EARLY DETECTION OF FAKE NEWS ON EMERGING TOPICS THROUGH WEAK SUPERVISION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SERHAT HAKKI AKDAĞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2023

Approval of the thesis:

EARLY DETECTION OF FAKE NEWS ON EMERGING TOPICS THROUGH WEAK SUPERVISION

submitted by **SERHAT HAKKI AKDAĞ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Nihan Kesim Çiçekli
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering, METU _____

Assoc. Prof. Dr. Hacer Yalım Keleş
Computer Engineering, Hacettepe University _____

Date:07.09.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Serhat Hakkı Akdağ

Signature :

ABSTRACT

EARLY DETECTION OF FAKE NEWS ON EMERGING TOPICS THROUGH WEAK SUPERVISION

Akdağ, Serhat Hakkı
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Nihan Kesim Çiçekli

September 2023, 80 pages

In this thesis, we present a novel solution to the early detection of fake news problem on emerging topics through weak supervision. Traditional techniques rely on fact-checkers or supervised learning with labeled data, which is not readily available for emerging topics. To address this, we introduce end-to-end Weakly Supervised Text Classification framework, WeSTeC, to programmatically label a large-scale text dataset of a particular domain and train supervised text classifiers with the assigned labels. The proposed framework combines multiple weak labeling strategies and aggregates the generated weak labels into a single weak label per data instance. The generated labels are then used to fine tune a pre-trained RoBERTa classifier for fake news detection. By using the weakly labeled dataset containing fake news related to the emerging topic, the trained fake news detection model becomes specialized for the topic at hand. We consider both semi-supervision and domain adaptation setups, utilizing small amounts of labeled data and labeled data from other domains respectively. The proposed model is evaluated on both the quality of aggregated weak labels generated and the fake news detection classifier. In both evaluations, the model outperforms all baselines in each setup considered. In addition, when compared to the

fully supervised counterpart, the fake news detection model trained on weak labels achieves an accuracy as close as 0.1%, showing the effectiveness of the weak labeling module of the proposed framework.

Keywords: Fake News Detection, Weakly Supervised Learning, Text Classification, Language Models

ÖZ

YENİ ORTAYA ÇIKAN KONULAR ÜZERİNDE ZAYIF DENETİM YOLUYLA SAHTE HABERLERİN ERKEN TESPİTİ

Akdağ, Serhat Hakkı
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Nihan Kesim Çiçekli

Eylül 2023 , 80 sayfa

Bu tezde, zayıf denetim yoluyla ortaya çıkan yeni konularda sahte haberlerin erken tespiti sorununa yönelik yeni bir çözüm sunuyoruz. Geleneksel teknikler, etiketli verilerle denetimli öğrenmeye veya teyit organizasyonlarına dayanmaktadır. Bu yöntemler yeni ortaya çıkan konular için hazır olarak bulunmamaktadır. Bunu çözmek amacıyla, belirli bir alana ait büyük ölçekli bir metin veri kümesini programlı olarak etiketlemek ve atanan etiketlerle denetimli metin sınıflandırıcılarını eğitmek için WeSTeC'i (Zayıf Denetimli Metin Sınıflandırma çerçevesi) sunuyoruz. Önerilen çerçeve, birden fazla zayıf etiketleme stratejisini birleştirir ve oluşturulan zayıf etiketleri tek bir birleştirilmiş zayıf etikete dönüştürür. Oluşturulan etiketler daha sonra sahte haber tespiti için önceden eğitilmiş RoBERTa sınıflandırıcısını ince ayarlamak için kullanılır. Zayıf etiketli veri kümesindeki sahte haberlerin yeni konuyla ilgili olduğu göz önüne alındığında, eğitilmiş sahte haber tespit modeli eldeki konuya özelleşir. Bu çalışmada yarı denetimli ve alan uyarlama kurulumlarınını ele alıyoruz. Bunlar sırasıyla az miktarda etiketli veri ve diğer alanlardaki etiketli veriyi kullanır. Önerilen modelin değerlendirmesi, oluşturulan birleştirilmiş zayıf etiketlerin kalitesi ve sahte

haber tespit sınıflandırıcısı üzerinde yapılır. Her iki değerlendirmede de, tüm temel yöntemlerden daha iyi performans gösterir. Ayrıca, tamamen denetimli olarak eğitilen sahte haber tespit modeli ile karşılaştırıldığında, zayıf etiketlerle eğitilen model doğruluk açısından yüzde 0.1'e kadar yakın sonuçlar verir. Bu da önerilen çerçevenin zayıf etiketleme modülünün etkinliğini göstermektedir.

Anahtar Kelimeler: Sahte Haber Tespiti, Zayıf Denetimli Öğrenme, Metin Sınıflandırması, Dil Modelleri

To my family

ACKNOWLEDGMENTS

I am deeply grateful to my advisor Prof. Dr. Nihan Kesim Çiçekli for her invaluable guidance and profound expertise throughout the entirety of my thesis journey. Her dedication, patience, and encouragement were instrumental in keeping me motivated and focused during challenging times.

I want to express my heartfelt gratitude to Defne for her unwavering support, tireless belief and encouragement in all my endeavors. Her support has been instrumental in my academic success, and I am truly grateful to have her by my side.

I would like to express my sincere appreciation for my family including my parents Esra, İbrahim and my two brothers Semih and Melih. I am forever grateful for their support, guidance and the countless sacrifices they have made to help me succeed.

I am grateful to TrustLab for introducing me to the field of Trust and Safety, showing the importance to making the web a better place. Their introduction played a huge part on my selection of research area, fake news detection. I would also like to thank for their support on providing access to cloud services that facilitated technical requirements of my study. Their invaluable support and collaboration have greatly enriched my research journey.

Last but not least, I would like to acknowledge my friends Mert, Kaan and Alperen for their constant support on my thesis journey.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Networks
GNN	Graph Neural Networks
IDF	Inverse Document Frequency
LF	Labeling Function
LDR	Labeled Data Ratio
LIWC	Linguistic Inquiry and Word Count
LR	Logistic Regression
MLP	Multi-Layer Perceptron
NB	Naive Bayes
POS	Part-of-Speech
RF	Random Forest
RoBERTa	A Robustly Optimized BERT Pretraining Approach
SME	Subject Matter Expert
SLP	Single-Layer Perceptron
TF	Term Frequency
WandB	Weights and Biases

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

With the rise of the internet, the way people consume news changed significantly. Individuals are able to reach news on any subject from different sources easily, which seems great at first glance. However, it also introduces challenges, one of which is the detection of fake news. The sources on the open web can include both reputable outlets as well as outlets that disseminate fabricated and fake information. To make sure readers are protected against fake news, it is significant to be able to distinguish between these two.

There are a couple of strategies commonly utilized to make this distinction. One of these approaches is making use of fact-checkers. Fact-checkers are credible individuals or organizations that verify the correctness of news stories and other forms of information. They use different techniques to achieve this, including researching primary sources, consulting with experts, and analyzing data. One can understand the trustworthiness of a news article by matching it to a fact-check article if there is one available. According to the agreements and disagreements between the news article and a matching fact-check article, a decision can be made. Another common approach to separate fake and real news is through supervised machine learning. Given a dataset where each news article is identified as fake or real, models can be trained to distinguish between these on the unseen news articles. Recent deep learning approaches proved successful in text classification, including transformers architectures trained with a high amount of labeled data. Such methods are also frequently employed to combat the fake news problem. There are a great number of studies digging

into this, which is further analyzed in section 2.3.4.

However, both of the mentioned techniques fail to solve the problem of early detection of fake news on emerging topics. When a new claim or topic appears, with the help of social media and open web, it spreads rapidly. Fact-checkers take time to investigate new claims, and by the time they successfully do, the claim can reach a large audience. Similarly, labeled datasets consisting of data samples on this particular topic or claim can't be found in the early stages of dissemination due to manual annotation of data taking excessive time. Considering these, it is important to explore different approaches for this issue. In this thesis we aim to tackle the particular problem of early detection of fake news on emerging topics.

Weak supervision is a branch of machine learning that aims to utilize noisy, lower quality weak labeling sources when there is unavailability of labeled data. Some of the noisy weak labeling sources can be listed as using high level inputs from domain experts, programmatic scripts, cheaper annotators or small amounts of labeled data. Through weak supervision, larger-scale training sets can be automatically constructed to then continue with traditional supervised machine learning approaches. We seek to pursue a solution to the early detection of fake news on emerging topics via weakly supervised learning. We believe if adequate labels to data instances of news articles on emerging topics can be programmatically assigned, then, one can utilize supervised machine learning approaches that proved successful in other text classification settings.

In the first part of this thesis, we introduce the Weakly Supervised Text Classification framework, **WeSTeC**, to programmatically label a large-scale text dataset of a particular domain and train supervised classifiers with the assigned labels. We utilize the created framework in two different fake news classification settings. First one is the semi-supervision, where there are small amounts of labeled fake news articles that are on the same domain with the target large-scale dataset, mentioning the same emerging topic. We experiment with a setup where the number of labeled data instances are less than 0.7% of the unlabeled large scale dataset. The second setting we experiment on is the domain adaptation setup where we have labeled data for fake news articles talking about some past event(s). Here, we aim to programmatically assign labels to a

large-scale dataset of a different domain, containing news on an emerging topic. We believe both settings are important to consider separately. Initially, there is no labeled data and through manual annotation of a small number of fake news articles focusing on an emerging event, we want to label a vast amount of news articles. This way, we can utilize the supervised learning methods that take advantage of a large number of data instances. In real world scenarios, enterprises have access to manually annotated labeled data on previously occurred events. The second setting aims to utilize available labeled data from other domains and previous events to automatically assign labels to data instances of an emerging topic. This way, one can detect fake news of emerging events even without obtaining small amounts of labeled data.

