[1]Bodor Moheel Almotairy

[2]Manal Abdullah

[3]Dimah Hussein Alahmadi

Munir: Weakly Supervised Transformer for Arabic Computational Propaganda Detection on Social Media

JES Journal of Electrical Systems

Abstract: - The intentional manipulation of public opinion has become prevalent in the realm of computational propaganda. The swift spread of misinformation through social networking sites poses significant challenges for governments and society, impacting various aspects of human life. Arab countries are considered among the most affected countries. Accurately identifying and countering computational propaganda is crucial, especially given the impracticality of manually annotating large volumes of social media-generated data. Moreover, the constant propagandists' evolving tactics pose a challenge to accounting models, making the immediate preparation of responsive training data difficult. To address this issue, this research proposes a novel weakly supervised learning approach, leveraging programmatic labeling to label training data in a systematic and timely manner. New labeling functions (LFs) are introduced, where experts' heuristics, knowledge, new proposed lexicons, different fine-tuned pre-trained models are turned into rules to label the data. Leveraging these LFs, we fine-tune a deep learning model for computational propaganda detection. The proposed model achieves a remarkable 94% accuracy and 86% precision in the minority class, outperforming a fine-tuned, fully supervised deep learning model. This research contributes a substantial dataset, a robust weakly supervised model, and lexicons, offering valuable tools for combating computational propaganda on Arabic social media. The code and the dataset are publicly available at https://github.com/Bmalmotairy/Arabic-Propaganda-Detection.

Keywords: Computational propaganda, disinformation, weakly supervised learning, programmatic labeling, deep learning, programmatic weak supervision.

1. Introduction

Propaganda is the major technique through which misinformation and disinformation are spread. It uses psychological and certain rhetorical approaches to appeal to the emotions of the audience and manipulate their opinion. The rise of social media has nourished the phenomenon of "computational propaganda." Computational propaganda, which uses technical means to disseminate information and create propaganda, has made it easier for individuals and groups to spread propaganda on a larger scale [1]. The proliferation of computational propaganda in social media platforms has raised significant concerns about the manipulation of public discourse and the potential influence on political, social, and cultural narratives [2]. Over 81 different nations have been manipulated over social media. Although, Arab countries are among the countries affected by computational propaganda [3], studies on Arab computational propaganda are very rare and need to be highlighted deeply [4].

Computational propaganda comprises a large amount of data moving at different speeds and changing over time. Existing literature has predominantly focused on supervised learning, relying heavily on manually batch-labeled datasets [4]. On the other hand, malicious accounts continuously change their behaviors and techniques, making them easier to escape from the detectors [5]. Moreover, they can profit from the long period of time that is taken to develop new detectors to mess with our online surroundings. Therefore, there is a need to update the detector in a very short time to align the changes. From a technical perspective, recent machine learning models have improved in complexity, power, and automation. Deep Learning (DL) models enable practitioners to obtain state-of-the-art scores on benchmark datasets without using manually selected features. At the same time, the observed trend is that DL models require huge amounts of hand-labeled data to work optimally [6]. In short, DL is data hungry. The expense of labelling the training data is high in terms of both time and money [7]. So, the main barrier to really using DL is the expense of training sets, especially in cases like computational propaganda, where malicious accounts change their strategy constantly.

[1 1*]Department of Information systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. Email: bateekalmutairi@stu.kau.edu.sa

[2]Department of Information systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. Email: maaabdullah@kau.edu.sa

[3]Department of Information systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. Email: dalahmadi@kau.edu.sa

"Weak supervision" refers to a group of machine learning approaches in which models are trained on newly weakly generated training data [8]. It adapts alternative approaches that mix several sources of data to approximate labels instead of having high-quality labeled data manually handled by a subject-matter expert (SME). Labels are regarded as "weak" as they are noisy, inaccurate, and have a margin of error. On the other hand, its strength is that this training newly generated data is more accessible than hand-labeled data [8]. Programmatic Weak Supervision (PWS) is a new software engineering paradigm of weak supervision that makes it possible to generate big, labeled training datasets programmatically in a way that is governable, adaptable, and scalable [9]. It provides a comprehensive framework for weak supervision in which training labels may come from multiple, potentially overlapping sources. The role of the experts has not been canceled, but their expertise has been utilized in an effective manner. Their expertise and heuristics and other supervision sources are encoded in a collection of programming functions called Labeling Functions (LFs). Weak supervision and leveraging PWS provide a promising avenue to overcome training data scarcity challenges. It provides training data in a timely and consistent manner. The LFs can be adjusted and refined quickly to adapt to changing patterns in the data. This approach allows the model to evolve over time as it encounters new instances [9].

The application of weak supervision, such as PWS, in the realm of Arabic computational propaganda detection has not been explored widely yet. The lack of study in this research gap limits the development of robust and scalable models tailored to the unique characteristics of Arabic social media landscapes. This research aims to switch from early to modern methods to combine and encode the cited heuristics and expertise. It makes many significant contributions to the field of computational propaganda detection on social media:

- This study addresses the scarcity of training data for propaganda detection tasks by proposing a novel, weakly supervised model called Munir, leveraging the PWS methods.

- Arabic dataset was introduced to facilitate comprehensive training and evaluation, providing a valuable resource for practitioners and researchers in the domain.

- A specialized model for detecting sarcasm in social media was proposed, enhancing the capabilities of the Munir framework to discern nuanced forms of online communication.

- To further enhance the precision of Munir, we have developed three lexicons that are specifically designed for the unique characteristics of Arabic computational propaganda.

In the subsequent sections, section 2 provides foundational information relevant to weak supervision. The nearly related works are presented in Section 3. The details of the dataset are explained in Section 4, while Section 5 details the methodology and presents experimental results. The research findings are discussed in Section 6. Finally, the research is concluded in Section

2. Background

This section provides foundational information relevant to weak supervision, specifically to PWS. It explains the PWS framework "Snorkel" that underpins the current investigation.

2.1 Programmatic Weak Supervision the New Paradigm

Machine learning developers have turned to less expensive sources of training data. The question is: Could the inputs from these sources be combined in a systematic and abstract manner? This issue was addressed in the innovative frameworks for PWS [10]. PWS is a new software engineering paradigm of weak supervision that makes it possible to generate big, labeled training datasets programmatically. The role of the experts has not been canceled, but their expertise has been utilized in an effective manner. Subject-matter expertise, heuristics, crowd worker labels, patterns, knowledge bases, external data, pre-trained models, and other signals are encoded into programming functions called Labeling Functions (LFs) for data labeling [9]. LFs are user-defined scripts that individually assign labels to a specific part of the data. The LFs often produce noisy labels with a range of error margin. Moreover, these labels may conflict with certain data points. To handle this issue, label models have been developed to aggregate the noisy labels of the LFs [10].

