 Arabic news media, and including over 600K articles. We sample a set of 2.8K articles following a stratified sampling approach over the news media. Thus, we ensure a versatile set, featuring a variety of writing styles and topics. After automatically parsing the articles, we split them into paragraphs and eliminate ill-formed paragraphs matching any of the following conditions: (i) containing any special character repeated more than three times (e.g., %, *, etc.), (ii) not Arabic as classified by langdetect, [5] and (iii) containing HTML tags. The paragraphs were de-duplicated using Cosine similarity, with a similarity >= 0.75 indicating duplication.

The resulting news paragraphs were then manually annotated using 23 propaganda techniques, adopted from an existing taxonomy (Piskorski et al., 2023a). The annotation process consisted of two phases: *(i)* in phase 1, three **annotators** individually annotated each paragraph, and *(ii)* in phase 2, two expert annotators revised and finalized the annotations. Each annotator in this phase is referred to as a **consolidator**. To facilitate the annotation process, a platform was developed and a comprehensive annotation guideline in the native language (Arabic) was provided to annotators. Additionally, several training iterations were conducted before beginning the annotation task.

Technique	Train	Dev	Test
Appeal to Authority	192	22	42
Appeal to Fear-Prejudice	93	11	21
Appeal to Hypocrisy	82	9	17
Appeal to Popularity	44	4	8
Appeal to Time	52	6	12
Appeal to Values	38	5	9
Causal Oversimplification	289	33	67
Consequential Oversimplification	81	10	19
Conversation Killer	53	6	13
Doubt	227	27	49
Exaggeration-Minimisation	967	113	210
False Dilemma/No Choice	60	6	13
Flag Waving	174	22	41
Guilt by Association	22	2	5
Loaded Language	7,862	856	1670
Name Calling-Labeling	1,526	158	328
Obfuscation-Vagueness-Confusion	562	62	132
Questioning the Reputation	587	58	131
Red Herring	38	4	8
Repetition	123	13	30
Slogans	101	19	24
Straw man	19	2	4
Whataboutism	20	4	4
Total	**13,212**	**1,452**	**2,857**

Table 1: Distribution of the techniques in different data splits at the span level.

The annotation agreement for span-level annotation is $\gamma = 0.546$. This γ agreement metric is specifically designed for span/segment-level annotation tasks, taking into account the span boundaries (i.e., start and end) and their labels (Mathet et al., 2015; Mathet, 2017). Table 1 reports the distribution of the span-level labels across the three dataset splits.

4 Propagandistic Spans Annotation

In this section, we describe our annotation framework including the manual annotation steps used for dataset construction, and the use of GPT-4 for different annotation roles. Figure 1 illustrates this framework. This section also describes a third annotation approach using fine-tuned models.

[4] A large-scale Arabic dataset was released in parallel to this work as part of the ArAIEval 2024 shared task (Hasanain et al., 2024b)