In the second part of the proposed framework, we use the programmatically labeled large-scale dataset to build supervised machine learning models. The aim of this part is to achieve generalization beyond assigned labels where the supervised end models trained on weak labels can get even higher classification scores when tested against actual labels. Our framework achieves higher accuracies compared to all benchmark weak supervision models currently available on the fake news detection problem. We further discuss our results and evaluation against the state-of-the-art weakly supervised fake news detection algorithms in Chapter 4.

1.2 Proposed Methods and Models

We introduce WeSTeC, an end-to-end framework to perform text classification on large-scale unlabeled datasets through weakly supervised learning. Proposed framework consists of two distinct modules, the weak labeling module and the text classification module. The weak labeling module is responsible for assigning an aggregated weak label per data instance in the unlabeled dataset. The text classification module uses the generated weak labels to train supervised machine learning models that have proven useful in various text classification settings. This module aims to generalize beyond the aggregated weak labels generated using the weak labeling module, when tested against the actual labels. WeSTeC provides an end-to-end solution for both labeling an unlabeled dataset through weak supervision strategies and easily using these labels to train state-of-the-art supervised text classification algorithms. In this

work, the proposed framework is particularly used to solve the problem of early detection of fake news on emerging topics. By using weak supervision strategies, we label large scale news article datasets on emerging topics. Then, we use the assigned weak labels to train supervised machine learning models that are specialized on the emerging topic at hand thanks to the aggregated weak labels.

The weak labeling module of the proposed framework takes two inputs; a labeled dataset and an unlabeled dataset. It uses weak supervision techniques to label the latter dataset, through the information that it learned and optimized from the former dataset. WeSTeC is utilized to test against both semi-supervision and domain adaptation settings to perform fake news classification on emerging topics. In the semi-supervision setting, both datasets provided to the weak labeling module are of the same domain, however, the labeled dataset contains only a small number of data instances. In the domain adaptation setting, input datasets contain data instances from two different domains. In both settings, the weak labeling module generates a single aggregated weak label for each data instance of the unlabeled dataset at the end of the execution.

Internally, the module generates weak labeling signals using multiple approaches. These signals are represented through labeling functions in the system. Labeling functions can be defined as noisy, programmatic rules and heuristics that assign labels to unlabeled training data.[1] [1] Using the labeling functions, WeSTeC is able to capture noisy weak signals obtained via two different approaches and easily aggregate them. We refer to these groups of labeling functions as content-based and module-based labeling functions. Content-based labeling functions leverage stylistic, complexity and readability measures based on different text columns of the datasets provided. For the news articles case, these are title and content. The module takes advantage of the labeled dataset to determine feature thresholds, which are then used to generate content based labeling functions for the identified content features. WeSTeC automatically selects the best content features in the threshold selection algorithm, allowing users to introduce as many content features as possible to the framework. We explain the currently supported content-based labeling functions and how threshold selection process works in section 3.3.1.1. On the other hand, model-based labeling

[1] https://www.snorkel.org/use-cases/01-spam-tutorial

functions utilize machine learning models that have been trained through the provided labeled data set. When applied to unlabeled dataset, these models do not provide strong enough predictions to be used as standalone ground truth labels. However, the framework utilizes them as weak labeling signals.

As the next step, the weak labeling module applies the captured labeling functions to the instances of the unlabeled large-scale dataset. Then, for each data instance, it combines the weak labels obtained, without having access to ground truth labels. Multiple aggregation techniques are supported as part of WeSTeC. In addition, the proposed framework provides a data selection layer. Probabilistic aggregated weak labels generated by some of the aggregation strategies can be utilized to select data instances where the aggregator is most confident without having access to any ground-truth labels. In the end, the weak labeling module outputs an aggregated weak label for each data instance in the unlabeled dataset. We evaluate the quality of aggregated weak labels by comparing them with the actual labels. Our aggregated weak labels achieve higher accuracy than all baseline weakly supervised fake news detection algorithms that automatically assign and aggregate weak labels. The comparison results are presented in section 4.5.1.

As the final step, the weakly labeled dataset is passed to the text classification module to train state-of-the-art supervised text classification algorithms. Since the weakly labeled dataset consists solely of news articles on the emerging topic, the text classifier trained at this stage is specialized in handling news in this topic. We evaluate performances of the trained models on actual labels. In addition to this, we train a classifier with the exact same setup using the actual ground-truth labels to be able to compare how close the model trained with weak labels can get to the model trained in fully supervised setup. In both semi-supervision and domain adaptation setups, our model achieves accuracies as close as 1-2% to the model trained in fully supervised setup. In addition, our model outperforms all baseline weakly supervised fake news algorithms. We present the detailed results and comparison with the existing weakly supervised fake news detection algorithms in section 4.5.2.

1.3 Contributions and Novelties

Our contributions are as follows:

- We introduce an end-to-end, weakly supervised text classification framework, WeSTeC, which enables users to easily execute weak-supervision pipelines from data labeling to text classification.
- We present a mechanism to automatically generate and select labeling functions based on content features, which eliminate the need to manually select content features.
- WeSTeC provides an infrastructure to easily combine multiple weak aggregation strategies. We introduce model-based labeling functions on top of content-based ones, which leverage machine learning models as weak labeling signals for the large-scale unlabeled dataset.
- The proposed framework provides a mechanism to aggregate the weak labeling sources and generate single weak labels for each data instance. It supports multiple aggregation strategies and a data selection layer for aggregation strategies that output probabilistic labels.
- We use the proposed framework to solve the early detection of fake news on emerging topics problem. We focus on both semi-supervised and domain adaptation scenarios. Thanks to the generic and extendible structure of the proposed weakly supervised text classification framework, we seamlessly test with both setups.
- On both domain adaptation and semi-supervision setups, aggregated weak labels generated by the proposed framework outperform all baseline weakly supervised fake news detection algorithms that programmatically assign weak labels.
- Fake news detection classifiers trained with aggregated weak labels in both semi-supervision and domain adaptation settings outperform state-of-the-art weakly supervised fake news detection algorithms.

1.4 The Outline of the Thesis

Chapter 2 provides a research background for our study. We first introduce text classification, weakly supervised learning and language models. We then dive into the specific problem of fake news classification, which is a subset of text classification. We introduce characteristics of fake news, both supervised and weakly supervised approaches taken in timely detection of fake news task and use of language models in the area.

Chapter 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 which is responsible for programmatically assigning weak labels and using the generated labels to train fake news detection models.

Chapter 4 presents the experiments conducted using the proposed framework. We first explain the dataset and metrics used in the experiments. We then delve into the experimentation setups for both domain adaptation and semi-supervision settings. We provide results for each step of the pipeline introduced. We also compare our results with the existing state-of-the-art weakly supervised fake news detection studies.

Chapter 5 summarizes the proposed model and the results obtained. We highlight the important achievements made in this study. Finally, we outline areas that need further exploration in the future work section.

CHAPTER 2

LITERATURE SURVEY

In this chapter, we provide a literature survey on the studies related to our work. We begin by introducing text classification, which encompasses the aim of this study, fake news detection. We go over both traditional supervised approaches taken for text classification as well as recent breakthroughs to the area with the introduction of transformer architecture. We discuss how the transformer based language models shaped the studies around natural language processing and benefits of using them in the context of text classification. We then delve into the topic of weak supervision, explaining how it differs from supervised classification techniques. We also introduce various approaches that are used as part of weakly supervised learning.

In the fake news detection section, we focus on the specific problem addressed in this study. This section is divided into four headings. First, we offer background information on the distinctive traits of fake news, widely employed as features in studies. After that, we focus on supervised fake news detection studies which benefit from the availability of ground truth labels. In the third subheading, we delve into the analysis of studies that focus on weakly supervised fake news detection, comparing the approaches adopted with our solution. Finally, we mention the use cases of transformer based architectures and language models in the context of fake news detection, providing a summary of breakthrough improvements it has made to the area.

2.1 Text Classification

Text classification, also known as text categorization, is the task of assigning predefined categories or labels to textual documents. It is an area containing many sub-

fields, including but not limited to spam filtering, sentiment analysis, topic classification. The main focus of this study, fake news classification, can also be defined as a text classification task where the end goal is to assign fake or real categories to the news articles.

In his study on text classification, Sebastiani [2] provides a brief history of how text classification was handled. In the 80s, rule-based simple approaches were mainly utilized whereas starting from the 90s, text classification started to be handled through machine learning approaches. There is a necessary step of identifying features from text documents to be able to train machine learning classifiers for text classification. Since the text is not simply the sum of its individual words, it is important to deduce sentence or word representations that can capture the semantic and syntactic information present in text. Mikolov et al. [3] introduced the Word2Vec model that represents words as dense, continuous vectors that can capture semantic relationships and analogies between words. Their work led to improvements in various NLP tasks, including text classification. In a similar study, Pennington et al. [4] introduced GloVe, which stands for Global Vectors for Word Representation. Different from the other word representation studies, they leveraged global word-occurrence statistics to generate the word representations. GloVe embeddings have been widely adopted in text classification and other NLP tasks.

