Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-024-00852-1

RESEARCH

Early detection of fake news on emerging topics through weak supervision

Serhat Hakki Akdag[1] · Nihan Kesim Cicekli[1]

Received: 17 October 2023 / Revised: 26 February 2024 / Accepted: 27 February 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
In this paper, we present a methodology for the early detection of fake news on emerging topics through the innovative application of weak supervision. Traditional techniques for fake news detection often rely on fact-checkers or supervised learning with labeled data, which is not readily available for emerging topics. To address this, we introduce the Weakly Supervised Text Classification framework (WeSTeC), an end-to-end solution designed to programmatically label large-scale text datasets within specific domains and train supervised text classifiers using the assigned labels. The proposed framework automatically generates labeling functions through multiple weak labeling strategies and eliminates underperforming ones. Labels assigned through the generated labeling functions are then used to fine-tune a pre-trained RoBERTa classifier for fake news detection. By using a weakly labeled dataset, which contains fake news related to the emerging topic, the trained fake news detection model becomes specialized for the topic under consideration. We explore both semi-supervision and domain adaptation setups, utilizing small amounts of labeled data and labeled data from other domains, respectively. The fake news classification model generated by the proposed framework excels when compared with all baselines in both setups. In addition, when compared to its fully supervised counterpart, our fake news detection model trained through weak labels achieves accuracy within 1%, emphasizing the robustness of the proposed framework's weak labeling capabilities.

Keywords Fake news detection · Weakly supervised learning · Text classification · Language models

1 Introduction

Fake news, defined as fabricated information that mimics news media content, poses a threat to the reliability of information online, risking the integrity of public discussion and opin-

✉ Nihan Kesim Cicekli
nihan@ceng.metu.edu.tr

Serhat Hakki Akdag
serhat.akdag@outlook.com

[1] Department of Computer Engineering, Middle East Technical University, Ankara 06800, Turkey

Published online: 15 March 2024

ion (Lazer et al., 2018). Protecting online communities from misinformation is crucial for maintaining a healthy information environment. Common strategies for detecting fake news include employing fact-checkers, credible individuals or organizations that manually verify news stories, and utilizing supervised machine learning on labeled datasets to train models (Zhou & Zafarani, 2020). However, these techniques face challenges in the early detection of fake news on emerging topics, as fact-checkers may be slow to investigate new claims, and labeled datasets are often unavailable during the early stages of dissemination. To address this issue, we aim to explore alternative approaches in this paper, focusing on the early detection of fake news on emerging topics using weakly supervised learning.

We introduce the Weakly Supervised Text Classification (**WeSTeC**) framework, to programmatically label a large-scale text dataset in a particular domain and train supervised classifiers with the assigned labels. The generated labeling functions are then applied to annotate instances within extensive datasets. Resulting annotations are aggregated into a single weak label per data instance, without having access to ground-truth labels. In the end, the weakly labeled large-scale dataset is used to train supervised text classification models. The proposed framework not only introduces a novel way to generate, eliminate, and apply labeling functions but also executes the aforementioned steps end-to-end, filling an important gap in the programmatic weakly supervised text classification landscape.

We utilize the developed framework in two different fake news classification setups. The first one is the semi-supervision setup, where there is only a limited number of labeled news articles within the same domain as the target large-scale dataset, referencing an emerging topic. We experiment with a setup in which the number of labeled data instances is less than 0.7% of the unlabeled large-scale dataset. The second setup that we experiment with is the domain adaptation setup where we have labeled data for fake news articles in a specific domain. Here, we aim to programmatically assign labels to a large-scale dataset of a different domain, containing news on an emerging topic. The generic and end-to-end nature of WeSTeC allows us to seamlessly test both setups by merely changing the inputs to the framework.

The contributions of this study can be categorized under two headings. (1) Many studies addressing the early detection of fake news rely on a static set of labeling functions, which are used to label the dataset at hand (Li et al., 2021; Leite et al., 2023). The manual generation of labeling functions is a time-consuming process, resulting in a smaller overall number of labeling functions. Furthermore, most of the studies we encountered only consider a single strategy to generate labeling functions, such as content features or social context (Özgöbek et al., 2022; Shu et al., 2020). Our work demonstrates the use of two distinct approaches, namely content-based and model-based, to automatically generate labeling functions and eliminate under-performing ones. With a greater number of quality labeling functions and the capability to benefit from two different strategies, our fake news detection models demonstrate superior performance compared to state-of-the-art weakly supervised fake news detection algorithms. (2) We introduce a novel framework, WeSTeC, capable of executing programmatic weakly supervised text classification tasks end-to-end. In this study, the framework is used for fake news classification tasks specifically, however, it is designed to handle various text classification tasks. Thanks to the fully automated and end-to-end nature of the framework, it can seamlessly run both semi-supervision and domain adaptation pipelines by only changing the inputs provided to the framework.

The rest of the paper is organized as follows. Section 2 provides a research background for our study. Section 3 introduces the approach we employ to address the problem of detecting fake news on emerging topics using weak supervision. It presents the proposed framework, WeSTeC, and details the individual steps involved in the overall pipeline. Section 4 presents the experiments conducted using the proposed framework, from the performance of the generated labeling functions to trained text classification models. Section 5 summarizes the proposed model and discusses the results of our experiments.

2 Related work

There have been several surveys on fake news detection which divide the existing research from different perspectives (Hu et al., 2022; Hamed et al., 2023). In Hamed et al. (2023), the approaches for detecting fake news are divided into categories like external knowledge-based, modality-based, and feature-based detection methods. Feature-based methods are further classified as content-based, social-context based, and hybrid approaches. Hu et al. (2022) follow a different approach and classify the existing work as supervised, weakly-supervised, and unsupervised. Our work aligns with the category of weakly-supervised content-based methods for fake news detection. In particular, we aim to tackle the problem of early detection of fake news on emerging topics.

The content-based methods use various types of information from the news, such as article content, news source, headline, and image/video, to build fake news detection classifiers. Horne and Adali (2017) divided the features into three distinct categories, stylistic, complexity, and psychological. Stylistic features refer to features based on natural language processing to understand the syntax, text style, and grammatical elements. Complexity features are based on sentence structure and readability levels. Psychological features are based on measures of cognitive processes, drives, and personal concerns. We use the analysis made in their work to select features that the content-based labeling functions utilize as part of the proposed framework. Validating the effectiveness of some of these features, Ngada and Haskins (2020) and Qin et al. (2016) also demonstrate the effectiveness of measuring punctuation and part-of-speech (POS) tagging-based features to distinguish between fake and real news articles. We also benefit from POS-tagging and punctuation-based features in our content-based labeling functions to strengthen the capability of differentiating between fake and real articles.

A significant amount of research has explored the use of social-context-based features, proving its effectiveness in detecting fake news (Ren et al., 2020; Shu et al., 2020; Wang et al., 2020; Yuan, 2020; Leite et al., 2023; Jlifi et al., 2023). Social context-based features are categorized into two types: network-based features and user-based features (Hamed et al., 2023). Network-based features are extracted by building specialized networks, such as propagation networks, interaction networks, and diffusion networks. User-based features include credibility, behavior, and profile characteristics. There are also hybrid approaches, integrating both news content and social context, recognizing the complementary nature of these dimensions for better fake news detection models (Raza & Ding, 2022). These approaches require social engagement data to be available for training, which contradicts our objective of detecting fake news on emerging topics before the dissemination through social media. The use of social context is only useful for the early detection of fake news if we have a large amount of social engagement data. The framework we propose focuses on the timely identification of fake news for unknown topics through domain-agnostic content features leveraged in automatically generated labeling functions. However, the framework can be extended to generate labeling functions based on social-context features as well, if such data is made available.