[5] https://pypi.org/project/langdetect/

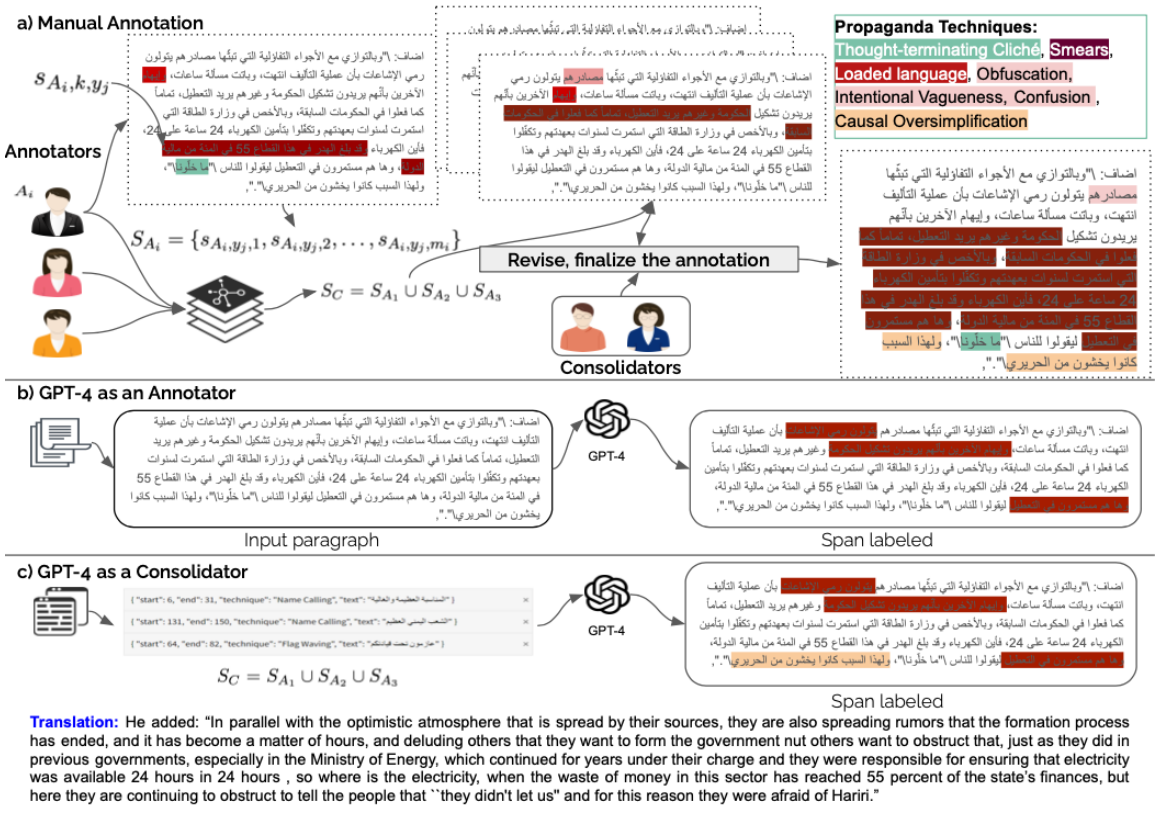

Figure 1: Existing span-level annotation process requiring human annotators and expert consolidators, while our proposed solution uses GPT-4 to support annotation and consolidation.

4.1 Manual Annotation

The manual annotation process went through in two phases. For a given text $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and a label (propaganda techniques) space $\mathcal{Y} = \{y_1, y_2, \ldots, y_o\}$, each annotator A_i provides a set of spans S_{A_i} and each span is represented as $s_{A_i,y_j,k}$, where k is the index of the span for the i-th annotator and y_j is the label. Note that k can range from 1 to the total number of spans identified by annotator A_i, and this total can be different for each annotator. Given this representation, for the i^{th} annotator the set of spans is defined as $S_{A_i} = \{s_{A_i,y_j,1}, s_{A_i,y_j,2}, \ldots, s_{A_i,y_j,m_i}\}$ where m_i is the total number of spans identified by annotator A_i and y_j represents any label from the label space, where j can vary from 1 to o. We combine the spans of all annotators into list $\mathcal{S}_C$ that goes through the consolidation phase to finalize the annotations by consolidators.

To denote the labels (techniques) in a paragraph (input text $\mathbf{x}$) annotated by an annotator A_i, we define the following formulation: $\mathbb{Y}_{A_i} = \bigcup_{j=1}^{p} A_{i,y_j}$ where $\mathbb{Y}_{A_i}$ represents the set of all labels $\{y_1, y_2, \ldots, y_p\}$ annotated by A_i, where p is the total number labels. $\mathbf{Y}$ represents the list of labels from all annotators for a paragraph.

4.2 Annotation with GPT-4

To formally define the problem, let us consider the model $\mathcal{M}$, text input $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, and label space $\mathcal{Y}$. The task of $\mathcal{M}$ is to identify the text span $\mathcal{S} = \{s_1, s_2, \ldots, s_{m_i}\}$ and an associated label for each span s_i, where $s_i = y \in \mathcal{Y}$. The model is conditioned using instruction $\mathcal{I}$, which describes both the task and the label space $\mathcal{Y}$. This conditioning can occur in two scenarios: with a few-shot approach, utilizing labeled examples $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_l$, or in a zero-shot context, where labeled examples are not provided. $\mathcal{D}_l$ represents the labeled dataset. We formulated three levels of difficulty for the propaganda span annotation task using GPT-4.