Advancements in deep learning played a significant role in the improvements on text classification systems. In 2014, Kim [5] introduced the use of convolutional neural networks (CNNs) [6] for sentence level text classification, proving that CNNs could effectively capture local patterns and hierarchical features in text. Kim achieved state-of-the-art results on various classification tasks. Even though there were many successful studies showing improvements in text classification tasks, the introduction of the Transformer model by Vaswani et al. [7] revolutionized how we approach the natural language processing tasks. Transformers leverage self-attention mechanisms to capture global dependencies and encode contextual information. Models built to tackle text classification tasks on top of transformer architecture have achieved state-of-the-art performance on various text classification benchmarks. One example of such models can be given as Bidirectional Encoder Representations from Transformers (BERT), introduced by Devlin et al. [8] in 2017. BERT is an advanced

language model that leverages the transformer-based architectures to learn contextualized representations of words or tokens in a large corpus of text data. BERT has had a profound impact on a wide range of natural language processing tasks, including text classification. Authors argue that it obtains new state-of-the-art results on eleven NLP tasks, with accuracy improvements ranging from 1.5% to 7.7%. Similar to BERT, many self-training methods are introduced on top of the Transformer architecture idea. Some of the most influential ones can be listed as ELMo [9], GPT [10], XLM [11] and XLNet [12]. Liu et al. [13] recognize the performance gains achieved by all these self-training methods. However, they also argue that it can be challenging to determine which aspects of the methods contribute to the performance gains the most. To overcome this challenge, they present a replication study of BERT pretraining, which includes a careful evaluation of the effects of hyperparameter tuning and training set size. They propose an improved recipe to BERT pretraining, which achieves state-of-the-art results on three baseline text classification tasks compared to all self-training methods listed. We discuss the use of RoBERTa on fake news detection tasks in section 2.3.4.

2.2 Weak Supervision

Supervised learning techniques build predictive models by learning from a large number of training instances where each instance is associated with a ground-truth label. [14] Text classification techniques building on top of the supervised learning paradigm showed great success, including the deep learning improvements and recent transformer architecture introduction. However, there is an important challenge associated with the supervised learning: the availability of ground-truth labels for large scale datasets. Data labeling process is costly, especially for areas that require subject matter experts (SMEs). Due to this difficulty, weak supervision is introduced. Weak supervision refers to the process of using weak signals like noisy or imperfect labels to train machine learning models. With weak supervision, one can leverage large amounts of unlabeled data, even though the labels introduced are not as accurate as the traditional supervised learning ground-truth labels.

There are different types of weak supervision. Zhou [14] lists these under three cate-

gories. First one is incomplete supervision where only a small subset of the training data are labeled while the remaining is unlabeled. This is also referred to as semi-supervision. The second type is inexact supervision where only coarse-grained labels are given like having higher level labels instead of labels for each instance. The third type is defined as inaccurate supervision, where the labels are not always ground-truth. Examples of these can be programmatically assigned labels based on heuristics or simple classifiers. In our study, we build a framework to benefit from multiple types of weak supervision and easily aggregate the weak labels for each data instance. This way, we achieve better results on text classification tasks, mainly fake news detection, than only benefiting from a certain type of weak supervision signals. We delve into how our framework supports the use of multiple weak supervision strategies to achieve superior results in fake news detection task in section 3.3.

Ratner et al. [1] introduce Snorkel, a system that enables users to train state-of-the-art models without hand labeling training data. To achieve this, they introduce labeling functions (LFs), which are used to express rules used to assign weak labels to training data. One of the many benefits of LFs is that they enable encapsulating weak labeling sources in a unified way. Many different weak labeling strategies can be represented through labeling functions. Our framework utilizes the labeling functions to easily represent weak labeling sources in the system, which are of two different types. As part of Snorkel, authors also introduce ways to aggregate weak labels. They argue that a weak label aggregator can be trained without having access to ground truth labels. The aggregator simply learns from agreements and disagreements of labeling functions when applied to many unlabeled training instances. In our framework, WeSTeC, we build on the idea of Snorkel, using the power of labeling functions and aggregation strategies as part of the end-to-end weakly supervised text classification pipeline. Our solution is not a generic weak supervision tool but instead it is an end-to-end pipeline that makes the text classification tasks through weak supervision easier. We focus on the details of the WeSTeC in section 3.1.

Similar to Snorkel, there are other weak supervision frameworks. One noteworthy framework is Snuba, introduced by Varma et al. [15] Snuba is another framework, built on top of the idea of Snorkel, however, more solely focusing on the semi-supervision techniques. Using a labeled subset of large-scale dataset, Snuba can learn

heuristics and generate labeling functions to label the remaining large-scale dataset. They show that it outperforms the other semi-supervised approaches. Our framework also has capabilities to generate labeling functions through semi-supervised approaches. However, in addition to this, WeSTeC supports generating labeling functions through other types of weak supervision, specialized on the text classification task. In addition to this, WeSTeC is not designed only to support semi-supervision; we experiment with it in both semi-supervision and domain adaptation setups.

2.3 Fake News Detection

2.3.1 Characteristics of Fake News

Fake news detection refers to the process of classifying news articles, information or media contents that are misleading or deceptive. The term fake news is often used as an umbrella term that covers both misinformation and disinformation. Misinformation refers to false or inaccurate information, regardless of the intent. Disinformation however is defined as false information which is deliberately intended to mislead. [1]

The aim of this study is to use weak supervision techniques to address the problem of early detection of fake news. In order to effectively detect and combat fake news, it is crucial to understand the key characteristics differentiating fake news from the credible ones. Horne and Adali. [16] argued that fake news is assumed to be written to look like real news, fooling the reader who does not check for reliability of the sources. They studied how fake and satire news distinguish from real news by looking at content features. They 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. The authors use Linguistic Inquiry and Word Count (LIWC) [17] dictionaries to measure these features. They highlight which features are best to distinguish fake and real news in their work, concluding that the content of fake and real

[1] https://www.apa.org/topics/journalism-facts/misinformation-disinformation

news articles are substantially different. We use the analysis made in this paper to select features that the content-based labeling functions utilize as part of the proposed framework. However we do not use the LIWC based features in our study, because the LIWC dictionaries are not publicly available.

Ngada and Haskins [18] use content-based features to train machine learning models. They validate the effectiveness of some of the features also introduced by Horne and Adali. [16] In addition to this, they show the effectiveness of measuring punctuation and part of speech (POS) tagging based features to distinguish between fake and real news articles. Similarly, Qin et al. [19] shows the effectiveness of POS-tagging features in the context of fake news detection in their work. In another study, Rubin et al. [20] proves the potency of punctuation features when detecting misleading 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. Castelo et al. [21] introduces a topic-agnostic approach for identifying fake news, using many of the content features commonly employed. They show the effectiveness of their approach on different domains, stating that their work shows promising results beyond political news domain on which they trained. Our aim is to address the challenge of early fake news detection in a domain adaptation setting, using content and model based labeling functions. The efficacy of content features in domain adaptation setup establishes a strong foundation for the approach we adopt in the proposed framework.

Various studies explore the use of features other than the ones extracted solely through the content of the news articles. Rastogi and Bansal [22] show that different categories of features can be employed to combat fake news. They group the features in four distinct categories; knowledge, user, content and propagation. Knowledge category encompasses features obtained through fact-checking and manual labeling. The user category consists of features extracted from the writer of the news articles including but not limited to the number of followers, number of posts, geo-location, etc. Finally, propagation features are based on the interaction of information on social media context like replies, shares, likes and propagation speed. Bondielli and Marcelloni [23] use both content and context based features to combat fake news. Similarly, Zhang and Ghorbani [24] point out the use of features based on news content as well as

social context, providing a guideline to researchers focused on solving the problem of fake news detection. Although the use of non-content features showed success on these studies, they are not applicable for solving the early detection of fake news problem on emerging topics. In our study, we explore ways to detect fake news on emerging topics before they propagate on the web. Using propagation information or knowledge features requires time and cannot be used as part of early fake news detection strategies for emerging topics. In addition, we focus on news articles as opposed to user generated social media content. The user information for news article authors, like number of posts, geolocation, registration age, etc. are not available on different news outlets, making it hard for us to use such user features. All in all, the most effective feature category to use in early detection of fake news articles on emerging topics is content features.

2.3.2 Supervised Fake News Detection

When ground truth labels are available for a set of news articles, supervised machine learning techniques employed on many text classification studies can be utilized in the context of fake news detection. Various successful text classification studies showing state-of-the-art results through employing supervised machine learning techniques are introduced in section 2.1. There is also a considerable amount of research focusing on the effectiveness of supervised approaches in fake news detection specifically. Perez-Rosas et al. [25] developed supervised classification models that rely on a combination of lexical, syntactic, semantic and readability properties. They argue that their best performing supervised models achieved accuracies that are comparable to human ability to spot fake content. This shows the effectiveness of supervised models in fake news detection when ground truth labels are available.