Deep learning and traditional supervised learning techniques construct predictive models by leveraging a substantial dataset of training instances, each paired with a corresponding ground-truth label. Many studies support the effectiveness of these methods in spotting fake news (Hu et al., 2022; Zhou & Zafarani, 2020; Singh et al., 2021). These models often use insights from research on fake news characteristics and social engagement data to carve out a list of features to use. Deep learning techniques are effective even for multilingual fake news detection, if such datasets are available (Mohawesh et al., 2023). In Galli et al. (2022) a benchmark framework is provided in order to analyze and discuss the most widely used and promising machine/deep learning techniques for fake news detection, also exploiting different features combinations w.r.t. the ones proposed in the literature.

Although supervised techniques used in fake news detection can attain accuracies on par with human capabilities to detect fake content, they require extensive labeled data for training. This limitation renders them unsuitable for addressing the early fake news detection problem in emerging topics. To overcome this challenge, numerous studies have focused on the use of weak supervision techniques as an alternative. Weak supervision approaches include methods such as semi-supervision (Konkobo et al., 2020; Dong et al., 2020), reinforcement learning (Wang et al., 2020), active learning (Ren et al., 2020) and distant supervision (Raza & Ding, 2022). Programmatic weak supervision (Ratner et al., 2017; Wu et al., 2023) aims to combine the aforementioned efforts by encoding potentially noisy probabilistic labels using labeling functions. To mitigate the noise from these weak signals, various frameworks aim to aggregate the outputs of several labeling functions into weak labels (Leite et al., 2023; Li et al., 2021; Özgöbek et al., 2022). We have also adopted programmatic weak supervision in attacking the early fake news detection problem. Therefore we focus on similar approaches in the rest of this section.

Li et al. (2021) show the effectiveness of multi-source domain adaptation in early fake news detection. They use domain-agnostic content features to weakly label the dataset of the target domain. They utilize three manually created labeling functions. In addition, they train source-specific fake news classifiers by fine-tuning models for the target domain. We also focus on the domain adaptation setting with weakly supervised strategies. Compared to the limited number of labeling functions in their work, our framework automatically generates up to 144 labeling functions, resulting in better aggregated weak label quality.

Leite et al. (2023) investigate the utilization of large language models (LLMs) to prompt 18 credibility signals effectively and produce weak labels for content veracity. By aggregating these labels using the Snorkel framework (Ratner et al., 2017), they can automatically generate weak labels for training. The paper offers valuable insights into the role of language models in creating individual credibility signals for predicting content veracity. However, it diverges from our work in that it does not specifically aim to detect fake news early as it emerges, and it also differs in its use of credibility issues and sentiments rather than content-based features.

Another related work (Shu et al., 2020) suggests employing weak social supervision for early fake news detection. User engagement with news articles, such as posts and comments, are considered weak signals for labeling fake news. The authors introduce three heuristic labeling functions based on user engagements to weakly label a large amount of data. They utilize a Label Weighting Network (LWN) to model the weights of these weak labels, contrasting with our approach that employs different aggregating strategies. Additionally, a distinction lies in the number of manually created labeling functions, whereas our framework is more flexible, automatically generating various labeling functions.

Özgöbek et al. (2022) outline a weakly supervised fake news detection schema through content features. Their work stands out as the closest counterpart in the existing literature to our research. Notably, it is the only study within our current knowledge that explores the auto-generation of labeling functions. They employ Snuba, as introduced by Varma and Ré (2018), to aggregate weak labels. Their content feature threshold selection algorithm is considered a part of our framework, albeit modified for our specific use case. We have enhanced the original algorithm with feature selection capabilities, enabling it to eliminate low-performing labeling functions based on content features without using ground-truth labels. The algorithm is further explained in Section 3.3.1. Their study exclusively focuses on content features for generating labeling functions, whereas our framework offers flexibility by accommodating various weak supervision strategies to derive labeling functions. Moreover, the versatility of our proposed framework, WeSTeC, enables support for multiple setups, including those examined in this research: semi-supervision and domain adaptation. Özgöbek et al. (2022) focus solely on a semi-supervision setup. Our framework achieves superior performance in fake news detection capabilities.

3 WeSTeC framework

We aim to address the problem of early detection of fake news on emerging topics in both semi-supervision and domain adaptation setups. To seamlessly test both of these setups, we introduce an end-to-end weakly supervised text classification framework (WeSTeC). The existing technology landscape is fragmented when it comes to programmatic weak supervision, and there is no easy solution to test weakly supervised text classification approaches end-to-end, from weak labeling to actually utilizing the assigned labels. We believe WeSTeC fills this gap by providing a consolidated and end-to-end solution, taking advantage of many weak supervision and text classification libraries that are popularly used, such as Snorkel[1] (Ratner et al., 2017), Hyper Label Model (Wu et al., 2023), SimpleTransformers[2], and more.

3.1 Overall architecture

Figure 1 illustrates the overall architecture of WeSTeC. The framework operates on two input datasets: a labeled dataset and an unlabeled dataset. It consists of two main modules: weak labeling and text classification. The weak labeling module is tailored to automatically assign aggregated weak labels to the unlabeled dataset, utilizing the labeled dataset in the labeling function generation process. This module employs two distinct submodules to automatically generate labeling functions and eliminate underperforming ones. The first submodule automatically generates content-based labeling functions using various features. The selected features and the elimination of under-performing labeling functions are explained in detail in Section 3.3.1. The other submodule is the model-based label function generator, which uses the given limited labeled data and employs various classification algorithms to generate labeling functions based on the resulting models. The details of this submodule are explained in Section 3.3.2.

Once all labeling functions are generated, they are applied to the unlabeled data. Each instance in the unlabeled dataset undergoes processing by the Content Feature Applier submodule which extracts the relevant features. The Labeling Function Applier submodule runs each labeling function on the unlabeled data instances one by one. Each labeling function generates a weak signal for the class of the unlabeled data instances. Depending on the number of labeling functions, that many weak signals will be created for each unlabeled data instance. These weak signals are then aggregated by the Weak Label Aggregator module.