- **Instruction only (Annotator)**: In this setup, the model is only provided with an instruction $\mathcal{I}$ asking it to annotate the text $\mathbf{x}$ by identifying the propaganda techniques used in it, and then extracting the corresponding spans $\mathcal{S}$.
- **Span extractor (Selector)**: We offer additional information for annotation and frame it as a span extraction problem. The model is asked to select the techniques manifesting in text from the list

$\mathbf{Y}$, and extract the matching text spans.

- **Annotation consolidator (Consolidator)**: This setup is the most resource rich, where the model is asked to act as a consolidator, given list $\mathcal{S}_\mathcal{C}$ as provided by annotators.

4.3 Annotation with PLMs

As a third annotation approach, we aim to train specialized models for the task, using manual and GPT-4 annotations to train a pre-trained language model (PLM). Fine-tuning PLMs, especially those following BERT (Devlin et al., 2019) architecture, has dominated recent approaches for propaganda span detection (Piskorski et al., 2023b; Hasanain et al., 2024b). We model our propaganda span detection and classification task as a span categorization problem, extended from typical token classification tasks like Named Entity Recognition. In this task, multiple labels can be assigned per token, as multiple propaganda techniques can appear as part of the same text span. Formally, we define the task as follows. Given an input token sequence $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ of length n, and a label (propaganda techniques) space $\mathcal{Y} = \{y_1, y_2, \ldots, y_o\}$, the task is to predict Y' of length 23 for each token, with one element for each label. An element y'_i in $\mathcal{Y}'$, is either 0 or 1, indicating whether the token belongs to technique y'_i.

For the model architecture, we select a BERT-based model, and apply a Sigmoid activation function at the output layer of the model, using a binary cross-entropy loss function. To decide whether a token x_i belongs to category y'_i, we set a threshold l, and a model logit $> l$ indicates x_i belongs to y'_i.

5 Experimental Setup

In this section, we describe the setup of the experiments and the evaluation approach followed to investigate the effectiveness of GPT-4 in playing different roles in the annotation process.

5.1 Datasets

Training and analysis: For the main experiments in this study, we used the training subset of ArPro, ArPro$_{train}$, (discussed in Section 3) including 6,002 annotated paragraphs. In particular, we consider *the annotations resulting from the consolidation phase as our gold standard labels in all experiments*.

PLM models training and testing: We train four specialized models, one over each of the following training sets: ArPro$_{train}$, and GPT-4 predicted labels when acting as a consolidator, selector, and an annotator. We evaluate the trained models over two test sets: (i) the test subset of ArPro, ArPro$_{test}$, and (ii) a recent testing subset released with Task 1 of the ArAIEval shared task at the ArabicNLP 2024 conference (Hasanain et al., 2024b). The ArAIEval test subset includes both news paragraphs and tweets, and labeled following the same taxonomy of 23 propaganda techniques we adopt in this work. We chose to test against a second subset, to explore the models robustness and to put the performance of the specialized models, trained over GPT-4 predicted labels, in-context of relevant baseline systems from the shared task.

5.2 Models

LLMs: Across our different experiments, we used zero-shot learning using GPT-4 (32K, version gpt-4-0314, temperature=0) (OpenAI, 2023). We chose this LLM due to its accessibility and superior performance compared to other open and closed models (Ahuja et al., 2023; Abdelali et al., 2024).[6]

PLM: In our experiments in building specialized models for the task, we fine-tune AraBERTv0.2-large (Antoun et al., 2020),[7] which is the most effective Arabic PLM to date over a variety of Arabic NLP tasks (Antoun et al., 2021).[8]

5.3 Instruction

Table 2 lists the exact prompts used to invoke GPT-4 to act in its three different roles of interest in this work. During some pilot studies over the development subset, we have experimented with a variety of prompts for each of the roles before identifying the prompts we eventually used as they had the best performance. We also note that model generally performed really well in responding with the required JSON format of output.

[6] Our initial experiments with another powerful closed model, Claude 3.5 Sonnet, showed that it performs similarly to GPT-4, so we opt to continue with GPT-4.

[7] https://huggingface.co/aubmindlab/bert-large-arabertv02

[8] We have run the same set of experiments with another widely-used Arabic BERT model (Safaya et al., 2020) and observed similar patterns, thus, we only report results using AraBERT in this paper.