Many of the earlier works focused on textual content of the news contents or user generated social media contents. In reality, online contents often consist of multiple types of media including text, pictures, videos and sound. Singh et al. [26] explore the use of multimodal analysis in fake news detection. They propose a system to combine text and visual analysis of online news stories to automatically detect fake news through supervised techniques. They show that multimodal analysis can help

improve the performance of purely textual or purely visual fake news detectors. In our study, we only focus on textual content because large-scale datasets with both visual and textual context are unavailable. However, the framework we propose can easily be extended to combine visual content based labeling functions in future studies.

2.3.3 Weakly Supervised Fake News Detection

Although supervised techniques used in fake news detection can achieve accuracies comparable to human ability to detect fake content, they require labeled data for training. This makes them unusable for the early fake news detection problem on emerging topics. There are considerable number of studies focused on the use of weak supervision techniques to combat the early fake news detection problem. Raza and Ding [27] explore the use of news content and social contexts in the early fake news detection problem. They propose an effective automated labeling technique to address the ground-truth label problem. Then through a model based on Transformer architecture, the proposed system learns useful representations from the fake news data for the decoder part to predict the future behavior based on past observation. Since their solution also utilizes social contexts, it requires the content to propagate even if only to a small extent. Their approach to automated labeling takes advantage of only three labeling functions. On the other hand, in our study, we utilize up to 144 labeling functions automatically generated and optimized, resulting in higher quality weak labels after aggregation. This assists our supervised end models trained through weak labels to achieve higher accuracies when tested against the actual labels.

Ozgobek et al. [28, 29] outline a weakly supervised fake news detection schema that is used as one of the baseline approaches to our work. Their model relies solely on content features to create labeling functions to weak label data instances. They utilize Snuba to apply and aggregate weak labels in semi-supervised setup. Specifically, their content feature threshold selection algorithm is considered as part of WeSTeC, although it has been improved for our specific use case. We add feature selection capabilities to the threshold search algorithm, enabling it to eliminate low performing labeling functions based on content features, without using ground-truth labels. Further details on the threshold selection algorithm can be seen in section 3.3.1.1.

Their work focuses solely on content features, whereas our framework can utilize different types of weak supervision strategies and combine their labels. In addition, our framework supports multiple setups, including the ones we explore in this study; semi-supervision and domain adaptation. This is possible thanks to the introduction of WeSTeC, an end-to-end weakly supervised text classification framework. Ozgobek et al. [28] do not focus on fine tuning the text classification model to a dedicated emerging topic. Our study however focuses on early detection of fake news of a particular topic. We achieve superior performance on both weak label aggregation and text classification evaluation steps. More detailed evaluation can be seen in section 4.5.

Several other studies focus on the semi-supervised setting to combat the early detection of fake news problem. Shu et al. [30] jointly leverage a limited amount of clean labeled data and a large amount of unlabeled data weakly labeled through social engagements. They train deep neural networks in a meta-learning framework with the combined dataset. They show that their model outperforms state-of-the-art baselines without using any user engagements at prediction time. However, their method requires social engagement data for training data, which makes their solution less usable for early detection of fake news on emerging topics. Their method is useful for early detection of fake news in known domains or topics. The framework we propose focuses on timely identification of fake news for unknown topics through domain-agnostic features leveraged in automatically generated labeling functions and fake news classification model trained specifically for an emerging topic.

Dong et al. [31] propose a two path semi-supervised learning framework for timely fake news detection. Their solution has two CNN paths, one for supervised learning with few labeled data and another with a huge amount of unlabeled data. Our study shows better results even though we experiment with lower labeled data ratio on the semi-supervised setting. In a similar study, Konkobo et al. [32] propose a three path CNN based deep learning model for early detection of fake news on social media. They show promising results when the labeled data ratio is 25% of the initial dataset. In contrast, we use less than 0.7% labeled data and achieve better accuracies. Detailed evaluation of existing state-of-the-art semi-supervised early identification of fake news systems with our proposed system can be seen in section 4.5.2.

Ren et al. [33] introduce a novel approach to combat fake news in early stages of dissemination. They propose a system to detect fake news in a heterogeneous information network. To solve the problem of scarcity of labeled data, they utilize active learning, continuously querying high-value candidate nodes for classifier training and tuning. This way, they achieve high performance even with a small amount of labeled data. Even though they show a novel approach to solve timely fake news detection in a semi-supervised setting, their solution requires social context to obtain the information network and our framework shows better results in a semi-supervised setting.

Li et al. [34] focus on the domain adaptation setup, showing effectiveness of multi-source domain adaptation in early fake news detection. They use domain-agnostic features to weakly label the dataset of the target domain. They utilize three labeling functions, focusing on stylistic and POS tagging content features. In addition, they train source-specific fake news classifiers by fine tuning models for the target domain. Their model outperforms baselines they compare against. Our study also focuses on the domain adaptation setting with weakly supervised strategies. Compared to the limited number of labeling functions in their work, our framework generates up to 144 labeling functions, resulting in better aggregated weak label quality with the help of feature selection capabilities. We provide detailed comparison in section 4.5.1.

2.3.4 Use of Language Models in Fake News Detection

Advancements in natural language processing tasks with the introduction of transformer architectures had a significant impact on fake news detection solutions. Samadi et al. [35] explore the use of deep contextualized text representations through language models to tackle the problem of fake news detection. The model they propose utilizes a deep contextualized representation embeddings provided by novel pre-trained models such as BERT [8], RoBERTa [13], GPT2 [10] and Funnel Transformer [36] combined with Single-Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), and CNN classifiers. Their models show superiority over existing state-of-the-art models with increase in classification accuracy ranging from 0.1% to 7% improvements. In our study, we explore fine tuning pre-trained RoBERTa [13] classifier on weakly labeled news dataset of emerging topic.

Various studies show the effectiveness of supervised fine tuning of pre-trained language models in text classification tasks. Kant et al. [37] seek a solution for an NLP task, namely sentiment classification. They use a supervised approach to fine tune pretrained language models to build a classifier. They demonstrate that the fine-tuned model outperforms general purpose commercially available APIs for sentiment and multidimensional emotion classification on the same dataset. Gasparetto et al. [38] provide a survey of text classification algorithms in their study. They show the dominance of transformer-based language models in all text classification tasks, highlighting the importance of transformer architecture in natural language processing tasks once again. Nonetheless, they acknowledge the challenges of language models, specifically the extensive number of parameters that must be loaded in memory to perform training. However, novel pre-trained language models that can be fine tuned in lower resource settings provide a solution to this issue. All these studies prove the effectiveness of fine-tuning pre-trained novel language models. Our approach differs from the studies that directly focus on utilizing supervised setting to fine tune pre-trained language models. Instead, our model programmatically generates aggregated weak labels which are then used in the fine-tuning of the language model. We demonstrate the effectiveness of this approach in the early fake news detection task.

CHAPTER 3

WESTEC FRAMEWORK

In this chapter, we introduce our proposed end-to-end framework for Weakly Supervised Text Classification called WeSTeC. We first provide an overview of the system architecture and describe the key components of the framework. In the next part, we explain two different setups that we consider in our work, semi-supervision and domain adaptation. Then, we describe the weak labeling module of the proposed framework under three separate headings. In the first one, we explain the labeling function creation process, including two different types of labeling functions; model-based and content-based. In the next part of the weak labeling module, we discuss the labeling function application process. As the last part of weak labeling module, we disclose the step where labeling function outputs are aggregated to obtain one final combined weak label for each data instance. Finally, we go over the text classification module of WeSTeC and how we utilize it in fake news classification settings. Here, we highlight the advantages of utilizing a large-scale dataset of an emerging topic with weak labels to train end models capable of leveraging the abundant labeled data instances.

3.1 Proposed Framework

In this thesis, we aim to address the early detection of fake news on emerging topics problem by considering semi-supervision and domain adaptation setups. In order to be able to test both of these setups with different variations at every step, we introduce an end-to-end weakly supervised text classification framework (WeSTeC). The existing technology landscape is scattered when it comes to weak supervision and

there is no solution to easily 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 popularly used such as Snorkel[1] [1], Hyper Label Model [39], SimpleTransformers[2], etc.

3.1.1 Overall Architecture

The diagram 3.1 shows the overall architecture of WeSTeC. The proposed framework consists of two main modules: weak labeling and text classification. The purpose of the weak labeling module is to programmatically assign aggregated weak labels to a large-scale unlabeled dataset. The text classification module is used to train supervised text classification models utilizing the aggregated weak labels generated through the weak labeling module. By providing both these modules, WeSTeC is able to execute a weakly supervised text classification pipeline end-to-end. The output of the whole pipeline is a model that is trained/fine-tuned using the large-scale unlabeled dataset provided to the framework as an input, utilizing the aggregated weak labels. Therefore, the trained model is a specialized model on the domain of the unlabeled input dataset. In the fake news classification task, the trained model is specialized on the news articles that mention the emerging topic, benefiting from this particular feature of the WeSTeC.