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

[2] https://simpletransformers.ai/

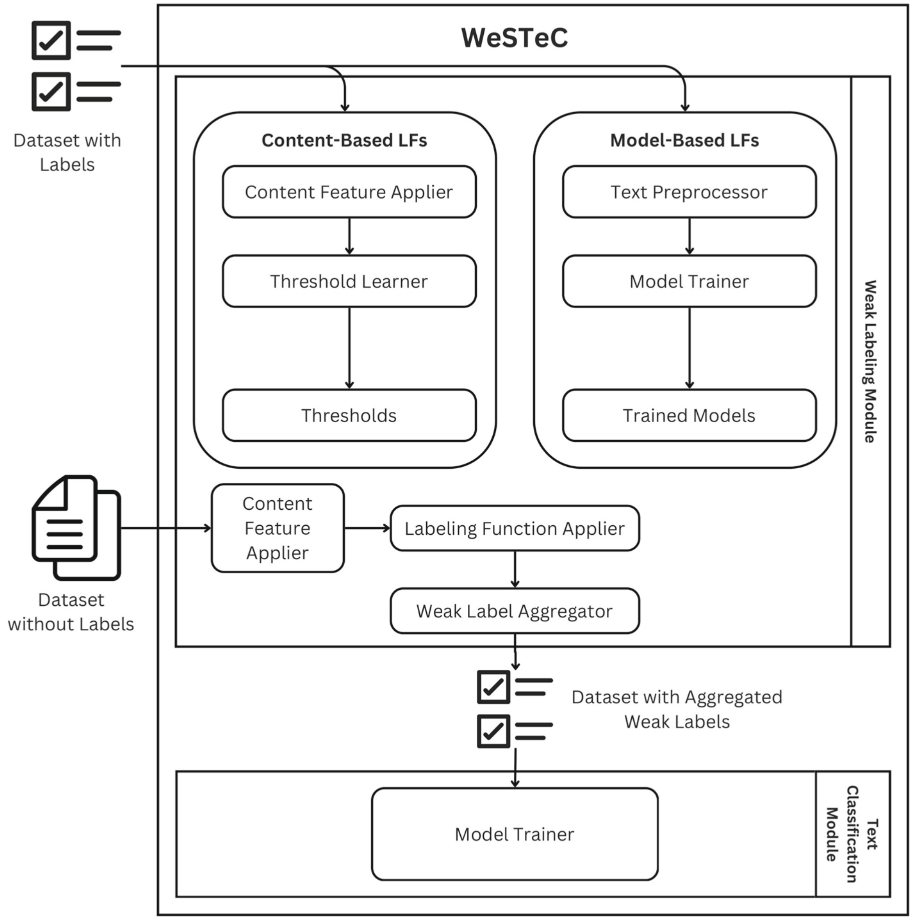

Fig. 1 Overall architecture of WeSTeC

WeSTeC supports multiple weak label aggregation strategies. The details of these submodules are explained in Section 3.4.

The next main module in the architecture is the text classification module, which is employed to train supervised text classification models using the aggregated weak labels applied to the unlabeled dataset. The details of this module is given in Section 3.5. By seamlessly integrating both modules, WeSTeC establishes a cohesive end-to-end weakly supervised text classification pipeline through programmatic weak supervision. The output of this comprehensive pipeline is a model trained on the unlabeled dataset provided to the framework. Therefore, the resulting model is finely tuned to the domain of the unlabeled input dataset.

3.2 Setups

To tackle the problem of fake news on emerging topics in the early stages of its dissemination through social media, we consider two practical setups, inspired by real-world scenarios; semi-supervision and domain adaptation.

Semi-Supervision In the first setup, we investigate the utilization of a limited amount of labeled data to automatically label a large dataset in the same domain. In the baseline scenario, assuming no labeled data is available for either the emerging topic or previous events and topics, semi-supervision techniques can be employed. Limited labeled data can be acquired through manual labeling. We separate a small subset of the unlabeled dataset and present it to the framework as the labeled dataset, with attached ground truth labels. The remaining portion is designated as the unlabeled dataset. Many early studies on fake news detection studies concentrate on similar setups, where the ratio of labeled data instances to unlabeled data instances typically ranges from 1% to 30% (Özgöbek et al., 2022; Dong et al., 2020; Konkobo et al., 2020). In our experiments, we consider the labeled data ratio to be less than 0.7%.

Domain Adaptation In practice, labeled data for different domains and past topics accumulate over time at the hands of enterprises. In domain adaptation setup, our goal is to use these labeled data sources to programmatically label data instances related to emerging topics. This enables our system to output a fake news detection model on the emerging topic, even without needing to provide a small manually labeled subset. This is preferable in the case of new emerging topics where manual labeling may be impractical even in small amounts.

3.3 Weak labeling

To reduce the human annotation efforts, the programmatic weak supervision paradigm abstracts weak supervision sources as labeling functions and involves a label model to aggregate the output of multiple labeling functions to produce training labels. Labeling functions are arbitrary code snippets that can encode various signals, such as patterns, heuristics, external data resources, noisy labels from crowd workers, weak classifiers, and more [3] (Ratner et al., 2017). The labeling functions can assign one of three options to each of the news articles: fake, real, or abstain.

WeSTeC generates and fine-tunes labeling functions using the labeled dataset, provided as one of the inputs to the weak labeling module. Then, the refined labeling functions are used to programmatically label the unlabeled dataset, provided as the other input. The labeling functions can be categorized into two different groups: content-based and model-based labeling functions.

3.3.1 Content-based labeling functions

Synthesizing findings from numerous studies, we have identified 41 content-based features to incorporate into our framework. The datasets used in our experiments consist of news articles with separate title and content columns. All 41 features are taken into account for the content section of news articles, whereas only 26 are applied to the title portion due to its concise length. At the onset of the weak labeling process, all these numerical features are calculated and appended to both labeled and unlabeled input datasets. We categorize the selected content features and provide a few examples for each category below. The complete list of features and the sections of news articles where they are utilized are provided in Table 1.

- **Stylistic features:** These features are based on text characteristics such as style, length, etc. (e.g., unique word ratio, average word length).

[3] http://ai.stanford.edu/blog/weak-supervision/

Table 1 Content Features

Stylistic Feature Name	Content	Title
Word count	✓	✓
Unique words count	✓	
Words per sentence	✓	
Stopwords ratio	✓	✓
Unique words ratio	✓	✓
Average sentence length	✓	
Average word length	✓	
Punctuation Feature Name	Content	Title
Punctuation ratio	✓	✓
Period ratio	✓	
Question mark ratio	✓	
Exclamation point ratio	✓	
Comma ratio	✓	
Semicolon ratio	✓	
Colon ratio	✓	
Parentheses opener ratio	✓	
Parentheses closer ratio	✓	
Quotation mark ratio	✓	
POS-Tagging Feature Name	Content	Title
Noun ratio	✓	✓
Proper noun ratio	✓	✓
Cardinal number ratio	✓	✓
Determiner ratio	✓	✓
Adposition ratio	✓	✓
Interjection ratio	✓	✓
Symbol ratio	✓	✓
Adjective ratio	✓	✓
Wh-determiner ratio	✓	✓
Verb ratio	✓	✓
Present participle verb ratio	✓	✓
Past participle verb ratio	✓	✓
Third-person verb ratio	✓	✓
Modal ratio	✓	✓
Adverb ratio	✓	✓
Comparative adverb ratio	✓	✓
Superlative adverb ratio	✓	✓
Existential ratio	✓	✓
Pronoun ratio	✓	✓
Personal pronoun ratio	✓	✓
Possessive pronoun ratio	✓	✓

Table 1 continued

Readability Feature Name	Content	Title
Gunning Fog Index	✓	
Automated Readability Idx	✓	
Flesch Kincaid Index	✓	

- **POS-tagging features:** These features are based on part-of-speech tags of words in a sentence. (e.g., proper noun ratio, adverb ratio).
- **Punctuation features:** These features explore various usages of punctuation symbols in news articles. (e.g., question mark ratio, period ratio).
- **Readability features:** They are used to estimate the education level required to understand the text. (e.g., automated readability index, Flesch Kincaid index).

One possible approach to create labeling functions based on the values of the listed features, is to manually analyze the distribution of feature values in both real and fake news articles using the labeled dataset. This analysis aims to identify specific feature values that can serve as thresholds for distinguishing between fake and real news. These thresholds, whether set as upper limits, lower limits, or both, establish a specific range of values for each feature. Subsequently, custom labeling functions can be created, using the determined threshold values for each feature, assigning labels such as fake, real, or abstain to news articles. However, this approach is time-consuming due to the manual determination of threshold values for each feature. In our weak labeling module, we employ a method for automatic threshold determination, inspired by the work of Özgöbek et al. (2022). We have modified the original algorithm, with a notable addition being the incorporation of an automated feature elimination mechanism.

In order to automatically determine thresholds, the distribution of the labeled dataset is examined for each feature. The labeled dataset is initially split into two subsets based on their labels. The distribution of the real and fake subsets is then identified for each feature under consideration. Percentiles are chosen with intervals of 0.05, effectively dividing the range (0, 1) into 20 slices. For example, consider the feature *content_words_per_sentence* in the elections dataset, representing the average number of words per sentence in the news article content. Figure 2 depicts the percentiles for this feature, illustrating differences across various percentiles for both the real and fake subsets of the dataset.