The process of labeling programmatically is considered governable, adaptable, and scalable. Its governability comes from its ability to retain the experts' thoughts in the LFs, unlike manual notation. So, developers can trace back and improve any LF in cases of bias or other undesirable behavior. Its adaptability comes from its ability to add or modify a small, focused number of LFs and then re-execute to relabel the training datasets when data drifts are detected or when the model goals change. Finally, it is scalable since millions of data points can be labeled without further human work after the LFs have been encoded [10].

2.2 **Snorkel, a Data Programming Framework**

Snorkel[2] is a framework that is proposed to programmatically and rapidly label training data. Snorkel is a framework that are proposed in 2016 at Stanford university. The revolutionary concept behind Snorkel was to programmatically and rapidly label, compile, model, manage, and maintain training data. It aims to introduce mathematical and systemic structure to the laborious and sometimes wholly manual process of creating and managing training data [10]. Figure 1 illustrates the Snorkel pipeline with the two main concepts, LFs and the generative model (label model). Snorkel comes equipped with LF analysis methods that allow us to evaluate the performance of the LFs. This feature allows the developers to add, remove, and refine the LFs. Next subsections explain how Snorkel framework works.

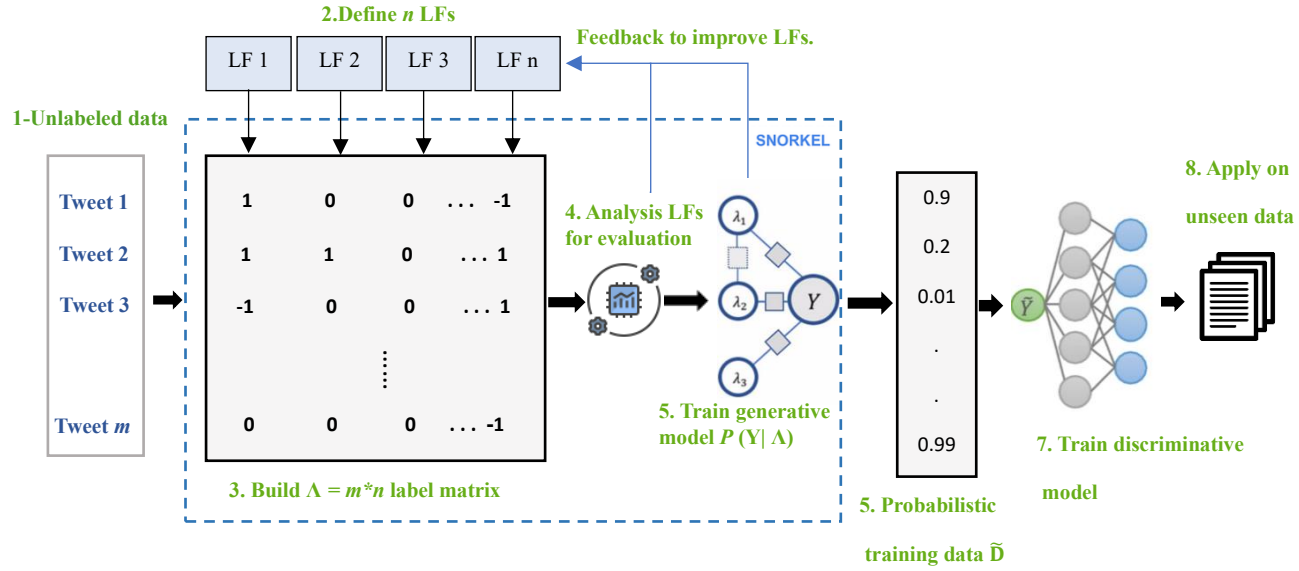

Figure 1 Snorkel Pipeline

2.2.1 **Defining and evaluating the LFs**

As mentioned, LFs are scripts that assign labels to the data point. The easiest method to develop LFs is by analyzing the dataset from different perspectives and identifying its characteristics. This analysis will result in an understanding of different properties that can be used as signals to differentiate each class of the problem at hand. Moreover, leveraging domain experts help create LFs that effectively identify patterns and properties in the data. Based on the extracted properties, LFs can be defined to examine each data point for those properties, and each data point either gets a "vote" for one of the classes or abstains [10]. After applying for the LFs, we must conduct an examination of their performance on a small hand labeled developing set. **LFAnalysis**, a feature of Snorkel, provides a summary of the LFs' performance by measuring evaluation metrics "polarity", "coverage", "overlaps", and "conflicts". **Polarity** represents the unique labels resulted by the LF (omitting abstains). The **coverage** represents the percentage of data that the LFs were able to label (omitting abstains). **Overlaps** indicates the percentage of the data points that has labeled by at least two LF (non-abstaining). **Conflicts** indicate the percentage of the data points in which a LF label (non-abstaining) decision has conflicted with another LF label

[2] https://www.snorkel.org.

(non-abstaining) decision. The developed set helps compute the extra statistics "Correct", "Incorrect," and "Empirical Accuracy". The **correct** and **incorrect** statistics represent the number of data points the LF has labeled correctly or incorrectly. The percentage of data points that have been accurately classified is known as **empirical accuracy** excluding (abstaining).

2.2.2 Modeling Correlations and Accuracies

It is important to keep in mind that users usually create LFs that are statistically dependent. So, modeling these dependencies is essential since they have an impact on true label estimations. ***Snorkel label model*** is the heart of the snorkel framework [10]. It learns the LFs over a generative model to decide on which dependencies to model using an estimator that relies on a hyperparameter. Based on these dependencies, it can produce a final probability-weighted label. The innovative point is that this step does not need any ground-truth data to assess the LFs accuracy. Instead, it depends on probabilistic graphical models to estimate accuracies based on the agreement and disagreements between the LFs. Moreover, it simplifies the development process since it gives helpful feedback on the effectiveness of the LFs [10].

2.2.3 Training a Discriminative Model

The Snorkel generative model primarily reweights the mixture of LFs, producing probabilistic labels but with limited coverage. So, a broad range of cutting-edge machine learning models may learn to generalize beyond the Label model while maintaining this precision, enhancing coverage and resilience on unseen data [10].

2.2.4 Example

For better understanding, assume $x_i \in X$ is piece of text posted on X Platform (Twitter) and $y_i \in Y$ is a label that indicates if this post is propaganda $\boldsymbol{y}$ =1 or non-propaganda $\boldsymbol{y}$ =0. There is no labeled training data, and there is access to a small, labeled dataset that is needed in the development phase, called the development set. Plus, there is a small held-out, blind, labeled test set for evaluation to avoid data leakage [10].