The weak labeling module is responsible for assigning aggregated weak labels to the data instances of the provided unlabeled dataset. It achieves this by combining multiple weak labeling approaches and using an additional input dataset that includes the ground truth labels. The framework uses labeling functions to represent each weak labeling source. Two different weak labeling approaches are supported by WeSTeC: content-based labeling functions and model-based labeling functions. Content-based labeling functions use the content features that explain stylistic, complexity and readability metrics of the text columns of the provided datasets. Using the provided labeled dataset, the module learns the feature thresholds for each identified content feature. By using the features and the thresholds, it then generates labeling functions

[1] https://www.snorkel.org/
[2] https://simpletransformers.ai/

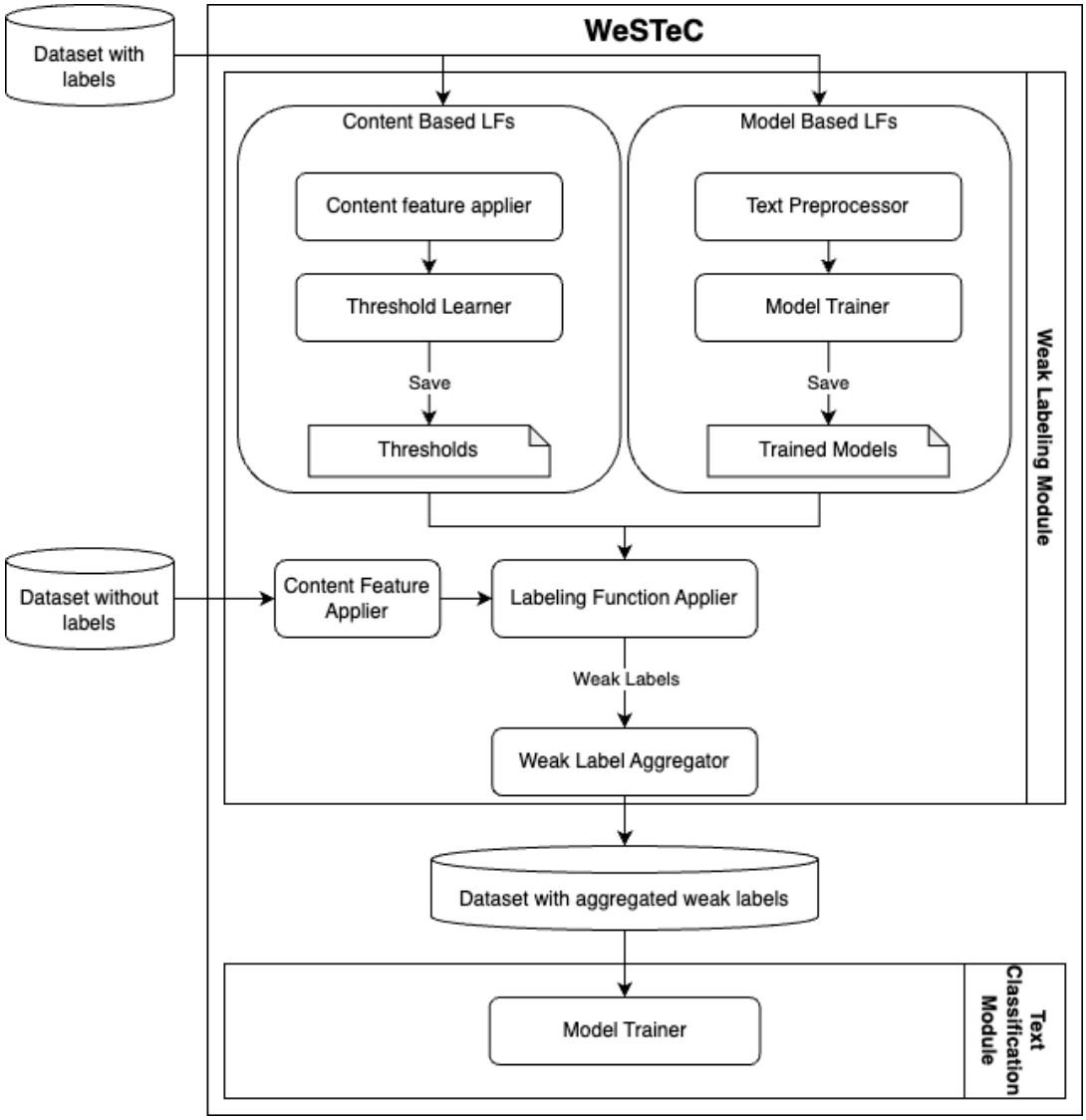

Figure 3.1: Overall architecture of WeSTeC

that are used to assign weak labels to the unlabeled large-scale dataset. We explain the details of generating content-based labeling functions in section 3.3.1.1. Model-based labeling functions utilize models trained on the dataset with ground truth labels. Each of the trained models is used to create a labeling function. When applied to the unlabeled dataset, weak labels generated through these labeling functions are not strong enough to be used as a final weak label directly. However, WeSTeC utilizes these labeling functions as one of many weak labelers and therefore is able to generalize beyond them. We further delve in the details of model-based labeling functions in section 3.3.1.2.

Once all labeling functions and related resources are prepared, the weak labeling module proceeds to the stage of applying the labeling functions. It applies all labeling functions to each instance of the unlabeled dataset, generating multiple weak labels for each data instance. Then, the weak label aggregator combines these weak labels into a single aggregated weak label. WeSTeC supports three different weak label aggregation strategies. We explain these in section 3.3.2.1. At the end of the weak labeling module, the framework outputs the dataset without explicit labels but with aggregated weak labels attached to each data instance. Then, the text classification module uses these weak labels to train supervised text classification models. If ground truth labeled data is available, the text classification module can evaluate the performance of the trained end models by testing against the actual labels of the large scale unlabeled dataset. Currently, WeSTeC supports the RoBERTa text classifier as a text classification algorithm. We provide further details on the text classification module in section 3.4.

3.2 Setups

In the early stages of an emerging topic disseminating through social media, there are no labeled datasets available to train supervised machine learning models. Therefore it is important to consider weakly supervised approaches. We consider two different setups to investigate this problem, inspired by real world scenarios. These are semi-supervision and domain adaptation. WeSTeC is built to seamlessly accommodate both of these setups.

Semi-Supervision In the first setup, we have small amounts of labeled data and we aim to label a large-scale dataset of the same domain by using it. In the base case, assuming no labeled data is available for either the emerging topic or previous events and topics, semi-supervision techniques can be utilized. Limited labeled data can be obtained through manual labeling. Many of the early fake news detection studies focus on similar setups where the ratio of labeled data instances to unlabeled data instances range from 1% to 25%. [28, 31, 32] We consider the labeled data ratio to be less than 0.7% in our experiments. In addition, WeSTeC is capable of easily covering this setup. Assuming we select an emerging topic to work with such as “COVID

related news articles", we can then separate a small subset of the labeled dataset as the first input to the framework. The rest of the data instances can be provided to the framework without their labels where the framework attaches aggregated weak labels to this substantially bigger piece of the starting dataset automatically.

Domain Adaptation In this setup, we have news articles on a specific domain, where there are no ground truth labels. This domain is viewed as an emerging topic discussed on the open web where there are no labeled datasets around. In real world scenarios, labeled data for different domains and past topics accumulate over time at the hands of enterprises. In domain adaptation setup, we aim to use those labeled data instances to programmatically label data instances of emerging topics. Similar to the previous setting, two different datasets can be provided to the framework. In this scenario, the module assigns weak labels to the dataset containing news articles on the emerging topic using the labels provided in the other dataset, which contains news articles from different domains along with their attached labels.

3.3 Weak Labeling

3.3.1 Generation of Labeling Functions

The main goal of weak supervision is to combine noisy weak labeling sources that would not be sufficient to label large scale datasets on their own. These sources are not easily unifiable, therefore, there is a need to express them in a consolidated way. We use labeling functions to express these sources. Labeling functions are arbitrary snippets of code that can encode arbitrary signals like patterns, heuristics, external data resources, noisy labels from crowd workers, weak classifiers, and more. [3] [1] In this study, the labeling functions can assign one of three options to each of the news articles: fake, real or abstain. When abstain is assigned, it means that this particular labeling function cannot successfully assign any of the two possible labels.

WeSTeC generates and fine-tunes labeling functions using the labeled dataset, provided as one of the inputs to the weak labeling module. Then, the finalized labeling

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

functions are used to programmatically label the dataset without labels, provided as the other input. The labeling functions can be categorized into two different groups. The first type is content-based labeling functions. Various studies explore the way fake and true news differs based on their content, including their style, language use, characteristics, complexity, etc. [16, 18, 19, 20] The labeling functions generated in this group utilizes such content features. Then, it uses the labeled data set provided as input to the weak labeling module to find out the labeling function thresholds to make it ready for labeling function application part. We explore what these thresholds are and how the threshold search algorithm looks like in section 3.3.1.1. The second type of labeling functions are model-based where the labeled dataset is used to train weak classifiers and these models are saved. Then, on the labeling function application part, these models are used to weakly label data instances of the unlabeled dataset.

3.3.1.1 Content-Based Labeling Functions

The first type of labeling functions prepared by the proposed weak labeling module is content-based. Using numerous studies that explore content differences between fake and true articles, we have selected 41 features that are considered for use in both the title and content portions of the news articles where possible. At this stage, the system introduces as many content based labeling functions as possible. In the threshold selection part, the framework automatically eliminates content-based labeling functions for which it cannot find a threshold. This enables users to add as many candidate content features as possible without worrying about downgrading the overall weak label aggregation accuracy. The employed content based features can be categorized into four classes: stylistic, POS-tagging, punctuation and readability features. The titles of news articles have smaller length, which makes some of the features not applicable to be used with titles. We define each feature and indicate where they are used under their respective categories below.