Given the existence of infinitely many potential threshold values for each feature, the problem is simplified by looking for two thresholds for each feature: one for percentiles below 0.5 (ranging from 0.05 to 0.5, called the lower threshold) and one for percentiles above 0.5 (ranging from 0.5 to 0.95, called the upper threshold). The pseudocode for searching the upper percentile threshold for a single feature is provided in Algorithm 1. For each percentile, the algorithm calculates the difference between the feature values in the real dataset and the fake dataset to identify a noteworthy distinction in the feature values between fake and real datasets. If a significant difference is detected at a percentile, the algorithm determines the threshold and the side. If no significant difference is found, the feature is not utilized to generate a labeling function. A similar algorithm is also applied for lower thresholds. In the case of lower thresholds, percentiles are iteratively adjusted from 0.5 to 0.05 until a specific threshold is identified or all percentiles within the range are exhausted. These algorithms are executed for all features, aiming to determine upper and/or lower percentile threshold or exclude the feature from further consideration.

The algorithm also outputs a side as fake or real. This is determined by examining the sign of the difference between fake and real features values at specific percentiles. If the difference

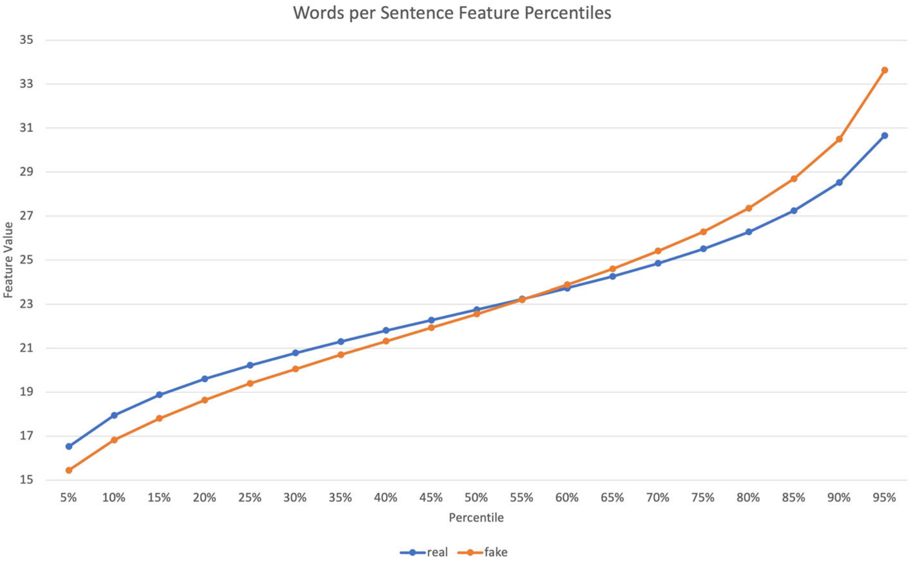

Fig. 2 *content_words_per_sentence* percentiles

is positive, it suggests that the values of the real dataset are more likely to be higher than those of the fake dataset beyond this particular percentile. Consequently, "real" is chosen as the side. Conversely, if the difference is negative, "fake" is selected as the side.

Algorithm 1 Threshold Search for Upper Percentiles

```
Require: fake dataset, real dataset, Feature
Ensure: threshold, side
$max\_fake \leftarrow$ maximum value of feature on the fake dataset
$min\_fake \leftarrow$ minimum value of feature on the fake dataset
$max\_real \leftarrow$ maximum value of feature on the real dataset
$min\_real \leftarrow$ minimum value of feature on the real dataset
$max\_all \leftarrow \max(max\_fake, max\_real)$
$min\_all \leftarrow \min(min\_fake, min\_real)$
$total\_diff \leftarrow max\_all - min\_all$
$threshold \leftarrow$ none
$side \leftarrow$ none
for each percentile p between 0.5 to 0.95 do
    $real\_value \leftarrow$ value of real dataset for p
    $fake\_value \leftarrow$ value of fake dataset for p
    $percentile\_diff \leftarrow real\_value - fake\_value$
    if $|percentile\_diff| > \frac{total\_diff}{C}$ then
        if $percentile\_diff > 0$ then
            $side \leftarrow$ real
            $threshold \leftarrow real\_value$
        else
            $side \leftarrow$ fake
            $threshold \leftarrow fake\_value$
        end if
        break
    end if
end for
return threshold, side
```

The thresholds and sides are then used to generate content-based labeling functions. It is possible that a feature may have two thresholds, representing the lower and upper percentiles, or just one threshold for either percentile. For every threshold discovered, the framework generates a distinct labeling function for that feature. It is also possible that certain features may not have identifiable thresholds; in such cases, the respective feature is eliminated from further consideration. Continuing our example *content_words_per_sentence* feature, let's assume the threshold search algorithm stops the upper sweep at the *90%* percentile mark. This means the percentile difference between fake and real subsets at this point is noteworthy and this point can be used to obtain a threshold value. For the upper threshold search, value of the higher subset at this point is taken as the threshold value, which is fake in our case, with a value of *30.488*. Therefore, the side is also selected as fake in this example. For the given scenario, the following labeling function is generated.

```
def words_per_sentence_upper_fake(x):
    return Labels.FAKE
    if x["words_per_sentence"] > 30.488
    else Labels.ABSTAIN
```

At most two thresholds are found for each feature. Since we have 67 features for content and title combined, potentially 134 labeling functions can be generated out of this process, if there are no eliminations. Feature elimination allows users to introduce as many content features as possible without concern for diminishing the overall aggregated weak label accuracy.

Once a threshold is identified for a feature, the remaining percentiles are skipped in Algorithm 1. It is preferable to find a threshold as early as possible, as this allows the algorithm to assign labels (real or fake) to a greater number of data instances rather than abstaining. The algorithm uses a constant, denoted as *C*, which plays a crucial role in detecting significant differences at a given percentile. A critical tradeoff exists with this constant – increasing it causes the algorithm to identify the threshold in percentiles closer to the middle, covering more data instances. However, as the difference between real and fake subsets becomes smaller, the threshold becomes less selective, leading to mislabeling instances by labeling functions created from the selected threshold. The constant can be chosen based on the desired tradeoff between coverage and accuracy, depending on specific application requirements.

We conducted a separate analysis to intelligently select the constant *C* for our use case. In this analysis, we utilized the labeled dataset provided in the domain adaptation setup. Half of the dataset was employed to generate content-based labeling functions by running the threshold selection algorithm, while the other half was reserved to evaluate the performance of the aggregated weak labels. Only content-based labeling functions and the Snorkel Label model were employed for this analysis. The threshold search algorithm was executed with various *C* values in the range of 1 to 20. We analyzed the number of labeling functions and accuracy results to determine a suitable constant. The results of this analysis are presented in Table 2.

In our study, we prioritized higher accuracies over a greater number of labeling functions. Upon reviewing the results, choosing *C* as 5 optimizes the aggregated accuracies while maintaining an adequate number of labeling functions for our use case. This value is configured in the framework and applied consistently throughout this study. The users of the framework have the option to either adhere to this value or undertake separate analyses to determine a more suitable constant for their specific use cases, subsequently updating the configuration accordingly.