The process starts by defining the $\boldsymbol{n}$ LFs $\lambda = \{\lambda_i, .., \lambda_n\}$. The LFs can be considered a black box, $\lambda: x \rightarrow y \cup \{\emptyset\}$, where the input is a data point $x_i \in X$ and the output is a discrete label $y_i \in Y$ where $Y = \{0, 1\}$ or $\emptyset$ when the LFs abstains. Given $\boldsymbol{m}$ data points in $\boldsymbol{D}$ dataset, Snorkel will apply $\boldsymbol{n}$ LFs and result in ($n * m$) matrix Λ of LFs outputs that contain m candidate labels $\Lambda \in (y \cup \{\emptyset\})^{m \times n}$.

However, for each data point in this matrix Λ, the predicted $\boldsymbol{n}$ labels that were produced from the LFs are conflicted and overlapped. The remaining Snorkel process in the generative model aims to denoise these noisy labels into a single vector of probabilistic training labels $\tilde{Y} = (\tilde{y}_1, ..., \tilde{y}_m)$ where $\tilde{y}_i \in [0, 1]$. Then, the discriminative model will be trained using these training labels.

3. Related work

Despite the pressing need to tackle computational propaganda, there is a notable research gap in exploring weakly supervised models, particularly in the Arabic context. Some literature erroneously treats "fake news" and "propaganda" as synonymous, but they are distinct concepts. Fake news involves false or misleading information presented as news, while propaganda is a broader term encompassing communication techniques intended to influence public opinions [11]. This paper specifically focuses on computational propaganda research within the broader context of misinformation and manipulation.

Related to this research, Leite et al used credibility as a weak signal. They believed that credibility signals, which are a wide range of heuristics typically used by journalists and fact-checkers, could be used to assess the veracity of online content [12]. They proposed a weak supervision method that promotes large language models (LLMs) with a set of 18 credibility signals to produce weak labels for each signal. These potentially noisy labels are then aggregated to predict the veracity of the content. Their approach outperformed state-of-the-art classifiers on two misinformation datasets without using any ground-truth labels for training.

Islam et al focused on the U.S. 2020 presidential elections on Facebook [13]. They proposed a weakly supervised graph embedding-based framework that measured similarity with knowledge to identify the issues and stances of

political ads. The experiment was applied to 0.8 million real-world political ads, achieving 73% accuracy and outperforming two fully supervised models.

Syed et al proposed a DL model to detect fake news resulting from cyber propagation [14]. They proposed a novel hybrid weakly supervised learning method, leveraging SVM to label the data. Then, Bi-GRU and Bi-LSTM were trained on the weak label training data. This approach achieved 90% accuracy.

4. Dataset

The experimentation was performed utilizing datasets from the X platform. It was introduced in our prior research [15]. It focuses on a Saudi propagandist dataset released by X in 2019. The size of the datasets is shown in table 1. There are 2100 tweets that were annotated manually with help of three journalists[3,4,5] guided by expert from the Oxford Internet Institute. [6] Often, labeling errors arise from the annotators themselves, particularly when dealing with intricate concepts that lack clarity and heavily rely on the annotator's comprehension, as in our case. The strength point is that a confident learning technique known as Cleanlab[7] was applied to systematically enhance the label quality. The data was labeled based on its reliability (propaganda and non-propaganda) and the used propaganda techniques. This research defines propaganda techniques as proposed in our prior research [15]. Finally, the data was preprocess using Farasa[8].

Category	Class	Size
Unlabeled datasets	Propaganda	56,000 (53,900)
	Non-propaganda	140,591
Manual annotated dataset	Mix Propaganda and non-propaganda	2100

Table 1 Dataset size and classification

5. Methodology

Figure 2 shows the proposed model, Munir. The process started by proposing candidate LFs. Iteratively, each LF was evaluated in an attempt to improve its efficiency. Based on the evaluation, the LFs selection component selected a portion of LFs to avoid noisy labels. After that, the Snorkel label model labeled the dataset based on the learned dependencies, heuristics, and accuracies. Finally, the final discriminative model was trained to generalize the noisy probabilistic labels. The model was then evaluated and compared with a fully supervised model (FSM) to validate the worthiness of the proposed weakly supervised model. Table 2 shows how the datasets were split to train, develop, and evaluate the weakly supervised model (WSM) and the fully supervised model (FSM).

[3] https://twitter.com/qaburibrahim?s=11&t=c3Ln2hTg674xoeXohmncLw

[4] https://twitter.com/faisalalhmyane?s=11&t=c3Ln2hTg674xoeXohmncLw

[5] https://twitter.com/abdulaziz_ali?s=21&t=VKxoRqdza6UTnrWhF_fgIA

[6] https://www.oii.ox.ac.uk/people/profiles/mona-elswah/

[7] https://github.com/cleanlab/cleanlab

[8] https://farasa.qcri.org/

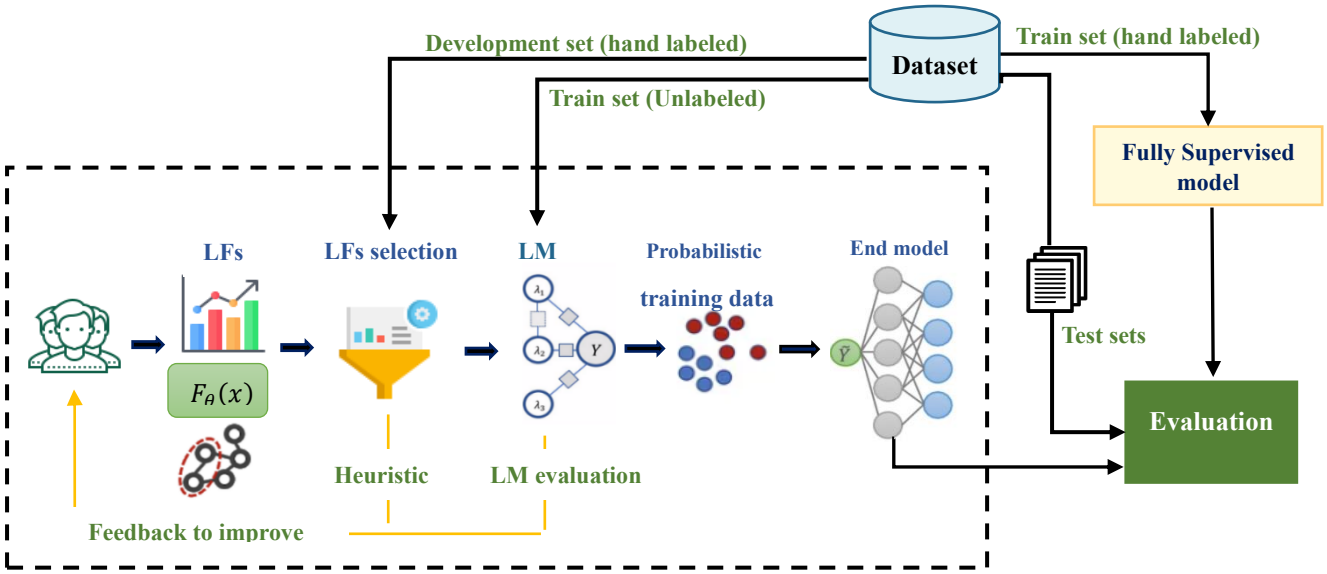

Figure 2 Munir framework

Table 2 The splits of the data

	WSM	FSM
Development	500 labeled tweets	--
Label Model	195671 unlabeled tweets	--
Train	105414 Unlabeled tweets	1260 labeled tweets
Validation	35139 Unlabeled tweets	420 labeled tweets
Test	420 labeled tweets	420 labeled tweets