Stylistic features These are the features based on text characteristics like style, length, etc. Table 3.1 presents each utilized stylistic feature along with the range of values these features can take and whether or not these features are used to create labeling functions for content and/or title section of fake news articles. Features named *stop-*

Table 3.1: Stylistic Features

Feature Name	Range	Content	Title
Word count	$i \epsilon \mathbb{Z} \mid 0 \leq i \leq \infty$	✓	✓
Unique words count	$i \epsilon \mathbb{Z} \mid 0 \leq i \leq \infty$	✓	
Words per sentence	$i \epsilon \mathbb{R} \mid 0 \leq i \leq \infty$	✓	
Stopwords ratio	$i \epsilon \mathbb{R} \mid 0 \leq i \leq 1$	✓	✓
Unique words ratio	$i \epsilon \mathbb{R} \mid 0 \leq i \leq 1$	✓	✓
Average sentence length	$i \epsilon \mathbb{R} \mid 0 \leq i \leq \infty$	✓	
Average word length	$i \epsilon \mathbb{R} \mid 0 \leq i \leq \infty$	✓	

words_ratio and *unique_words* ratio are calculated by dividing their respective values by the number of total words in the text.

POS-tagging features: These are the features based on part-of-speech tags of words in a sentence. We use Spacy library to generate part-of-speech tags. We provide brief summary of how this process works in Appendix A. All features are applied to both title and content where separate labeling functions for each are generated. Each of the features are considered as a "ratio", which is calculated by dividing the number of occurrences of a given POS-TAG combination to the total number of words. Therefore, the value range each POS-tagging feature can take is $\{i \epsilon \mathbb{R} \mid 0 \leq i \leq 1\}$. Table 3.2 shows utilized POS-tagging features along with the POS and TAG identifiers as given in Spacy library and whether or not these features are used to create labeling functions for content and/or title portion of fake news articles. Descriptions for all POS and TAG identifiers mentioned in the Table 3.2 can be seen in Table A.1, provided under Appendix A. If the TAG identifier is not given, all tags under the given POS are included while calculating the feature value.

Punctuation features These are the features exploring various punctuation symbol usages in news articles. Given the smaller length of the news article titles, only the total number of punctuation symbols is utilized as a labeling function. However, more granular labeling functions are introduced for the content portion of the news articles, such as ratios of individual punctuation symbols like period, question mark, etc. Note that each of the following features are considered as a "ratio" where the num-

Table 3.2: POS-Tagging Features

Feature Name	POS	TAG	Content	Title
Noun ratio	NOUN		✓	✓
Proper noun ratio	PROPN		✓	✓
Cardinal number ratio	NUM	CD	✓	✓
Determiner ratio	DET		✓	✓
Adposition ratio	ADP		✓	✓
Interjection ratio	INTJ		✓	✓
Symbol ratio	SYM		✓	✓
Adjective ratio	ADJ		✓	✓
Wh-determiner ratio	PRON	WDT	✓	✓
Verb ratio	VERB		✓	✓
Present participle verb ratio	VERB	VBG	✓	✓
Past participle verb ratio	VERB	VBN	✓	✓
Third person verb ratio	VERB	VBZ	✓	✓
Modal ratio	AUX	MD	✓	✓
Adverb ratio	ADV		✓	✓
Comparative adverb ratio	ADV	RBR	✓	✓
Superlative adverb ratio	ADV	RBS	✓	✓
Existential ratio	PRON	EX	✓	✓
Pronoun ratio	PRON		✓	✓
Personal pronoun ratio	PRON	WP	✓	✓
Possessive pronoun ratio	PRON	PRP$	✓	✓

Table 3.3: Punctuation Features

Feature Name	POS	Text	Content	Title
Punctuation ratio	PUNCT		✓	✓
Period ratio	PUNCT	.	✓	
Question mark ratio	PUNCT	?	✓	
Exclamation point ratio	PUNCT	!	✓	
Comma ratio	PUNCT	,	✓	
Semicolon ratio	PUNCT	;	✓	
Colon ratio	PUNCT	:	✓	
Parentheses opener ratio	PUNCT	(	✓	
Parentheses closer ratio	PUNCT	)	✓	
Quotation mark ratio	PUNCT	“	✓	

ber of given punctuation symbol(s) is divided to the total number of words, similar to POS-tagging features. Therefore the value range punctuation features can take is $\{i \, \epsilon \, \mathbb{R} \mid 0 \leq i \leq 1\}$. Table 3.3 lists the employed punctuation features along with the POS identifiers as given in the Spacy library, individual punctuation symbols and whether or not these features are used to create labeling functions for content and/or title portion of fake news articles.

Readability features The final content feature category is related to readability analysis of the texts. Readability metrics are used to estimate the education level required to understand the text. The range of values changes depending on the metric used. We have selected three of the popular readability metrics which proved successful in text classification and fake news detection contexts. Readability metrics give more accurate results if a certain number of words is available, therefore the title portion of the fake news articles are not used while creating labeling functions with readability features. All these three features are listed in Table 3.4 along with the value range the features can take.

At the beginning of the weak labeling module, all introduced features are calculated and appended to both labeled and unlabeled input datasets. At this point, raw values with given ranges are directly appended without applying any normalization. A total

Table 3.4: Readability Features

Feature Name	Range	Content	Title
Gunning Fog Index	$i \epsilon \mathbb{Z} \mid 1 \leq i \leq 20$	✓	
Automated Readability Index	$i \epsilon \mathbb{Z} \mid 1 \leq i \leq 14$	✓	
Flesch Kincaid Index	$i \epsilon \mathbb{Z} \mid 1 \leq i \leq 18$	✓	

of 67 features are appended to both labeled and unlabeled datasets that are input to the weak labeling module. 41 of these features are for the content portion of the news articles and the remaining 26 of them are for the title portion of the news articles.

The objective is to generate labeling functions based on the listed features. One approach is to manually analyze the feature distributions of real and fake news articles. Custom labeling functions can then be created, where the function assigns fake, real or abstain values to news articles based on specific feature thresholds that are manually determined. These thresholds can be upper limits, lower limits or both, to define a certain range depending on the manual approach taken. However, this approach is time consuming as it requires manual work. In addition, it requires labels to be available for the dataset at hand. In our weak labeling module, we utilize a method to learn thresholds automatically so that generated labeling functions can then be applied to the dataset without labels. This approach is inspired by the work by Ozgobek et al. [28] [29]. We have made modifications to their algorithm and added the capability to perform feature selection.

In order to determine thresholds, descriptive statistics for features are used, which provides the distribution of the labeled dataset for a particular feature. To understand the differences in distribution, the labeled dataset is first divided into two subsets based on their labels. Then, the **describe** method provided in the pandas package, is used to generate the distribution of the real and fake datasets for the selected feature. The percentiles are selected as changing in every 0.05 points, dividing the (0, 1) range to 20 percentiles. Figure 3.2 illustrates the percentiles for the feature named *content_words_per_sentence* for NELA-GT elections dataset. This feature shows the average number of words per sentence in the content part of the news articles. Figure 3.2, visualizes the percentiles for both the real and fake subsets of the dataset,

highlighting the differences between fake and real values across various percentiles.

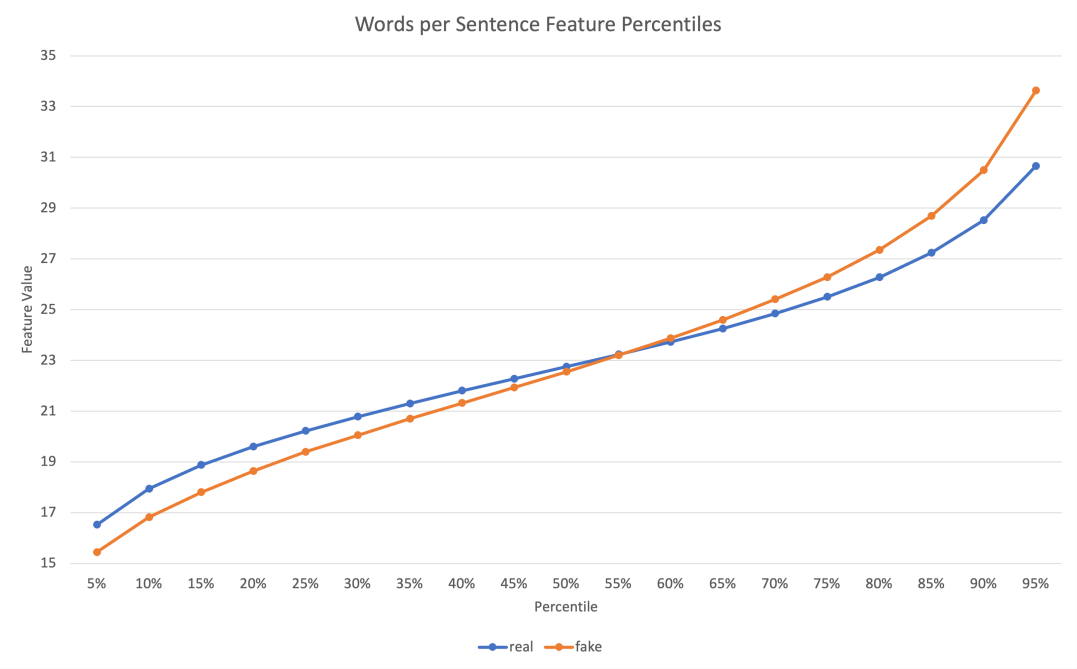

Figure 3.2: content_words_per_sentence percentile value differences between fake and real subsets of NELA-GT elections dataset

For every feature, there are infinitely many thresholds that can be found depending on the strategy. In order to automate the selection of thresholds, the problem is converted into a smaller one where we look for two thresholds, one for percentiles below .5 (.5 to .05) and one for above (.5 to .95). For the upper percentiles, the value of the feature in the selected percentile is then used as threshold and the data instances with higher values than the threshold are assigned either fake or real based on the side the algorithm chooses. Lower percentiles work in a mirrored way where data instances with lower values than the selected threshold are assigned either fake or real depending on the side the algorithm chooses. The pseudocode for upper percentile threshold search algorithm for a feature is given in algorithm 1. This algorithm is run for all features to either find an upper percentile threshold or to eliminate the feature and percentile list pair.