Table 2 Threshold Search Algorithm Constant Selection Analysis

Constant Value (C)	Snorkel Label Model Acc.	Number of LFs
1	N/A	2
3	0.627	10
5	0.640	27
7	0.633	40
9	0.617	49
11	0.614	55
13	0.608	62
15	0.530	70
17	0.487	76
19	0.489	79

The algorithm requires percentile values for each feature to be prepared before execution. We utilize the *describe* method in the Pandas library to obtain these statistics.[4] Under the hood, it employs the *numpy.percentile* function for computation.[5] This function utilizes a linear interpolation algorithm with a time complexity of $O(n)$, where n is the number of data points in the provided dataset. We apply the *describe* method to both real and fake subsets. Retrieving the minimum and maximum values for a feature takes $O(n)$ time, where n is the number of data instances. Given we have a fixed number of percentiles (20), the remaining part of the algorithm takes $O(1)$ time. Overall, the entire algorithm has a time complexity of $O(n)$. This algorithm is executed for each selected feature.

3.3.2 Model-based labeling functions

The main goal of weak labeling systems is to combine as many diverse weak label sources as possible for optimal utilization. In addition to numerous content-based labeling functions, incorporating other types of weak labels can enhance the accuracy of the combined weak labels. To achieve this, we integrate the content-based labeling function generation strategy with model-based labeling functions. This includes using the labeled dataset to train supervised machine-learning models, with these trained models subsequently acting as weak labeling sources for the unlabeled dataset.

The models trained for model-based labeling functions serve as one of the various weak signals for generating different labeling functions. Consequently, we chose simpler machine learning algorithms that can perform effectively with limited labeled data and require minimal resources for training, avoiding more complex deep learning approaches for the specified reasons. The considered models are Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), Adaboost, and XGBoost.

Similar to content-based labeling functions, model-based labeling functions are also generated for both the content and title sections of news articles separately. After training the models with the labeled dataset, the trained models are saved for use in the subsequent steps. In total, we have 10 models saved and ready to be employed as model-based labeling functions.

[4] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html

[5] https://numpy.org/doc/stable/reference/generated/numpy.percentile.html

The labeling functions created for these models predict the label using the trained models and directly output the predicted value as the labeling function output. Unlike content-based labeling functions, these do not return the value “abstain”, as the trained models always assign a real or fake value. This also implies that the coverage of these labeling functions is always 100%, where all data points are assigned either real or fake values as a result. This differs from content-based labeling functions, where each labeling function assigns only one of the real or fake values depending on the threshold or abstains, leading to various coverage values depending on the content feature.

3.4 Weak label aggregation

Once all labeling functions are generated, the next stage in the pipeline involves applying these labeling functions to the unlabeled dataset. Assuming *m* labeling functions are generated in total and there is *n* news articles in the unlabeled dataset, a matrix with dimensions *m* x *n* is generated at the end of the labeling function application. In this matrix, each row identifies a data instance and all columns identify the outcome of labeling functions where the cells can take values of -1, 0, or 1, representing abstain, real, and fake respectively.

Various studies explore how to combine weak labels without having access to ground truth labels. WeSTeC supports three of the commonly used weak signal aggregation strategies which are explained below.

Majority Vote This is the simplest approach to aggregate weak labels per data instance. Assuming the weight of each labeling function is equal, a single aggregated label can be assigned to each instance by counting the number of assigned fake and real weak labels.

Snorkel Label Model Ratner et al. (2017) introduce a probabilistic graphical model-based approach to combine noisy weak labels into a final set of training labels. The model estimates the accuracy and correlations of the labeling functions, combining their outputs to generate a probabilistic distribution over the true labels, resulting in a final continuous value describing the probability of the instance being fake. We convert these probabilistic labels to discrete labels before proceeding to the text classification phase.

Hyper Label Model The final strategy WeSTeC supports is the hyper label model. Wu et al. (2023) introduced a graph neural networks (GNN) based label model, which infers the aggregated labels in a single forward pass. It leverages deep learning to approximate an optimal analytical solution, for label aggregation. Different from the Snorkel label model, the hyper label model also considers conditional dependencies between labeling functions. It adjusts the weights and directly outputs discrete labels assigned by the model, as opposed to probabilistic labels like the Snorkel Label Model.

The output of Weak Label Aggregator (see Fig. 1) is a dataset with aggregated weak labels, which serves as the training data for the text classification module. One can utilize the entire dataset for training, or alternatively, choose only the most confidently labeled data. To facilitate this selection process, WeSTeC includes an additional mechanism based on probabilistic labels generated by the Snorkel Label Model. If the user specifies a desired number of data instances for training and the initial unlabeled dataset exceeds this count, the framework selects instances with the highest confidence according to the Snorkel label model. Assuming *x* is specified as the number of data instances to select, the framework chooses *x/2* instances, starting from those with the highest likelihood of being fake and decreasing as the selection progresses. Similarly, *x/2* instances are selected, starting from those with the lowest probability of being fake and increasing as the selection proceeds, in order to create a balanced training dataset.

3.5 Text classification

Various studies show the success of large language models in the text classification setting compared to previous approaches (Gasparetto et al., 2022). These also include studies in fake news classification (Samadi et al., 2021; Özgöbek et al., 2022). The RoBERTa text classifier is selected as the single end-model in our framework given its superiority demonstrated by many studies.

RoBERTa is a transformer-based language model pre-trained on a vast corpus of text using a masked language modeling objective (Liu et al., 2019). For text classification, RoBERTa is fine-tuned on labeled data, adding task-specific layers. During inference, the model processes raw text input, tokenizes it, and outputs predictions based on its learned contextualized representations.

The text classification module in WeSTeC utilizes training data derived from selecting the highest-performing aggregated weak labels (as explained in Section 3.4). The pre-trained RoBERTa model undergoes fine-tuning on this training data. We also conduct hyperparameter tuning for the learning rate. The output of the text classification module is a ready-to-use, fine-tuned RoBERTa text classifier specialized for the domain of the unlabeled dataset. Evaluation of our model is performed using the actual labels, and given the various weak label aggregation strategies employed in our work, only the highest-performing strategy is chosen for implementation in the text classification module.

3.6 Time complexity analysis

The proposed framework executes a series of modules sequentially, as explained in Section 3.1. The complexities of the content-based labeling function generation pipeline, content feature applier, and labeling function applier are all linear time proportional to the total size of the labeled dataset (n) and the unlabeled dataset (m). These datasets undergo preprocessing for the extraction of the content-based features once. The content-based labeling functions are generated by a single pass over the labeled data set. Then the content feature applier passes over the unlabeled dataset once, and the chosen labeling functions are applied to each data instance in the unlabeled dataset. Given that the complexity of each labeling function is $O(1)$ and there is a constant number of labeling functions, we can deduce that the time complexity of the pipeline so far is $O(n+m)$. However, the complexity of the overall pipeline is dominated by the complexities of the remaining three modules (i.e. model-based labeling function generation pipeline, weak label aggregator, and text classification model training). Their complexities depend on third-party libraries, such as Snorkel, Hyper label model, the machine learning algorithms used to generate the model-based labeling functions and the final RoBERTa text classifier.

4 Results & discussions

This section presents experiments carried out on the proposed framework in both semi-supervision and domain adaptation setups. It also presents a comparison with the reported results of the state-of-the-art methods in weakly supervised fake news detection.

4.1 Dataset

Various publicly accessible datasets are available for research on fake news detection, and D'ulizia et al. (2021) and Hu et al. (2022) offer a comprehensive comparison of these datasets.

We have chosen to use the 2021 and 2022 editions of the NELA-GT news article dataset (Gruppi et al., 2021), due to its alignment with criteria such as the number of data instances, suitability for domain transfer, and the availability of ground truth labels. NELA-GT datasets comprise regularly published news article datasets sourced from over 500 news outlets. The labeling methodology of this dataset is based on source reliability. Since its initial edition in 2017, various subsets have been released, each focusing on specific topics such as the US elections and Covid for the 2020 edition, and the US Capitol attack and Covid for the 2021 edition. We choose Covid and US Elections as two distinct topics. Table 3 shows statistics for both datasets. The datasets contain more real news articles than fake news articles. We undersample both datasets to obtain balanced datasets.