5.1 Labeling Functions

This section describes the proposed LFs based on conducted exploratory data analysis (EDA) on the dataset, [9] previous research review (LR) and expert heuristics and expertise. Table 3 shows the proposed LFs. There are some LFs that were developed based on the users' characteristics, while others are based on the content characteristics. In cases of unbalanced datasets like ours, it is recommended to split the LFs that produce multiple signals to understand and maximize the accuracy of each class [10]. We followed this behavior and split the LFs that don't have a specific label into two LFs. We also split the lexicon to use only the ones that maximized the label model performance. Finally, we ended up with 50 LFs.

Seven LFs (from LF1 to LF6) were developed based on EDA to discern and establish optimal thresholds to distinguish propaganda. Moreover, new lexicons were suggested with expert help to improve the effectiveness of LFs, LF7, 8, and 9. These LFs label the tweet a propogandist tweet if it contains at least one word from the lexicon. To be more accurate, in each examined tweet, each token, bigram token, lemma, and bigram lemma were compared with the words in the lexicon. The three lexicons can be accessed on the GitHub repository. [10]

[9] https://github.com/Bmalmotairy/Arabic-Propaganda-Detection/blob/main/notebooks/lf_ideas_users.ipynb

[10] https://github.com/Bmalmotairy/Arabic-Propaganda-Detection/blob/main/Lexicons

Regarding LF8, we have taken a benefit from the lexicons proposed by Barrón-Cedeño et al to detect propagandistic content in their proposed model called Proppy [16]. Their lexicon covers different aspects: actives, hedges, implicatives, report verbs, bias, negative words, and positive words. We reviewed and translated all the words in the Proppy lexicon. With the expert's help, synonymous colloquial words were added for each word in the lexicons; for example, "باينه" is the colloquial synonym of "واضح". Regarding positive and negative words, we separated the colloquial lexicon from the classical lexicon because they are expected to be the most used lexicons for propaganda technique name-calling and loaded language techniques, so we want to know how they are used more precisely. For each lexicon, two LFs were proposed: one LF labels the tweet a propogandist tweet if it contains at least one word from the Arabic Proppy lexicon, while the second LF labels the tweet a non-propogandist tweet if it does not contain any word from the Arabic Proppy lexicon. Two more LFs were proposed; they aggregate all lexicons as one lexicon. One of the LFs labels the tweet a propogandist tweet if it contains at least one word from the aggregated Arabic Proppy lexicon, while the second LF labels the tweet a non-propogandist tweet if it does not contain any word from the aggregated Arabic Proppy lexicon. Finally, 10 Proppay LFs were developed.

In LF10, distance similarity metrics are employed to ensure nuanced sensitivity in the labeling process. Distance metrics are utilized in ML to measure similarity between data points [17]. It would be very useful to build on researchers' previous efforts in labeling our dataset, such as measuring the similarity with newly released propaganda datasets such as WANLP 2022. [11] To measure the similarity, we followed the following steps: First: we cleaned the WANLP 2022 using Farasa; our dataset is already cleaned (see Section 4). Next, all the tweets in both datasets were vectorized using FastText[12] with a Skip-gram model [18]. Finally, cosine similarity was used to measure the similarity factors [19].

The LFs (from LF11 to LF18) have been enriched through the integration of Natural Language Processing (NLP) methods, leveraging the capabilities of Stanza[13]. Stanza is an NLP module. This strategic enables the LFs to effectively extract propaganda signals within the analyzed content. Moreover, all the tokenization and lemmatization needed in all the LFs were performed using Stanza. In LF19, 20 and 21, we exploit the capabilities of pre-trained models, such as the zero-shot models (ZSL) [20]. ZSL is a deep learning model that has been trained to generalize on a class of samples. It is usually used when there is no training data. ZSL can employ their inherent knowledge to bolster our labeling mechanism. We applied a ZSL named xlm-roberta-large-xnli[14] as it achieved remarkable results on many cross-lingual benchmarks [21]. To detect hate speech, we applied a multilingual model. [15] The model is developed above a pre-trained XLM-T model for multilingual representation [22].

To detect sarcasm in LF22 and LF23, we fine-tuned the MarBERT transformer model, aligning it with the intricacies of our specific task. Often the efficacy of sarcasm detection models is largely dependent on the quality of the dataset. So, we fine-tuned a version of MarBERT on an ArSarcasm-v2[16] dataset. The dataset contains 15,548 tweets total—12,548 training tweets and 3,000 testing tweets. We utilized a confident learning technique known as Cleanlab to systematically enhance the label quality. Figure 3 shows the steps of cleansing the label and fine-tuning the model. In our case, the model is only intended to be used in a working environment and not benchmarked against the test dataset, so we cleaned up the labeling issues in the training and testing sets by following four steps: first, all the null values, non-Arabic texts, URLs, punctuation marks, whitespace, and new lines were removed. Second, the MarBERT[17] Arabic transformer model was utilized as a text encoder to extract

[11] https://sites.google.com/view/propaganda-detection-in-arabic/home?authuser=0

[12] https://github.com/facebookresearch/fastText

[13] https://stanfordnlp.github.io/stanza/index.html

[14] https://huggingface.co/joeddav/xlm-roberta-large-xnli

[15] https://huggingface.co/Andrazp/multilingual-hate-speech-robacofi

[16] https://github.com/iabufarha/ArSarcasm-v2

[17] https://huggingface.co/UBC-NLP/MARBERT

the input features using the SentenceTransformer[18] framework. Third, a dummy logistic regression model was trained to predict the soft labels (logits) needed for Cleanlab procedures. Shallow models can provide a reasonable baseline performance as they allow us to identify real mislabeled data points without memorizing the small pattern. A K-fold cross-validation technique was used as suggested by [23]. Fourth, the CleanLab object used logits and input features as inputs to identify labeling issues. We used this reasoning repeatedly to detect label errors, correct them, and train the model with the updated, ostensibly higher-quality labels. By eliminating these issues entirely, we obtained a clean dataset including 12037 tweets. After improving the label quality, a pre-trained MarBERT sequence classifier model was trained to classify tweets as sarcasm and non-sarcasm. The dataset was split into a 60% train dataset, a 20% validation dataset, and a 20% test dataset.