Setup	Prompt
Annotator	**Instruction** ($\mathcal{I}$)**:** Label the "Paragraph" by the following propaganda techniques: [techniques list]. Answer exactly and only by returning a list of the matching labels from the aforementioned techniques and specify the start position and end position of the text span matching each technique. Use this template {"technique": , "text": , "start": , "end": } **Paragraph**: {. . . } **Response**:
Selector	**Instruction** ($\mathcal{I}$)**:** Given the following "Paragraph" and "Annotations" showing propaganda techniques potentially in it. Choose the techniques you are most confident appeared in Paragraph from all Annotations and return a Response. Answer exactly and only by returning a list of the matching labels and specify the start position and end position of the text span matching each technique. Use this template Use this template {"technique": , "text": , "start": , "end": } **Paragraph**: {. . . } **Annotations**: $\mathbf{Y}$ **Response**:
Consolidator	**Instruction** ($\mathcal{I}$)**:** Given the following "Paragraph" and "Annotations" showing propaganda techniques potentially in it, and excerpt from the Paragraph where a technique is found. Choose the techniques you are most confident appeared in Paragraph from all Annotations and return a Response. Answer exactly and only by returning a list of the matching annotations. **Paragraph**: {. . . } **Annotations**: $\mathcal{S}_\mathcal{C}$ **Response**:

Table 2: Different prompts used to instruct GPT-4 to annotate input paragraphs by propaganda techniques and spans.

5.4 PLM Fine-tuning

For *each* of the training sets (listed in Section 5.1), we fine-tune the PLM for ep epochs, setting the maximum sequence length to 256, a weight decay of 0.001, a train batch size of 16 and a learning rate of $1e-5$. *For each of the four models we train*, the number of epochs ep and the prediction threshold l (Section 4.3) are hyperparameters we tune over the development subset of ArPro, and report performance of the best model over the testing subset. For hyperparameter tuning, we follow a grid search approach, experimenting with $0.05 <= l <= 0.5$ (step=0.05) and $5 <= ep <= 30$ (step=5).

5.5 Evaluation

We take two approaches to evaluate the performance of models for our tasks.

Standard System Evaluation. For both GPT-4 and fine-tuned models, we computed a modified version of the F_1 measure (macro- and micro-averaged) that accounts for partial matching between the spans across the gold labels and the predictions (Alam et al., 2022b).

Inter-rater Agreement. As we are investigating GPT-4's ability as an annotator, we can also evaluate its performance through the computation of inter-rater agreement between its annotations and the gold labels from the dataset. We specifically computed γ (Mathet et al., 2015; Mathet, 2017), a measure used in similar tasks (Da San Martino et al., 2019), which is designed for span/segment-level annotation tasks.

6 Results and Discussion

To address our research questions, we ran each of the annotation setup prompts (Table 2) over all 6,002 paragraphs in the training split. Table 3 shows the results of evaluating the post-processed model's outputs.

Role	Micro-F_1	Macro-F_1	Span (γ)
Annotator	0.050	0.045	0.247
Selector	0.137	0.144	0.477
Consolidator	0.671	0.570	0.609

Table 3: Performance of GPT-4 (with its different roles) in propaganda span annotation using standard evaluation measures and annotation agreement.

As shown in Table 3, the more information provided to GPT-4 during annotation, the more improvement we observed in its performance. In an information rich setup with GPT-4 as a "consolidator", where we used all the span-level annotations from three annotators, it led to significantly strong model performance. However, it should be noted that the task of a consolidator is not limited to deciding which of the initial annotations are the most accurate. They also had the freedom to modify the annotations by updating the annotation span length or by changing the label for a given span. As for annotation agreement, we can also see that the agreement scores were higher, when more information was provided to GPT-4 in the consolidator role, than the setups with less information.

Incorrect start and end indices. In addition to detecting propaganda techniques, the model was

Role	Micro-F$_{1orig}$	Micro-F$_{1correct}$
Annotator	0.050	0.117
Selector	0.137	0.297
Consolidator	0.671	0.670

Table 4: Performance of GPT-4 with ($_{correct}$) and without ($_{orig}$) span indices correction.

required to provide the text spans matching these techniques (in the "annotator" and "selector" roles). Since a span might occur multiple times in a paragraph, with different context and propagandistic technique, the model should also specify the start and end indices of these spans. We observed that although GPT-4 can correctly provide labels and extract associated text spans, it frequently generated indices not matching the corresponding spans in a paragraph. This lead to mismatch between the start and end indices of spans as compared to gold labels (As Figure 2 shows).