We evaluated both semi-supervision and domain adaptation setups using these datasets. In both setups, we consistently use Covid-related news articles as the unlabeled dataset, and we tailor the labeled input dataset based on the specific setup. Encompassing the entire process, from automated labeling function generation to text classification model training, WeSTeC provides an effective foundation for conducting these experiments and allows us to test both setups by simply changing the inputs.

In the semi-supervision setup, we extract 2000 data instances from the Covid dataset to create a separate labeled dataset. The original dataset is left with 295,518 data instances, resulting in a labeled-to-unlabeled data ratio of 0.67%. Instances for the subset dataset are randomly selected in a balanced manner, resulting in 1000 data instances for each label type. In the domain adaptation setup, we use the same unlabeled dataset of 295,518 data instances, but we switch the labeled dataset with the elections dataset, which consists of 80,369 news articles in the politics domain.

4.2 Evaluation of the weak labeling module

As detailed in Section 3.3.1, we conduct feature elimination using the threshold search algorithm for content-based labeling functions. Consequently, the number of labeling functions may vary based on the provided labeled dataset and the constant C chosen within the threshold selection algorithm. In the configuration described in Section 3.3.1, for the unlabeled dataset comprising Covid-related news articles in the semi-supervision setup, 35 labeling functions are generated out of a possible 134. In the domain adaptation setup, maintaining C at a value of 5, the module produces 34 labeling functions.

Tables 4 and 5 show the top-performing 10 and worst-performing 5 labeling functions after their application to the unlabeled dataset. Observing the worst-performing labeling functions in both setups reveals that, individually evaluated without combination with other labeling functions, all identified labeling functions, except one, exhibit higher accuracy than random guessing. This underscores the efficacy of the feature elimination capabilities of the threshold search algorithm. Examining the top-performing labeling functions demonstrates that both model and content-based labeling functions contribute to the top 10 in both setups,

Table 3 NELA-GT Dataset Statistics

Statistic	Covid	Elections
Total number of rows	479,245	118,525
Number of rows where the label is “fake”	148,759	40,011
Number of rows where the label is “real”	330,486	78,514
Total number of rows after balanced undersampling	297,518	80,022

Table 4 Top-Performing Labeling Functions

Semi-Supervision Labeling Function Name	Emp. Acc.	Domain Adaptation Labeling Function Name	Emp. Acc.
model_log_regression_content	0.768	model_xgboost_content	0.849
model_naive_bayes_content	0.753	model_log_regression_content	0.820
model_xgboost_content	0.744	model_naive_bayes_content	0.729
model_random_forest_content	0.738	content_exc_point_upper_fake	0.723
content_exc_point_upper_fake	0.736	model_xgboost_title	0.715
content_fkincaid_index_upper_fake	0.728	model_log_regression_title	0.712
title_noun_ratio_upper_real	0.719	title_proper_noun_upper_fake	0.711
content_readability_index_upper_fake	0.717	model_naive_bayes_title	0.705
title_proper_noun_ratio_upper_fake	0.711	model_adaboost_content	0.696
title_word_count_upper_fake	0.708	content_parantheses_upper_fake	0.689

highlighting the effectiveness of both approaches in generating accurate labeling functions. Comparatively, model-based labeling functions exhibit more dominant results in the domain adaptation setup than in the semi-supervision setup. This shows the effectiveness of model-based labeling functions in setups with a larger number of labeled data instances, even if they are in different domains.

We present the evaluation results of the performance of weak label aggregation strategies for both setups using the accuracy, precision, recall, F1-score, and coverage metrics in Table 6. Snorkel label model results for both before and after the data instance selection are also presented in the table. Examining the results, it is evident that all strategies achieve higher accuracy compared to individual labeling function performances. In Table 4, within the semi-supervision setup, the top-performing labeling function attains an accuracy of 0.768, while in Table 6, even the lowest aggregation results exhibit a higher accuracy of 0.784. The difference becomes more substantial when considering the data selection layer. This highlights the efficacy of weak supervision and shows the importance of being able to combine various weak labeling sources.

When comparing various labeling function aggregation strategies without data selection, all strategies yield similar results. However, the majority vote strategy slightly outperforms the other, more complex approaches. This finding aligns with the results obtained by Wu et al. (2023), who developed the hyper-label model aggregation strategy. Nevertheless, it's worth noting that the majority vote strategy abstains when the number of votes is the same, leading to slightly less coverage than 100%.

Table 5 Worst-Performing Labeling Functions

Semi-Supervision Labeling Function Name	Emp. Acc.	Domain Adaptation Labeling Function Name	Emp. Acc.
title_card_number_upper_fake	0.490	title_determiner_upper_real	0.491
title_punctuation_lower_real	0.510	title_adverb_upper_fake	0.507
content_existential_lower_fake	0.512	title_verb_present_upper_fake	0.516
content_existential_upper_real	0.535	content_semicolon_upper_real	0.518
title_pronoun_upper_fake	0.542	title_adposition_lower_real	0.527

Table 6 Performance of Weak Label Aggregation Strategies

Setup	Aggregation Strategy	Acc.	Prec.	Recall	F1	Cov.
Semi-Supervision	Majority Vote	0.794	0.790	0.806	0.798	0.963
	Hyper LM	0.784	0.785	0.784	0.783	1.000
	Snorkel LM	0.791	0.790	0.792	0.791	1.000
	Snorkel LM, w/ selection	0.931	0.931	0.932	0.931	1.000
Domain Adaptation	Majority Vote	0.839	0.853	0.818	0.835	0.964
	Snorkel LM	0.834	0.824	0.851	0.837	1.000
	Hyper LM	0.825	0.827	0.825	0.825	1.000
	Snorkel LM, w/ selection	0.948	0.948	0.948	0.948	1.000

Furthermore, the data selection process applied to the Snorkel label model strategy significantly improves accuracy in both setups, showing the effectiveness of probabilistic labels. When comparing the two setups, the results indicate improved accuracies for the aggregated labels compared to the previous configuration. This shows the advantage of having a higher number of labeled data instances, even if they originate from another domain. The weak labeling module terminates by assigning aggregated weak labels to the unlabeled dataset obtained by each aggregation strategy.

4.3 Evaluation of the text classification module

In this section, we present the evaluation results of text classification for fake news detection. The performance of the classification module heavily relies on the performance of the weak labeling module. This dependence arises because the text classification module utilizes the assigned weak labels to fine-tune a pre-trained RoBERTa text classification model. Therefore, we selected the best-performing aggregation strategy in evaluating the text classification, which is Snorkel label model with data selection layer. 50,000 data instances are selected while ensuring a balanced distribution between fake and real news. We utilize the Simple transformers library[6], which offers complete support for RoBERTa text classification. The pipeline is initiated by converting text into tokens, as expected by the RoBERTa text classifier. Training is conducted for 3 epochs with default model hyperparameters, except for the learning rate, which is fine-tuned through sweeps using WandB[7].

Table 7 presents the results for the trained RoBERTa text classifier using an 80/20 train-test split. The first two rows show the outcomes for cases where the training is conducted using aggregated weak labels in both setups. It's important to note that the trained model is tested against the actual labels, which are not available during training. Additionally, an identical training configuration is adopted for the scenario involving actual ground truth labels in the training process, and the results for this setup are shown in the third row of Table 7. This approach aims to assess how closely the proposed solution in both setups can approach its fully supervised counterpart. All other details, including model hyperparameters and selected data instances, are kept the same for comparable results.