To evaluate the model, we trained a logistic regression model as a baseline model on the cleaned data. The transformer model obtained a 77.86 F1-sarcastic on the same held-out test set used for the baseline model. The model's performance exceeded the baseline model by 56.86. It also exceeded the highest F1-sarcastic achieved in the WANLP 2021 shared task on Arabic sarcasm detection leaderboard by 15.61[24]. The transformer model is presented as open-source software (OSS) on the HuggingFace Hub[19].

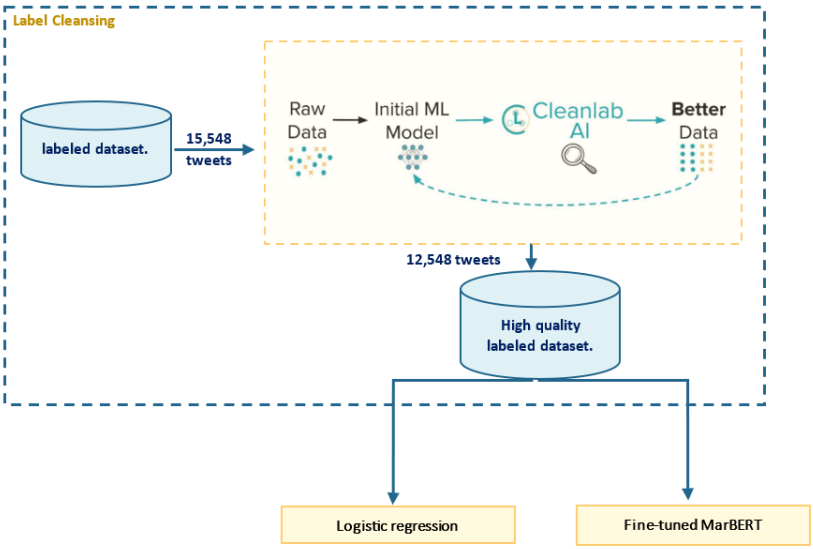

Figure 3 Framework to Improve Dataset Quality to Facilitate Training of the Sarcasm Detection

Table 3 The proposed LFs, P indicates propaganda, PN indicates non-propaganda.

LFs	P	NP	Heuristic	Justification	The used techniques
LF1: Missing bio	✓		Expert/EDA	The propogandist users are automated and usually there are no descriptions on their bios.	Analysis
LF2: Account age	✓	✓	Expert/EDA	The propagandist accounts are likely to be established in the same year as the campaign. Based on the EDA, the propagandist accounts were mostly created in 2018 and 2019.	
LF3: Including special words in the bios		✓	EDA	The non-propogandist tend to use certain words in their bios as shown in Table 4.	

[18] https://www.sbert.net/

[19] https://huggingface.co/Bmalmotairy/marbert-finetuned-wanlp_sarcasm

LF4: Following to follower's ratio	✓	✓	EDA	The ratio is between 0 and 0.2 for propagandists and 0.8 and 1 for non-propogandists.	
LF5: Including URLs		✓	EDA	URLs appear in 87 % of non-propagandist tweets.	
LF6: Including mentions	✓		Expert\LR	Propaganda' tweets typically include mentions of the target person or anyone they want to draw audience attention to his claims [25].	
LF7: Reductio ad Hitlerum lexicons	✓		Expert	A reductio ad hitlerum technique happens when propagandists try to persuade a target audience to refuse an idea because it is adopted by repugnant groups.	Compare with the proposed Lexicon
LF8: Proppay lexicons (10 LFs)	✓	✓	Expert\LR	Take a benefit from the previous lexicons in the field [16]	
LF9: Loaded language lexicons.	✓	✓	Expert\LR	In loaded language technique, the propagandists try to influence the target audience by using strong words and phrases that have significant emotional connotations (either good or negative) [26] .	
LF10: Distant supervision	✓	✓	LR	The tweet is considered a propogandist tweet if it's 90% like at least one of the WANLP 2022 propogandist tweets, and the tweet is considered a non-propogandist tweet if it is 60% similar to at least one of the WANLP 2022 non-propagandist tweets.	Cosine similarity
LF11: Missing location	✓		Expert	The propagandists' accounts typically have a sketchy location. For example, I live on earth.	NER (entities tagging)
LF12: Including entities	✓	✓	Expert	The propaganda techniques, such as loaded language, name-calling, exaggeration or minimization, and smears, may be directed at specific entities, either persons or locations.	
LF13: Including pronouns.	✓		LR	Propaganda materials tend to use pronouns [27].	POS (pronouns tagging)
LF14: Flag waving technique	✓		Expert\LR	In flag waving, propagandists use strong national feelings or any group feelings (such as gender or race) to promote an idea, for example, "our country" ("وطننا"). Often, propagandists tend to use plural pronouns like "we" and "our" to create a sense of unity and include the target audience in a collective identity [25].	
LF15: Exaggeration technique	✓		Expert	An exaggeration technique occurs when propagandists try to amplify an idea or a person in an excessive manner. The experts stated that exaggerated statements usually include the "أفعل" preference form, such as (الأجمل) (the most beautiful). So, we must look for the adjective words where it's the lemma started with (أ).	POS (adjective tagging)

LF16: Doubt technique	✓		Expert	Doubt techniques come in different forms, but usually propagandists tend to ask skeptical questions.	POS (interrogative tagging)
LF17: Slogans technique	✓		Expert	A slogan is a succinct and striking sentence that could include stereotypes and labels. Table 4 shows the most used phrase to detect slogans.	Tokenization
LF18: Repetition technique	✓		Expert	Propagandists sometimes repeatedly use the same term in a statement to create a feeling of urgency.	
LF19: Knowledge from pre-trained models	✓	✓	Expert	Pre-trained models can be utilized as LFs to provide weak labels [28].	ZSL
LF20: Loaded hate language	✓	✓	Expert	Propaganda may destroy democracies by encouraging hate propaganda, whether the target is present or hidden.	Multilingual model
LF21: Hate speech and entity	✓		Expert	Hate speech is used as a negative label in name-calling	Multilingual models + NER (entities tagging)
LF22: Loaded sarcasm language	✓	✓	Expert\LR	Sarcasm is used to support and strengthen propaganda techniques [29], whether the target is present or hidden.	The fine-tuned MarBERT
LF23: Sarcasm and entity	✓		Expert\LR	A name-calling propaganda technique is giving an entity (someone or something) a negative label that is easy to remember [25]. Sarcasm is usually used as a negative label in name-calling [29].	The fine-tuned MarBERT + NER (entities tagging)

Table 4 the common using words.

No.	The words usage	Arabic word	Translation to English
1	In the non-propogandist profiles	الحساب الرسمي	Official account
2		عضو	member
3		رئيس	President, Manger
4		كاتب	Author
5		إدارة	Management
6	In the slogan technique	لا لـ	No to
7		لا بديل لـ	There is no alternative to
8		نعم لـ	Yes to

5.2 LFs Evaluation

The LFs are evaluated using the evaluation metrics "coverage," "overlaps," and "conflicts," as explained in section 2.2.1. Moreover, we already have a small sample of labeled data, which can serve as ground truth labels. So, this

data was used as a development set, which added three more metrics to evaluate the LFS: "correct," "incorrect," and “empirical accuracy." The parameters of the label model (generative model) are learned when there are enough sources of better-than-random supervision. Therefore, it is necessary to select the most influential LFs. With enough signal to estimate the latent class labels better than random guessing, those estimates can be refined until the model is identified.