To overcome this problem, we a apply a post-processing step by assigning for each predicted span, the start and end indices of its first occurrence in a paragraph. Table 4 reports the performance of GPT-4 following this correction. It reveals the severity of inaccurate span positions prediction. With the first two roles of the model, we observe the performance increasing by a factor of two with the applied correction. Interestingly, in its third role, as a consolidator, this problem did not manifest, as the model was only selecting annotations, including span and indices, from the list $\mathcal{S}_\mathcal{C}$ of all annotations.

Agreement with consolidators. We delve deeper into the quality of the model's annotations by comparing two values: (a) the agreement of the initial, less-experienced, annotators with the consolidators and (b) GPT-4 agreement with the consolidators (*after start indices correction*). As Figure 3 shows, GPT-4 has notably higher agreement with consolidators compared to initial annotators. It demonstrates a 38% improved agreement when playing the role of a consolidator. *These values demonstrate that GPT-4 achieves comparable or better agreement with the expert consolidators as compared to less experienced human annotators. Moreover, it shows that the model is learning from the given initial annotations to produce improved annotations, closer to the consolidators' performance.*

Per technique performance. Our next research question is: which propaganda techniques can GPT-4 annotate best? We looked at the top five per-technique agreement levels (γ) of the model's labels versus gold labels (Table 5). Over all its roles, the model showed high agreement with expert annotators (consolidators) for three techniques: Doubt, Appeal to Hypocrisy and Loaded Language. It is interesting to see that GPT-4 was highly effective in annotation of the "Doubt" technique, which contradicts with a recent ranking of annotation difficulty of the same taxonomy, derived from humans' performance, in the same task across a multilingual dataset (Stefanovitch and Piskorski, 2023). However, its strong performance with the other two techniques is inline with the aforementioned ranking. The model's ability to annotate "Loaded Language" is particularly useful, as it is the most prevalent technique in the dataset, appearing 7.9K times in the training split under investigation. Replacing human consolidators by GPT-4 to annotate for that technique can save tremendous time and cost. We believe these agreement levels give further evidence of the strong potential of employing GPT-4 as a propaganda span annotator, at least for some techniques. This analysis also provides data needed to inform decisions on which stages of annotation we can inject LLMs like GPT-4.

Technique	Annotator
Causal Oversimplification	0.889
Consequential Oversimplification	0.835
Doubt	0.815
Obfuscation /Vagueness /Confusion	0.791
Appeal to Hypocrisy	0.746
	Selector
Doubt	0.802
Flag Waving	0.705
Appeal to Hypocrisy	0.660
Loaded Language	0.654
Slogans	0.642
	Consolidator
False Dilemma /No Choice	0.872
Loaded Language	0.774
Straw Man	0.697
Doubt	0.695
Name Calling /Labeling	0.680

Table 5: Agreement level (measured by γ) between GPT-4 and gold labels for top five techniques per role, with ($_{correct}$) span indices correction. Underlined are techniques appearing in at least two annotation roles.

Performance of the specialized model. To gain a deeper understanding of the effect of using GPT-4 as an annotator, we use the labels provided by the model in its different annotation roles to train

اضاف: "وبالتوازي مع الأجواء التفاؤلية التي تبثّها مصادرهم يتولون رمي الإشاعات بأن عملية التأليف انتهت، وباتت مسألة ساعات،،،

gold	{"start": 58, "end": 77, "technique": "Loaded_Language", "text": " يتولون رمي الإشاعات"}
predicted	{"start": 82, "end": 101, "technique": "Loaded_Language", "text": " يتولون رمي الإشاعات"}

Figure 2: Example of wrongly generated span indices by GPT-4.