The results show that models trained on programmatically assigned labels can achieve scores that are comparable to those obtained using the actual ground truth labels. In both

[6] https://simpletransformers.ai/

[7] https://wandb.ai/site

Table 7 Fake News Detection Results

Setup	Accuracy	Precision	Recall	F1 score
Semi-Supervision	0.952	0.952	0.953	0.952
Domain Adaptation	0.961	0.961	0.961	0.960
Supervised	0.968	0.968	0.968	0.968

setups, the difference in accuracy scores between the fake news classifier trained with weak labels and actual labels is below 2%, validating the effectiveness of the aggregated weak labels. When comparing the setups, it is observed that the domain adaptation setup slightly outperforms the semi-supervision setup. This is expected, considering the higher aggregated weak label performances shown in Table 6.

In the semi-supervision setup, the accuracy score of the aggregated labeling function is reported as 0.931 in Table 6. These labels are used to train the text classification model, which achieves an accuracy of 0.952 when tested against the actual labels. A similar pattern is observed in the domain adaptation setup, where the accuracy of the aggregated weak label is 0.948, while the fake news classifier achieves an accuracy of 0.961. This substantiates the hypothesis that classification models trained on aggregated weak labels can generalize beyond the initial labels and achieve higher accuracies.

4.4 Comparison with existing weakly supervised fake news detection studies

In this section, we present a comparison between the performance of the proposed model and the reported performances of other weakly supervised fake news detection studies. We provide a brief description of each study included in our comparison.

- **TDSL** Dong et al. (2020) introduce a semi-supervised learning framework for timely fake news detection through two paths CNN. Small amounts of labeled data are fed through one of the CNN paths while the other path is provided with a huge amount of unlabeled data. They show their results through different labeled data ratios. We include experiments where they make use of 1% and 30% labeled data ratios separately.
- **AA-HGNN** Ren et al. (2020) propose a novel approach that uses heterogeneous information networks to detect fake news in a timely manner. They use active learning to continuously query high-value candidate nodes for classifier training and tuning, achieving high performance even in the semi-supervision setup.
- **SSLNews** Konkobo et al. (2020) developed a three-path CNN-based deep learning model for the early detection of fake news. They mainly utilize user interactions through comments. Their experiments are conducted on a setup where the labeled to unlabeled data ratio is 25%.
- **MDA-WS** In their study, Li et al. (2021) focus on multi-source domain adaptation setup. They use domain-agnostic features to weakly label the dataset of the target domain. They also introduce a schema to train source-specific fake news classifiers by fine-tuning models for the target domain. They evaluate their results on three different domains in a 2-fold cross-validation fashion. We take the average of all three results to include in our evaluation.
- **MWSS** Shu et al. (2020) introduced a model to leverage multi-source weak social supervision for early detection of fake news. They utilize contextual social media information like user and content engagements.

Table 8 Fake News Detection Results

Method	Accuracy	F1 score
TDSL, 1% LDR Dong et al. (2020)	0.798	0.886
TDSL, 30% LDR Dong et al. (2020)	0.834	0.909
AA-HGNN Ren et al. (2020)	0.675	0.639
SSLNews, 25% LDR Konkobo et al. (2020)	0.695	-
MDA-WS Li et al. (2021)	0.769	0.768
CNN-MWSS Shu et al. (2020)	0.795	0.805
RoBERTa-MWSS Shu et al. (2020)	0.810	0.810
FND-NS, domain adaptation Raza and Ding (2022)	0.748	0.749
Ozgobek et al., <1% LDR Özgöbek et al. (2022)	0.942	0.942
WeSTeC, semi-supervision, <1% LDR	0.952	0.952
WeSTeC, domain adaptation	0.961	0.961
RoBERTa, supervised	0.968	0.968

- **FND-NS** Raza and Ding (2022) propose a transformer-based approach to detect fake news based on both news content and social contexts. Their work is focused on effective automated labeling to address the ground-truth label problem.
- **Ozgobek et al.** In their study, Özgöbek et al. (2022) proposed a weakly supervised fake news detection model using only content-based features. They utilized Snuba to weakly label fake news articles in the semi-supervision setup, followed by training a fake news detection classifier using the weak labels.

We provide the comparison of all mentioned approaches and our results together in Table 8. We use accuracy and F1 score metrics to present the results. For studies that explore the semi-supervised setups, we also highlight the labeled data ratio (LDR), indicating the ratio of labeled data instances to unlabeled data instances.

The results show that our approach demonstrates superior performance when compared with all state-of-the-art baselines in both the semi-supervision and domain adaptation setups. Our approach combines different weak labeling strategies, resulting in higher performance of the weak labels. Consequently, the fake news detection classifier trained with weak labels achieves better performance when evaluated against the actual labels.

Furthermore, the proposed framework allows for the integration of many content features, unlike other studies that typically utilize a limited set of features. The only exception in the list that also benefits from numerous content features is the work by Özgöbek et al. (2022) which shows the highest performance after our study. This shows the importance of using as many features and weak labeling sources as possible. With the content feature selection layer provided by WeSTeC, many content features can be introduced without compromising the overall performance of the aggregated weak labels.

5 Conclusion

In this paper, we propose a framework for the early detection of fake news on emerging topics, leveraging a programmatic weak supervision approach. Traditional methods such as supervised learning and fact-check-based detection mechanisms struggle with effectively addressing the early detection of fake news when a new topic emerges on the open web due to the absence of prior knowledge, labeled datasets, or fact-check articles. Our work demonstrates how to programmatically label large-scale datasets related to emerging topics and then employ supervised classification approaches using the automatically assigned weak labels.

We consider two essential setups for the early detection of fake news, inspired by real-world use cases. The first is the semi-supervision setup, where only a small amount of labeled data instances is available. In this setup, the proposed framework generates labeling functions using these instances, which are then employed to programmatically label large-scale unlabeled datasets on emerging topics. The second setup is domain adaptation, where labeled data from past events and different domains are accessible. WeSTeC utilizes these labeled datasets from different domains to generate labeling functions and effectively label a large-scale dataset related to an emerging topic. This setup eliminates the need for manual labeling of even a small portion of the unlabeled dataset, enabling more timely detection of fake news on the emerging topic at hand.

To accommodate both setups seamlessly, we introduce an end-to-end weakly supervised text classification framework. This framework is not only beneficial for fake news detection but is also applicable to various text classification tasks. Adapting to various weakly supervised learning tools and libraries often involves a steep learning curve, requiring similar steps for text classification tasks. With WeSTeC, we empower users to easily execute fully automated weakly supervised text classification pipelines by providing only two inputs: a limited labeled dataset and an unlabeled dataset. The versatile structure of WeSTeC facilitates its use in different setups, including, but not limited to, the two setups we experimented with: semi-supervision and domain adaptation. We believe that WeSTeC addresses a crucial gap in the programmatic weak supervision technology landscape.

The proposed framework integrates key text classification and fake news detection techniques, incorporating novel improvements like combining multiple weak labeling approaches and automatic content feature elimination. These enhancements have enabled us to surpass similar weakly supervised fake news detection baselines by 1%. We have demonstrated the effectiveness of our weak labeling strategy by training fake news classifiers with both generated weak labels and actual labels. The disparity in accuracy scores between these classifiers is below 2%, validating the robustness of the weak labeling pipeline in both setups we experimented with. Additionally, WeSTeC facilitated experimentation with different alternatives of the same steps, ensuring the selection of the highest-performing option. For instance, by having access to three distinct weak label aggregation strategies, we visualized how each approach performs in our case.