After evaluating the proposed 50 LFs we found that some LFs have very high accuracy, but at the same time, they have too low coverage. On the other hand, some of the LFs have high coverage but low accuracy. Having low coverage does not mean eliminating the LF, as it may have high observed accuracy in labeling its class during training phase. The objective is to strike a balance so that we can optimize coverage and accuracy. To manage the trade-off between accuracy and coverage, the Snorkel team recommended setting confidence thresholds for each LF and only accepting those that are greater than the thresholds. It is also recommended by the Snorkel team to use only the LFs that we are sure have at least a 50% precision score. Also, it is recommended but not required to use all the LFs that optimize the label model performance regardless of their precision score on the development dataset, as long as it exceeds a threshold of 20% [10].

We set a confidence threshold to accept any LF that separately covers less than 50% and has greater than or equal to 35% accuracy. Our interpretation is that we tried to eliminate the LFs that are suspected of providing a majority-class signal rather than a correct one. Plus, we assumed that the LF that gives acceptable accuracy (above > 20) and can identify the propaganda correctly in the development dataset can be certain to be generalized well to the training data. This optimal accuracy threshold was determined based on the label model's accuracy after several iterations. Finally, eight LFs were chosen. They are labeling_sarcasm, loaded_language, loaded_sarcasm, distant_supervision_prop, distant_supervision_gen, reductio, xlmroberta_prop, and xlmroberta_gen. Table 5 shows the evaluation metrices of the selected LFs. All the LFs results can be accessed in the project GitHub repository[20].

Table 5 LFs evaluation using Snorkel LFAnalysis

LF	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
labeling_sarcasm	[1]	0.006	0.006	0.006	2	1	0.666667
loaded_language	[1]	0.052	0.052	0.052	17	9	0.653846
loaded_sarcasm	[1]	0.008	0.008	0.008	3	1	0.75
distant_supervision_prop	[1]	0.008	0.008	0.008	3	1	0.75
distant_supervision_gen	[0]	0.24	0.24	0.238	111	9	0.925
reductio	[1]	0.012	0.012	0.012	4	2	0.666667
xlmroberta_prop	[1]	0.022	0.022	0.022	4	7	0.363636
xlmroberta_gen	[0]	0.454	0.454	0.45	208	19	0.9163
genuine_proppy_positive_colloquial_words	[0]	0.644	0.644	0.632	296	26	0.919255

5.3 Label Model

In order to aggregate the LFs’ votes, a probabilistic graphical model was trained to learn the accuracies and correlation dependencies between the LFs and the true (hidden) label. The label model was trained using a constant learning rate scheduler and an Adam optimizer with a 0.05 warmup ratio. The number of training epochs picked was 2000 to ensure convergence. The L2 regularization parameters were also fine-tuned.

The label model was applied to the entire unlabeled dataset, which includes 195671 unlabeled tweets, achieving 90% accuracy base on the validation data. Table 6 shows the performance metrics of the label model for each class. The label model covered about 72% of the data, which represents 140,883 tweets. The label model labeled 18.01% of the unlabeled tweets as propaganda, 53.8% as non-propaganda, and abstained from labeling 28.7% of

[20] https://github.com/Bmalmotairy/Arabic-Propaganda-Detection/blob/main/notebooks/labeling_functions.ipynb

the unlabeled data. Finally, we obtained a training dataset with 75% non-propaganda and 25% propaganda. he resulting weakly labeled dataset is accessible online in the project GitHub repository.[21]

5.4 Weakly Supervised Model (WSM).

Since the role of the label model is to produce weakly labeled training data, we need to train an end-to-end discriminative model to generalize over the data. The end model was trained using a noise-aware objective (loss) function which is Active-Passive Losses[22]. AraBERT[23] version 2 model was picked as autoencoders because they have been adapted to accept processed text from Farasa, which we used as the text processing tool throughout the project. The weakly labeled data resulting from the label model (140,883 tweets) was split into a 75% train set and a 25% validation set. The model was tested on 420 tweets from our ground truth data. It is worthwhile to note that we did not validate the model in the development dataset to prevent data leakage. In the training, we only train the model for one epoch. This is a very important and tricky point, as the data is weakly labeled and has an error margin, so increasing training time leads to learning the error patterns. As a result, the WSM achieved 94% accuracy, 93% weighted-average F1, and 78% macro-averaged F1 scores. Figure 4 shows the confusion matrix. The model is presented as open-source software (OSS) on Hugging Face. [24]

Table 6 Label model performance

	Precision	Recall	F1-score
Non- propaganda	0.96	0.93	0.94
Propaganda	0.58	0.7	0.63
Accuracy	0.9		

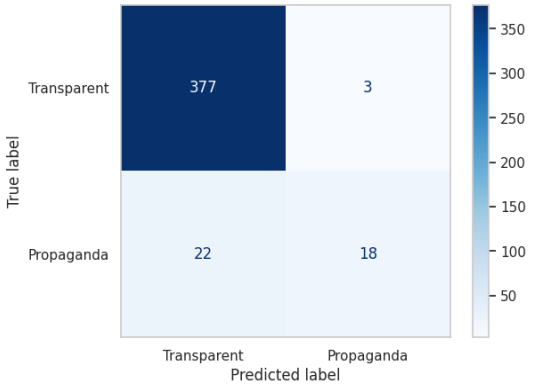

Figure 4 The WSM confusion matrix

5.5 Evaluation

To validate the worthiness of the WSM, a FSM was trained, validated, and tested on the labeled dataset (ground truth). The data was split into 60% for training, 20% for validation, and 20% for testing. The test data set is the same test set used to test the WSM. For the main FSM we used the AraBERT version family autoencoders as we did in the WSM. Again, the model was trained using a weighted cross entropy loss to overcome the class imbalance. Figure shows the training results, the best model was loaded at epoch 3. The FSM achieved 90%

[21] https://github.com/Bmalmotairy/Arabic-Propaganda-Detection/tree/main/Data

[22] https://github.com/HanxunH/Active-Passive-Losses

[23] https://huggingface.co/aubmindlab/bert-base-arabertv2

[24] https://huggingface.co/Bmalmotairy/arabertv2-weakly-supervised-arabic-propaganda

accuracy, 91% weighted-average F1, and 78% macro-averaged F1 scores. Figure 5 shows the training result, and figure 6 shows the confusion matrix. The model is presented as open-source software (OSS) on Hugging Face.[25]

Figure 5 The FSM training results.