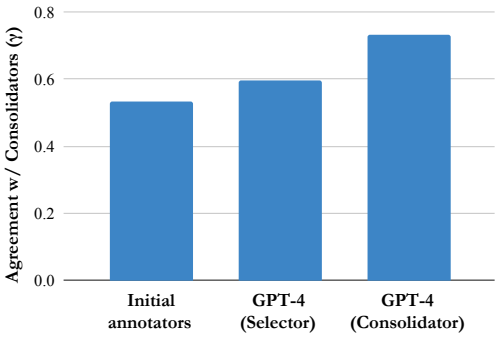

Figure 3: Agreement between consolidators and different types of annotators.

Model	Train Set	ep	l	Micro-F_1
Random	-	-	-	0.010
GPT-4	-	-	-	0.117
AraBERT	GPT-$4_{Annotator}$	20	0.10	0.127
AraBERT	GPT-$4_{Selector}$	25	0.30	0.236
AraBERT	GPT-$4_{Consolidator}$	25	0.15	0.335
AraBERT	ArPro_{train}	30	0.25	0.387

Table 6: Performance of the PLM when fine-tuned on different training sets, and tested on ArPro_{test}. Span indices correction was applied to all GPT-4 predictions. ep: number of training epochs, l: prediction threshold.

specialized models for the task.

Table 6 compares the performance of AraBERT on the ArPro test subset, when fine-tuned with different training sets. We also compare its performance to two baselines: (i) a random baseline, that randomly assigns propaganda techniques to random spans of text in a paragraph (Alam et al., 2022b), and (ii) prompting GPT-4 to predict labels on the test set using the first prompt in Table 2.

Results in Table 6 lead to several conclusions. First, models fine-tuned for the task in all four setups outperform GPT-4 when directly used to detect and label propagandistic spans (2^{nd} row). This motivates the need for specialized models for such complex span categorization task. Second, compared to training the model on the gold labels (6^{th} row), training the model on GPT-4's labels when serving as a consolidator (5^{th} row) reduces performance by only 13%, further supporting our conclusions on the value of using GPT-4 as a consolidator.

Model	Train Set	Micro-F_1
CUET_sstm	-	0.300
AraBERT	GPT-$4_{Annotator}$	0.124
AraBERT	GPT-$4_{Selector}$	0.257
AraBERT	GPT-$4_{Consolidator}$	0.334
AraBERT	ArPro_{train}	0.406

Table 7: Performance of the fine-tuned PLM when tested on $\text{ArAIEval24T1}_{test}$.

We further evaluate the quality of GPT-4 annotations for model fine-tuning, by testing the trained models over a second testing subset, $\text{ArAIEval24T1}_{test}$ (Hasanain et al., 2024b). We compare the models performance to the top performing system from the shared task, CUET_sstm (Labib et al., 2024).

Results in Table 7 endorse using GPT-$4_{Consolidator}$ labels to train specialized models, as it lead to relative improvement over the baseline by 11%. Furthermore, we observe a 35% relative improvement over the top team from the shared task, when we train our model on the ArPro_{train} subset; achieving state-of-the-art for the propaganda span detection and categorization task over this large-scale Arabic testing dataset.

7 Conclusions

In this study, we first investigate GPT-4's ability to play different roles in detecting propagandistic spans and annotating them in Arabic news paragraphs. We investigate if GPT-4 can be used as an annotator when provided with sets of information of varied richness, which represents an increased cost and effort in hiring human annotators. Moreover, we study the value of GPT-4's labels when used to train specialized models for the task. Our experimental results suggest that providing more information significantly improves the model's annotation performance and agreement with human expert consolidators. The study also reveals the great potential of the model to replace consolidators, for some propaganda techniques. Finally, we find that we can train effective models using labels

provided by GPT-4 when acting as a consolidator. We offer an in-depth analysis of the model's performance across various annotation stages, facilitating a more informed adoption of this annotation approach. Future research will explore additional models and learning setups.

8 Limitations

The current version of our work focuses on the analysis and evaluation of GPT-4 specifically limited to Arabic. For this study, we chose to use an Arabic dataset because annotated labels from multiple annotators are available, which are often difficult to obtain. We have evaluated only a closed LLM, as it is currently the most effective model for a large variety of NLP tasks and languages, as reported in a myriad of studies. Moreover, we have ran experiments with large and effective open models for the task, which revealed that they are either unable to understand the task or showed more inferior performance compared to the closed LLM.