One limitation of the proposed model is its exclusive reliance on textual content, overlooking potential insights derived from social context in fake news detection. Moreover, our current approach neglects the incorporation of multi-modal features such as images or videos, sometimes available alongside textual content, as the chosen dataset is restricted to text-only news articles. While these additional features are not suitable for this study, the architecture of WeSTeC allows for the easy addition of more labeling function generation strategies. In future work, with suitable datasets, additional labeling function generation strategies can be incorporated into the proposed framework to enhance its fake news detection capabilities.

Furthermore, we plan to assess the applicability of the proposed framework in diverse text classification tasks beyond fake news detection. Additionally, we aim to enhance WeSTeC to handle weakly supervised multi-class text classification tasks. Currently, the threshold selection algorithm for content-based labeling functions facilitates binary classification only.

Moreover, we aim to validate the effectiveness of the domain adaptation setup through other datasets containing news articles or user-generated content originating from diverse domains.

Acknowledgements Not applicable

Author Contributions S.A. and N.K.C. jointly contributed to this manuscript. N.C. revised the work for intellectual content and approved the final version for publication. S.A. played a pivotal role in the conception, design, data analysis, and software development for the study. Both authors reviewed and finalized the manuscript.

Funding Not applicable

Availability of supporting data The supporting data and associated programs for this journal submission are available in a dedicated GitHub repository. Access to these resources is available upon request. Interested parties may request access to the data and programs by contacting the corresponding author of this submission.

Declarations

Competing interests The authors claim that they do not have any conflicts of interest.

Ethical Approval Not applicable

References

Dong, X., Victor, U., & Qian, L. (2020). Two-path deep semisupervised learning for timely fake news detection. *IEEE Transactions on Computational Social Systems, 7*(6), 1386–1398. https://doi.org/10.1109/TCSS.2020.3027639

D'ulizia, A., Caschera, M.C., Ferri, F., et al. (2021). Fake news detection: a survey of evaluation datasets. *PeerJ Computer Science, 7*, e518. https://doi.org/10.7717/peerj-cs.518

Galli, A., Masciari, E., Moscato, V., et al. (2022). A comprehensive benchmark for fake news detection. *Journal of Intelligent Information Systems, 59*(1), 237–261. https://doi.org/10.1007/s10844-021-00646-9

Gasparetto, A., Marcuzzo, M., Zangari, A., et al. (2022). A survey on text classification algorithms: From text to predictions. *Information 13*(2). https://doi.org/10.3390/info13020083

Gruppi, M., Horne, B.D., & Adalı, S. (2021). Nela-gt-2020: A large multi-labelled news dataset for the study of misinformation in news articles. arXiv preprint arXiv:2102.04567 https://doi.org/10.48550/arXiv.2102.04567

Hamed, S. K., Aziz, M. J. A., & Yaakub, M. R. (2023). A review of fake news detection approaches: A critical analysis of relevant studies and highlighting key challenges associated with the dataset, feature representation, and data fusion. *Heliyon 9*(10). https://doi.org/10.1016/j.heliyon.2023.e20382

Horne, B. D., & Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. arXiv:1703.09398. https://api.semanticscholar.org/CorpusID:7083781

Hu, L., Wei, S., Zhao, Z., et al. (2022). Deep learning for fake news detection: A comprehensive survey. *AI Open, 3*, 133–155. https://doi.org/10.1016/j.aiopen.2022.09.001

Jlifi, B., Sakrani, C., & Duvallet, C. (2023). Towards a soft three-level voting model (soft t-lvm) for fake news detection. *Journal of Intelligent Information Systems, 61*(1), 249–269. https://doi.org/10.1007/s10844-022-00769-7

Konkobo, P. M., Zhang, R., Huang, S., et al. (2020). A deep learning model for early detection of fake news on social media. In: *2020 7th International Conference on Behavioural and Social Computing (BESC)*, IEEE, (pp 1–6). https://doi.org/10.1109/BESC51023.2020.9348311

Lazer, D. M., Baum, M. A., Benkler, Y., et al. (2018). The science of fake news. *Science, 359*(6380), 1094–1096. https://doi.org/10.1126/science.aao2998

Leite, J. A., Razuvayevskaya, O., Bontcheva, K., et al. (2023). Detecting misinformation with llm-predicted credibility signals and weak supervision. arXiv:2309.07601. https://doi.org/10.48550/arXiv.2309.07601

Li, Y., Lee, K., Kordzadeh, N., et al. (2021). Multi-source domain adaptation with weak supervision for early fake news detection. In: *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, (pp. 668–676). https://doi.org/10.1109/BigData52589.2021.9671592

Liu, Y., Ott, M., Goyal, N., et al. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692. https://doi.org/10.48550/arXiv.1907.11692

Mohawesh, R., Maqsood, S., & Althebyan, Q. (2023). Multilingual deep learning framework for fake news detection using capsule neural network. *Journal of Intelligent Information Systems* (pp. 1–17). https://doi.org/10.1007/s10844-023-00788-y

Ngada, O., & Haskins, B. (2020). Fake news detection using content-based features and machine learning. In: *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, (pp. 1–6). https://doi.org/10.1109/CSDE50874.2020.9411638

Özgöbek, Ö., Kille, B., From, A. R., et al. (2022). Fake news detection by weakly supervised learning based on content features. In: *Symposium of the Norwegian AI Society*, (pp. 52–64), https://doi.org/10.1007/978-3-031-17030-0_5

Qin, Y., Wurzer, D., Lavrenko, V., et al. (2016). Spotting rumors via novelty detection. arXiv:1611.06322. https://doi.org/10.48550/arXiv.1611.06322

Ratner, A. J., Bach, S. H., Ehrenberg, H. R., et al. (2017). Snorkel: rapid training data creation with weak supervision. *The VLDB Journal, 29*, 709–730. https://doi.org/10.1007/s00778-019-00552-1

Raza, S., & Ding, C. (2022). Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics, 13*, 335–362. https://doi.org/10.1007/s41060-021-00302-z

Ren, Y., Wang, B., Zhang, J., et al (2020) Adversarial active learning based heterogeneous graph neural network for fake news detection. *2020 IEEE International Conference on Data Mining (ICDM)* (pp. 452–461). https://doi.org/10.1109/ICDM50108.2020.00054

Samadi, M., Mousavian, M., & Momtazi, S. (2021). Deep contextualized text representation and learning for fake news detection. *Information processing & management 58*(6). https://doi.org/10.1016/j.ipm.2021.102723

Shu, K., Zheng, G., Li, Y., et al. (2020). Early detection of fake news with multi-source weak social supervision. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020*, Ghent, Belgium, Sep. 14–18, Proceedings, Part III, https://doi.org/10.1007/978-3-030-67664-3_39

Singh, V. K., Ghosh, I., & Sonagara, D. (2021). Detecting fake news stories via multimodal analysis. *Journal of the Association for Information Science and Technology, 72*(1), 3–17. https://doi.org/10.1002/asi.24359

Varma, P., & Ré, C. (2018). Snuba: Automating weak supervision to label training data. In: Proceedings of the VLDB Endowment. *International Conference on Very Large Data Bases*, (p. 223). https://doi.org/10.14778/3291264.3291268

Wang, Y., Yang, W., Ma, F., et al. (2020). Weak supervision for fake news detection via reinforcement learning. In: *Proceedings of the AAAI conference on artificial intelligence*, (pp. 516–523). https://doi.org/10.1609/aaai.v34i01.5389

Wu, R., Chen, S. E., Zhang, J., et al. (2023). Learning hyper label model for programmatic weak supervision. https://doi.org/10.48550/arXiv.2207.13545

Yuan C., et al. (2020) Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. In: *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online), (pp. 5444–5454). https://doi.org/10.18653/v1/2020.coling-main.475

Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys, 53*(5), 1–40. https://doi.org/10.1145/3395046

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.