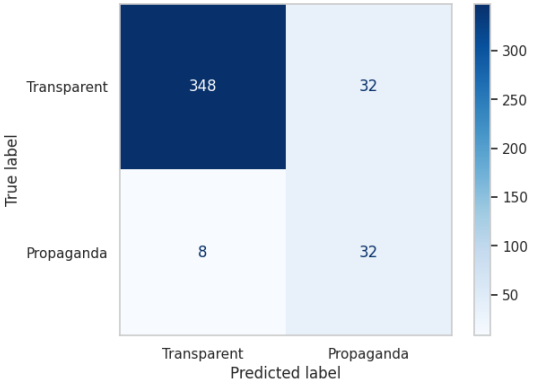

Figure 6 The FSM confusion matrix

Table 7 compares the evaluation metrics of the FSM and the WSM [30]. Comparing the two classifiers, the WSM surpassed the FSM in the non-propaganda (majority) class by 2% in the F1-score and by 7% in the recall. The WSM can classify 99% of the non-propaganda tweets correctly, despite the high similarity between the normal posts and the propagandist posts. Regarding the propaganda (minority) class, the WSM surpassed the FSM by 36% in precision, having 86%. In computational propaganda detection, precision is considered a very important metric to measure the model's performance in detecting the minority class (propaganda). We need to be sure that the posts that were assigned as propaganda are really propaganda, as it affects the users' reliability. Overall, the WSM was surpassed by 4 % in accuracy and by 2 % in the weighted average.

Table 7 Performance metric comparison between FSM and WSM.

Class	**Metrics**	**FSM**	**WSM**	**Δ**
Non-propaganda	F1- score	0.95	0.97	+0.02
	Precision	0.98	0.94	-0.04
	Recall	0.92	0.99	+0.07
Propaganda	F1-score	0.62	0.59	-0.03
	Precision	0.50	0.86	+0.36
	Recall	0.80	0.45	-0.35
Both classes	Accuracy	0.90	0.94	+0.04
Both classes	Weighted avg	0.91	0.93	+0.02

[25] https://huggingface.co/Bmalmotairy/arabertv2-fully-supervised-arabic-propaganda

6. Discussion

The issue of a lack of training data can be solved successfully and flexibly with weak supervision. In this research we adopt PWS to generate big, labeled training datasets programmatically in a way that is governable, adaptable, and scalable. We proposed 50 LFs in an attempt to cover all the task perspectives. Subsequently, a threshold was established to choose the most suitable labeling functions (LFs) that enhanced the performance of the label model. This process yielded a total of eight LFs. We have observed that all the LFs that increase the label model performance are related to propaganda-loaded language techniques, reductio ad Hitlerum techniques, sarcasm, pretrained models, and distance supervision. These results support previous research, which proved that loud language is the most important technique used in propaganda [25]. One noteworthy finding is the significance of sarcasm as a key labeling function (LF) contributing to detecting computational propaganda. Even though sarcasm and propaganda are distinct concepts, in certain situations, they may intersect when sarcasm is employed within propaganda to add a persuasive or emotive element to the messaging.

Using a pre-trained model, such as a zero-shot model, as a LF to estimate the latent label is advantageous due to its ability to leverage the model's generalization capabilities across novel tasks. However, we have to keep in mind the limitations and biases of the pre-trained model and consider incorporating other sources of weak supervision to improve the overall quality of the labeling process. But in any case, continuous training of such models is very useful for detecting propaganda. Distance supervision approves its efficiency in propaganda detection by utilizing pre-existing labeled datasets to extend labels to a broader range of unlabeled tweets. We have to note that the effectiveness of this approach depends on the quality and representativeness of the datasets used for distance supervision. Although previous studies stated that propaganda may contain offensive language, our results suggest hate speech is characterized by offensive or discriminatory language, whereas propaganda often involves the dissemination of information with a specific intent to shape public opinion, influence beliefs, or promote a particular agenda. So, the linguistic patterns and cues in hate speech may not align with those in propaganda.

None of the LFs related to the users' characteristics play a vital role in estimating the latent labels. This proves the extent to which propagandists excel at imitating reliable accounts to hide their identities The lexicons provided by experts, including the loaded language lexicon and the reductio ad hitlerum lexicon, enhanced Munir's performance as they provided a more refined comprehension of the context of Arabic computational propaganda. At the same time, none of the LFs of the Proppay lexicons provide signals to distinguish between propaganda and non-propaganda, although they provide a good result in the English context[16]. The rationale behind this observation assures the influence of linguistic and cultural disparities which should be considered carefully in the realm of Arabic computational. Moreover, these lexicons frequently serve as general linguistic features, giving insights on the style or tone of the text. But they might not be detailed enough to discern between propaganda and non-propaganda content.

Based on our experiments' results, the WSM's accuracy outperforms the FSM by 4%, and its precision in the minority class outperforms by 36%. At the same time, there was no need to bear the annotation costs. Weak supervision, including programmatic labeling, approves its ability to be a powerful approach for annotating datasets, particularly in tasks like propaganda detection where propaganda techniques are dynamic. Programmatic labeling allows for flexibility in adapting to these changes. If there is a shift in the propagandist's technique, the weak supervision sources can be updated or modified without the need for manual reannotation. This flexibility is essential to a propaganda detection system's long-term viability.

7. Conclusion, Limitation, and Future Work

This work proposes a weakly supervised learning model to detect Arabic computational propaganda. The proposed model achieved a remarkable 94% accuracy, outperforming the fully supervised model by 36% in the minority class without any need to train a dataset. This research contributes a substantial dataset, a robust, weakly supervised model, and lexicons. The contributions together make up a complete framework to detect Arabic computational propaganda and provide valuable resources for researchers and practitioners working, particularly in the linguistic context of Arabic social media. No study is perfect; this study has some limitations. First, the

dataset is limited to X data exclusively from Saudi Arabia. Hence, preliminary testing is necessary in order to apply the model to X data originating from Arab nations. Second, the hand-annotated data includes 2100 tweets and covers 15 propaganda techniques out of 20. The scope and size of the hand-labeled data were constrained by available resources. Our future aim is to enhance Munir capabilities by including the detection of many modalities, such as text, photos, and videos.

References

[1] G. Bolsover and P. Howard, "Computational Propaganda and Political Big Data: Moving Toward a More Critical Research Agenda," *Big Data*, vol. 5, no. 4, pp. 273–276, Dec. 2017, doi: 10.1089/big.2017.29024.cpr.

[2] D. S. Pfister, "The Logos of the Blogosphere: Flooding the Zone, Invention, and Attention in the Lott Imbroglio," *Argumentation and Advocacy*, vol. 47, no. 3, pp. 141–162, Jan. 2011, doi: 10.1080/00028533.2011.11821743.