A similar algorithm is also applied for lower thresholds. In the lower thresholds case, percentiles are swept from 0.5 to 0.05 until a certain threshold is found or all percentiles in the range are exhausted. There are a couple of important points to

Algorithm 1 Threshold Search for Upper Percentiles

Require: fake dataset, real dataset, Feature

Ensure: threshold, side

$max\_fake \leftarrow$ maximum value of feature on fake dataset
$min\_fake \leftarrow$ minimum value of feature on fake dataset
$max\_real \leftarrow$ maximum value of feature on real dataset
$min\_real \leftarrow$ minimum value of feature on 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**
        $threshold \leftarrow p$
        if $percentile\_diff > 0$ **then**
            $side \leftarrow$ real
        else
            $side \leftarrow$ fake
        end if
        break
    end if
end for
return threshold, side

highlight about this algorithm.

- When a threshold is found, remaining percentiles are skipped. Finding threshold earlier if possible is better, because it can then cover a higher number of data instances. This is because the algorithm starts from percentile 0.5 and sweeps towards both ends.
- There is also a chance that the algorithm cannot find a threshold for a certain feature and percentile list. This means the values of the real and fake datasets in this range do not differ much. This makes the algorithm not yield any thresholds for the feature. We eliminate the feature/percentile list pairs where threshold is not found by the algorithm. This enables users of the proposed framework to introduce as many content features as possible to the system without worrying about diminishing the overall aggregated weak label accuracy.
- There is a constant C in the algorithm, which can be tuned to alter the threshold selection algorithm. There is an important tradeoff to consider for this constant. Increasing it makes the algorithm find the threshold in percentiles closer to the middle, which covers more data instances. However, given the difference value is smaller, the threshold becomes less selective. This makes the labeling function created from the selected threshold to mislabel more instances. The constant can be selected based on the desired tradeoff between coverage/accuracy for distinct applications.
- The algorithm not only finds a value but it also selects a side. This is done by looking at the sign of the difference between fake and real percentile. If it is positive, this means the values of the real dataset are more likely to be higher than the values of the fake dataset after this certain percentile, therefore real is selected as the side. In the opposite case, fake is selected as the side.

At the end of this process, labeling functions can be created out of features where threshold is found. In the best case, 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. The determined thresholds are saved into a dedicated JSON file where both upper and lower thresholds for each feature are listed, if they exist. An example of saved thresholds for the feature named *personal_pronoun_ratio* for the title part of the news articles is given

in Figure 3.3. In this example, two different labeling functions are created. In the first one, data instances with *title_personal_pronoun_ratio* value greater than the value listed in upper threshold are assigned as real. In the second one, data instances with *title_personal_pronoun_ratio* value smaller than the value listed in lower threshold are assigned as fake.

Figure 3.3: Example saved thresholds for a content feature

3.3.1.2 Model-Based Labeling Functions

The main goal of weak labeling systems is to combine as many dirty weak label sources as possible to benefit from each and every one of them. Even though we generate plenty of content based labeling functions, having other forms of weak labels can be beneficial to increase the accuracy of combined weak labels. Oftentimes the studies in the weakly labeled fake news classification area are only focused on one strategy to generate weak labeling sources. [21, 25, 28] Some of these strategies can be listed as only using content-based labeling sources or only using social context around the news articles. In our framework, we aim to combine the content-based labeling function generation strategy with model-based labeling functions where the labeled dataset given as input to the system is used to train supervised machine learning models to be used as weak labeling sources for the dataset without labels.

Certain qualities are considered when selecting models to be used in this part. Two important ones can be counted as being able to work well when trained with small amounts of labeled data and not requiring extensive resources to train. One of the setups we work with to solve the timely detection of fake news problem is semi-supervision. In this setup, we aim to label a large-scale dataset using the information automatically learned from limited labeled data. To make sure the model-based labeling functions work well with this setup, we aim to select models that do not significantly lose prediction performance when the number of data instances gets smaller. Complex models have a greater capacity and can closely fit the training data, which increases the risk of overfitting. [40] Therefore, we opt to incorporate some of the more interpretable and less complex approaches as part of the model-based labeling functions. Logistic regression classifier, multinomial naive bayes and random forest classifier are chosen for the weak labeling module for this reason.

Freund and Schapire [41] introduced boosting algorithms in 1996, along with a new algorithm called Adaboost, which stands for Adaptive Boosting. The basic idea behind boosting is to start with weak learners and then train a series of stronger learners that focus on the examples that the weak learner misclassified. This process is repeated until the desired level of accuracy is reached. In their study, Freund and Schapire argue that the boosting algorithms are more robust to overfitting, which makes them particularly suitable for small datasets. We opt to include Adaboost as part of our model-based labeling functions due to this feature of the algorithm. In another study, Chen and Guestrin [42] introduce XGBoost, which is a scalable tree boosting algorithm. XGBoost improves over Adaboost by incorporating additional features such as flexible loss functions, regularization techniques and advanced optimization methods. Chen and Guestrin point out that XGBoost's regularization methods further improve the algorithms ability to prevent overfitting, making it a good choice for small datasets. We also incorporate XGBoost in our work as part of model-based labeling functions. Our final set of supervised algorithms utilized in the weak labeling module of WeSTeC can be listed as follows;

- Logistic Regression (LR)
- Naive Bayes (NB)
- Random Forest (RF)

- XGBoost
- Adaboost

Similar to content-based labeling functions, model-based labeling functions are also created for both content and title portions of news articles separately. After training the models with the labeled dataset provided as input to the weak labeling module, they are saved to be used in the labeling function application step. Overall, we have 10 models saved and ready to be used as model-based labeling functions. The labeling functions created for these models predict the label using the trained and saved models and return the value directly as labeling function output. Different from content-based labeling functions, these do not return the value abstain, since the trained models always assign a real or fake value. This also means the coverage of these labeling functions are always 100% where all data points are assigned either real or fake values as a result. This is different in content-based labeling functions where each labeling function assigns only one of real or fake values depending on the threshold, or they abstain, resulting in various coverages depending on the content feature. The decision on which labeling function is programmed to assign either fake or real is determined by the "side" output of the threshold search algorithm described in section 3.3.1.1.

3.3.2 Application of Labeling Functions

At this stage in the pipeline, labeling functions generated through two different strategies are ready to be applied to the dataset without labels. There are always 10 model-based labeling functions. In addition, there are potentially 134 content-based labeling functions. Depending on the constant selected in the threshold selection algorithm for content-based labeling functions, this number can diminish significantly.

The Snorkel library[4] [1] is used to represent the labeling functions in a unified way and apply them to the dataset without labels. PandasLFApplier module is used to apply labeling functions to all instances. Assuming *m* labeling functions are generated in total and there are *n* number of news articles in the dataset without labels, at the end of the labeling function application, a matrix with dimensions *m* x *n* is generated. In this matrix, each row identifies a data instance and all columns identify the outcome

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

of labeling functions where the cells can take values of -1, 0 or 1. These values represent abstain, real and fake respectively.

3.3.2.1 Weak Label Aggregation

One of the most challenging aspects of weak labeling is to aggregate applied weak labels into a single probabilistic label, without any labeled data. There are various studies exploring how to combine weak labels without having access to ground truth labels. WeSTeC supports three of the commonly used weak signal aggregation strategies. We explain these approaches further 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. If the counts are equal, the majority vote strategy fails to assign a single aggregated label and abstains. Otherwise it assigns the label type that has the highest vote. This strategy does not end up with probabilistic labels, it directly assigns values of -1, 0 or 1 as a single aggregated label.

Snorkel Label Model Ratner et al. [1] introduce a label model that tries to combine noisy weak labels into a final set of training labels by assigning probabilistic weights to the output of each labeling function based on their estimated accuracies and correlations. Then, the labels from all labeling functions are aggregated probabilistically, taking into account the weights assigned by the label model without having access to ground truth labels. This model is available as part of the Snorkel library, as the LabelModel aggregator.[5] Label model uses a conditionally independent model where all labeling functions are assumed to be independent.

Hyper Label Model Final strategy WeSTeC supports is hyper label model. Wu et al. [39] introduced graph neural networks (GNN) based label model, which infers the aggregated labels in a single forward pass. They show that the hyper label model overperforms existing aggregation strategies over 14 benchmark datasets both in terms of accuracy and efficiency. Different from the label model proposed by Ratner et al. [1]

[5] https://snorkel.readthedocs.io/en/v0.9.3/packages/labeling.html

the hyper label model also considers conditional dependencies between labeling functions to alter the weights. Hyper label model directly outputs discrete labels assigned by the model, as opposed to probabilistic labels like the Snorkel Label Model.