Ethics and Broader Impact

We do not foresee any ethical issues in this study. We utilized an in-house dataset consisting of paragraphs curated from various news articles. Our analysis will contribute to the future development of datasets and resources in a cost-effective manner. Human annotators identity will not be shared and cannot be inferred from the annotations we plan to release. We would like to warn users to carefully use the annotations that we plan to release. Its misuse (e.g., using them to generate similar content) may lead to potential risks.

Acknowledgments

The work of M. Hasanain and F. Ahmad is supported by the NPRP grant 14C-0916-210015 from the Qatar National Research Fund part of Qatar Research Development and Innovation Council (QRDI). The findings achieved herein are solely the responsibility of the authors.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, Nizi Nazar, Youssef Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, Majd Hawasly, Nadir Durrani, and Firoj Alam. 2024. LAraBench: Benchmarking Arabic AI with large language models. pages 487–520.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022a. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Preslav Nakov, and Giovanni Da San Martino. 2022b. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop*, WANLP '22, Abu Dhabi, UAE.

Meysam Alizadeh, Maël Kubli, Zeynab Samei, Shirin Dehghani, Juan Diego Bermeo, Maria Korobeynikova, and Fabrizio Gilardi. 2023. Open-source large language models outperform crowd workers and approach chatgpt in text-annotation tasks. *arXiv preprint arXiv:2307.02179*.

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

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of

propaganda techniques in news articles. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, SemEval 2020, Barcelona, Spain.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, EMNLP-IJCNLP 2019, Hong Kong, China.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Maram Hasanain, Fatema Ahmad, 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)*, pages 2724–2744.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Freihat. 2023. ArAIEval shared task: Persuasion techniques and disinformation detection in Arabic text. In *Proceedings of ArabicNLP 2023*, pages 483–493, Singapore (Hybrid). 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)*. Association for Computational Linguistics.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744.

Momtazul Labib, Samia Rahman, Hasan Murad, and Udoy Das. 2024. Cuet_sstm at araieval shared task: Unimodal (text) propagandistic technique detection using transformer-based model. In *The Second Arabic Natural Language Processing Conference (ArabicNLP 2024)*, Bangkok. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. A survey on computational propaganda detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI '20, pages 4826–4832.

Yann Mathet. 2017. The agreement measure γ_{cat} a complement to γ focused on categorization of a continuum. *Computational Linguistics*, 43(3):661–681.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The Unified and Holistic Method Gamma (γ) for Inter-Annotator Agreement Measure and Alignment. *Computational Linguistics*, 41(3):437–479.

OpenAI. 2023. GPT-4 technical report. Technical report, OpenAI.

Andrew Perrin. 2015. Social media usage. *Pew research center*, pages 52–68.

Jakub Piskorski, Nicolas Stefanovitch, Valerie-Anne Bausier, Nicolo Faggiani, Jens Linge, Sopho Kharazi, Nikolaos Nikolaidis, Giulia Teodori, Bertrand De Longueville, Brian Doherty, Jason Gonin, Camelia Ignat, Bonka Kotseva, Eleonora Mantica, Lorena Marcaletti, Enrico Rossi, Alessio Spadaro, Marco Verile, Giovanni Da San Martino, Firoj Alam, and Preslav Nakov. 2023a. News categorization, framing and persuasion techniques: Annotation guidelines. Technical report, European Commission Joint Research Centre, Ispra (Italy).

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023b. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Piotr Przybyła and Konrad Kaczyński. 2023. Where does it end? long named entity recognition for propaganda detection and beyond. In *Proceedings of the*

International Conference of the Spanish Society for Natural Language Processing.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937. Association for Computational Linguistics.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases*, ECML-PKDD '11, pages 145–158, Berlin, Heidelberg. Springer Berlin Heidelberg.

Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022. Detecting and understanding harmful memes: A survey. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, IJCAI '22, pages 5597–5606, Vienna, Austria. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Kilian Sprenkamp, Daniel Gordon Jones, and Liudmila Zavolokina. 2023. Large language models for propaganda detection. *arXiv 2310.06422*.

Nicolas Stefanovitch and Jakub Piskorski. 2023. Holistic inter-annotator agreement and corpus coherence estimation in a large-scale multilingual annotation campaign. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 71–86.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large language models can accurately predict searcher preferences. *arXiv preprint arXiv:2309.10621*.