[3] S. Bradshaw, H. Bailey, and P. Howard, "Computational Propaganda | Industrialized Disinformation: 2020 Global Inventory of Organized Social Media Manipulation," 2020. Accessed: Feb. 11, 2021. [Online]. Available: https://comprop.oii.ox.ac.uk/research/posts/industrialized-disinformation/

[4] G. D. S. Martino, S. Cresci, A. Barron-Cedeno, S. Yu, R. Di Pietro, and P. Nakov, "A Survey on Computational Propaganda Detection," in *IJCAI International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Jul. 2020, pp. 4826–4832. Accessed: Feb. 10, 2021. [Online]. Available: http://arxiv.org/abs/2007.08024

[5] S. Cresci, A. Spognardi, M. Petrocchi, M. Tesconi, and R. di Pietro, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in *26th International World Wide Web Conference 2017, WWW 2017 Companion*, New York, New York, USA: International World Wide Web Conferences Steering Committee, 2019, pp. 963–972. doi: 10.1145/3041021.3055135.

[6] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput Sci*, vol. 2, no. 6, pp. 1–20, Nov. 2021, doi: 10.1007/S42979-021-00815-1/FIGURES/13.

[7] J. Mustafi, "Natural Language Processing and Machine Learning for Big Data," in *Techniques and Environments for Big Data Analysis*, Springer, Cham, 2016, pp. 53–74. doi: 10.1007/978-3-319-27520-8_4.

[8] R. Poyiadzi, D. Bacaicoa-Barber, J. Cid-Sueiro, M. Perello-Nieto, P. Flach, and R. Santos-Rodriguez, "The Weak Supervision Landscape," *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom Workshops 2022*, pp. 218–223, 2022, doi: 10.1109/PERCOMWORKSHOPS53856.2022.9767420.

[9] J. Zhang, C.-Y. Hsieh, Y. Yu, C. Zhang, and A. Ratner, "A Survey on Programmatic Weak Supervision," Feb. 2022, Accessed: Jul. 16, 2023. [Online]. Available: https://arxiv.org/abs/2202.05433v2

[10] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: rapid training data creation with weak supervision," *VLDB Journal*, vol. 29, no. 2–3, pp. 709–730, May 2020, doi: 10.1007/S00778-019-00552-1/FIGURES/15.

[11] I. Vamanu, "Fake News and Propaganda: A Critical Discourse Research Perspective," *Open Information Science*, vol. 3, no. 1, pp. 197–208, Jan. 2019, doi: 10.1515/OPIS-2019-0014/MACHINEREADABLECITATION/RIS.

[12] J. A. Leite, O. Razuvayevskaya, K. Bontcheva, and C. Scarton, "Detecting Misinformation with LLM-Predicted Credibility Signals and Weak Supervision," Sep. 2023, Accessed: Jan. 17, 2024. [Online]. Available: https://arxiv.org/abs/2309.07601v1

[13] T. Islam, S. Roy, and D. Goldwasser, "Weakly Supervised Learning for Analyzing Political Campaigns on Facebook," in *Proceedings of the International AAAI Conference on Web and Social Media*, Association for the Advancement of Artificial Intelligence (AAAI), Jun. 2023, pp. 411–422. doi: 10.1609/ICWSM.V17I1.22156.

[14] L. Syed, A. Alsaeedi, L. A. Alhuri, and H. R. Aljohani, "Hybrid weakly supervised learning with deep learning technique for detection of fake news from cyber propaganda," *Array*, vol. 19, p. 100309, Sep. 2023, doi: 10.1016/J.ARRAY.2023.100309.

[15] B. M. Almotairy, M. Abdullah, and D. H. Alahmadi, "Dataset for Detecting and Characterizing Arab Computation Propaganda on X," *Data Brief*, p. 110089, Jan. 2024, doi: 10.1016/J.DIB.2024.110089

[16] A. Barrón-Cedeño, I. Jaradat, G. Da San Martino, and P. Nakov, "Proppy: Organizing the news based on their propagandistic content," *Inf Process Manag*, vol. 56, no. 5, pp. 1849–1864, Sep. 2019, doi: 10.1016/j.ipm.2019.03.005.

[17] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," *ACL and AFNLP*, pp. 1003–1011, 2009, Accessed: Jul. 15, 2023. [Online]. Available: https://aclanthology.org/P09-1113

[18] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans Assoc Comput Linguist*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[19] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8206 LNCS, pp. 611–618, 2013, doi: 10.1007/978-3-642-41278-3_74/COVER.

[20] F. Pourpanah *et al.*, "A Review of Generalized Zero-Shot Learning Methods," *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 4, pp. 4051–4070, Apr. 2023, doi: 10.1109/TPAMI.2022.3191696.

[21] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Nov. 2019, doi: 10.18653/v1/2020.acl-main.747.

[22] F. Barbieri, L. E. Anke, and J. Camacho-Collados, "XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond," *2022 Language Resources and Evaluation Conference, LREC 2022*, pp. 258–266, Apr. 2021, Accessed: Dec. 19, 2023. [Online]. Available: https://arxiv.org/abs/2104.12250v2

[23] P. Chen, B. Liao, G. Chen, and S. Zhang, "Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 1833–1841, May 2019, Accessed: Oct. 07, 2023. [Online]. Available: https://arxiv.org/abs/1905.05040v1

[24] I. A. Farha, W. Zaghouani, and W. Magdy, "Overview of the WANLP 2021 Shared Task on Sarcasm and Sentiment Detection in Arabic," in *the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Virtual): Association for Computational Linguistics, 2021, pp. 296–305. Accessed: Oct. 24, 2023. [Online]. Available: https://aclanthology.org/2021.wanlp-1.36

[25] H. Thayer, "21st Century Propaganda: The Age of Twitter," Pace University, 2018.

[26] J. Li, Z. Ye, and L. Xiao, "Detection of Propaganda Using Logistic Regression," in *Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, Association for Computational Linguistics (ACL), Nov. 2019, pp. 119–124. doi: 10.18653/V1/D19-5017.

[27] A. Maritz, "Propaganda language: Quantifiers and pronouns," *Tydskrif vir Geesteswetenskappe*, vol. 61, no. 4–1, pp. 1057–1078, Dec. 2021, doi: 10.17159/2224-7912/2021/V61N4-1A6.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans Knowl Data Eng*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.

[29] M. Sandhu, C. D. Vinson, V. K. Mago, and P. J. Giabbanelli, "From associations to sarcasm: Mining the shift of opinions regarding the Supreme Court on twitter," *Online Soc Netw Media*, vol. 14, p. 100054, Nov. 2019, doi: 10.1016/J.OSNEM.2019.100054.

[30] M. Steurer, R. J. Hill, and N. Pfeifer, "Metrics for evaluating the performance of machine learning based automated valuation models," *Journal of Property Research*, vol. 38, no. 2, pp. 99–129, Apr. 2021, doi: 10.1080/09599916.2020.1858937.