Out of three aggregation strategies, only the Snorkel Label Model outputs probabilistic labels. WeSTeC has support for data elimination depending on the probabilistic labels. If the the user specifies the minimum number of data instances required, the framework selects data instances where the Snorkel label model exhibits the highest confidence without disrupting the balance of the dataset. We discuss how this improves the performance of weak labeling aggregation and help the text classification task in section 4.4.2. Whether or not data elimination is conducted, the probabilistic labels are converted to discrete labels by rounding up to the nearest discrete label at the end of labeling function aggregation step. As the output of the weak label aggregation step, the combined label obtained through each aggregation strategy is saved to a dedicated dataframe, along with the actual news articles.

3.4 Text Classification

The main purpose of our work is to be able to classify fake news articles in early stages of dissemination. In the weak labeling module, weak labels are generated and combined to obtain a single label for each data instance. As the next step, we run a supervised text classification model using the aggregated weak labels. Then, we evaluate the results of our models with the actual labels. Since there are multiple weak label aggregation strategies in our work, only the highest performing strategy is selected to test in the text classification module.

3.4.1 RoBERTa Text Classification

Various studies show the success of large language models in the text classification setting compared to previous approaches. [37, 38] These also include studies in fake news classification. [35, 28] The RoBERTa text classifier is selected as the single end-model in our framework given its superiority demonstrated by many studies.

In the text classification module of the proposed framework, the first step involves preparing the text contents for RoBERTa text classification. The classifier expects a single text column. In our study, we conduct experiments using a dataset of news articles where the title and content portions are stored in separate columns. We combine the title and content portions of the news articles by adding a dot and a space character in between.

Simple transformers library[6] is used to accelerate our work since it provides an easier interface to interact with models based on transformers architecture. It has full support for RoBERTa text classification as well. The pipeline is initialized by turning text into tokens as expected by the RoBERTa text classifier. This is done through the RoBERTa tokenizer provided by HuggingFace library[7]. Then, the pretrained RoBERTa base model is downloaded. By using the highest performing aggregated weak labels, the model is finetuned for our task, which is fake news classification of an emerging topic in our case. In addition, Weights and Biases[8] (WandB) library is used to be able to easily track and visualize the training process. We also run hyperparameter tuning for learning rate through the WandB library. At the end of the text classification module, a ready-to-use fine tuned RoBERTa text classifier specialized on the domain of unlabeled dataset is generated as the output.

WeSTeC provides additional capabilities to evaluate the performance of text classifiers. If actual labels are provided, the model fine-tuned on weak labels is tested against the actual labels and the results are reported. In order to validate our weak labels, we follow the steps of fine-tuning the text classifier when actual labels are available in the training phase. The pretrained RoBERTa base model is fine-tuned using the actual labels and then tested using the actual labels as well. To make the comparison meaningful, all other variables in the pipeline, including hyperparameters, are kept the same. This approach allows us to show the usefulness of the aggregated weak labels generated by the weak labeling module of the proposed framework. We provide a detailed analysis of the text classification results in the early fake news detection task in section 4.4.2.

[6] https://simpletransformers.ai/
[7] https://huggingface.co/docs/transformers/model_doc/roberta
[8] https://wandb.ai/site

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, we present the experiments conducted on the proposed framework and the achieved results. Firstly, we discuss the process of dataset selection and provide various statistics on the selected dataset. Next, we discuss the experiments conducted on the semi-supervision setting. These experiments include the results obtained for model-based labeling functions, the overall aggregation results of labeling functions and the performance of text classification models trained on weak labels compared to models trained on actual labels.

We also present the results for the setting where labeled data from another domain is available, showing how the proposed framework performs in domain transfer setting. Additionally, we compare our experiments with the state-of-the-art studies in weakly supervised fake news detection. We first compare our weak labeling module with studies that generate and aggregate weak labels to programmatically label large-scale datasets. Then, we evaluate the performance of our text classification experiments in early detection of fake news tasks against other studies in weakly-supervised timely fake news identification.

4.1 Dataset

There are some important qualities that our study expects from the dataset. These can be briefly listed as follows.

- The dataset should consist of news articles and labels depicting if each article is fake or credible for evaluation purposes.

- The dataset should include enough number of data instances to test the idea of programmatically labeling a large-scale dataset and utilizing supervised end models on top of it.
- The dataset should enable us to test the idea of weakly labeling news articles of emerging topic. One of the setups we consider is domain adaptation where labeled datasets containing news articles on older topics are used to programmatically label large-scale unlabeled dataset of news articles on an emerging topic. Therefore, it should contain news articles of multiple topics.

There are several publicly available datasets for research on fake news detection. D'Ulizia et.al. [43] and Hu et. al. [44] provide a detailed comparison of these datasets. Considering our requirements regarding the number of data instances, applicability for domain transfer, and availability of ground truth labels, we conduct a thorough search to identify the most suitable dataset. In order to quantitatively evaluate our proposed approach, datasets with ground truth labels are required.

There are several methods to assign ground truth labels to data instances, including manual labeling, source-based labeling, etc. Manual labeling is considered the most accurate method for obtaining ground truth labels because of its case-by-case analysis nature. However, manual labeling is time-consuming, resulting in publicly available datasets with small sizes. The focus of our work is programmatically labeling large-scale datasets through small semi-supervision or domain transfer, therefore, we require larger volumes of data instances. In addition, we require the dataset to have data instances from multiple topics, allowing us to evaluate the domain transfer approach in our work.

Considering all these constraints, we have selected the NELA-GT news article dataset [45] to use in our work, specifically 2021 and 2022 editions. NELA-GT datasets are a series of regularly published news article datasets from over 500 news outlets. The first edition of the dataset was published in 2017. Since then, authors have continued to publish the dataset regularly, improving different aspects of the dataset each year. The labeling of this dataset is based on source reliability. The authors used 8 different data sources that assess the reliability and bias of the news outlets and combined these to achieve reliability assessment for each news outlet in the dataset.

These assessment sources include both research community and practitioner community organizations. Each of these organizations use a different criteria and methods to make their assessments. In order to create a large, centralized set of veracity labels, the authors combined all these sources and assigned one of three reliability labels (reliable, unreliable and mixed) to each news outlet.

The authors of the NELA-GT dataset also published subsets of the original dataset, each containing news articles regarding only certain topics. These topics are US elections and Covid for the 2020 edition of the dataset while US Capitol attack and Covid for the 2021 edition. We select Covid and US Elections as two distinct topics to use in our work. To make sure we benefit from all available data, we combine Covid datasets from both 2020 and 2021 editions. When it comes to the columns included in the NELA-GT datasets, we only benefit from four fields: *id*, *source*, *title* and *content*. The *source* field is only used to obtain labels for data instances by merging the source reliability dataset with the original news article dataset. The source reliability dataset consists of two fields: source and label, where the label column can take values reliable, unreliable and mixed. The merge operation is done using the shared *source* field in both datasets. Then, the label column from the source reliability dataset is kept, while the original source column is removed. As a result, the source reliability values turn into news article labels: *real*, *fake* and *mixed*, corresponding to *reliable*, *unreliable* and *mixed*, respectively.

Before the actual pipeline starts, we apply several base preprocessing and data elimination steps to both elections and covid datasets. These can be listed as follows.

- **Eliminate data instances with the label field *mixed*.** Since we use source-labels as the ground truth labels, we eliminate the news articles from sources where the source reliability is marked as mixed.
- **Eliminate data instances where either title or content is empty.**
- **Eliminate @ tokens placed for copyright purposes.** The authors suggest that articles collected from news outlets may be subject to copyright protection. Therefore, they applied a transformation to the original text, making it unsuitable for their originally intended purpose, namely news consumption. This way, news articles can still be used for text analysis. The authors replace 7

Table 4.1: NELA-GT Dataset Statistics

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

tokens with "@" every 100 tokens for articles longer than 200 tokens. For articles below 200 tokens, they replace every 5 tokens with "@" every 20 tokens. We remove these consecutive "@" tokens completely. Even though this preprocessing affects the original text, approximately 93% of the text content is still there and it does not deeply affect the subsequent analysis.

- **Eliminate data instances where the content field has less than 200 tokens.** The "@" transformation rate is higher for data instances with fewer than 200 tokens, resulting in approximately 25% of the original content being lost. To make sure that our work is not affected by this loss, we remove the rows where the content field contains a number of words less than this threshold.

Table 4.1 shows statistics for both the elections and covid datasets. In both covid and election datasets, the number of real news articles are substantially higher than the number of fake news articles. We undersample both datasets in order to obtain balanced datasets. This is possible due to the high number of news articles in both datasets available.

Certain text statistics for content and title portion of news articles in both covid and election datasets are provided in Table 4.2 and Table 4.3. Statistics are given separately for fake and real news articles and higher values between fake and real subsets are highlighted. By looking at the comparisons, we can see that fake news have longer titles in both datasets compared to real news. Conversely, the average number of words in news contents are more lengthy in the real datasets compared to fake ones. This may highlight the clickbait nature of fake news titles but when the actual content is read, it is more shallow compared to real news contents. Also, we can see that the average number of words per sentence is higher in fake news for both content

Table 4.2: NELA-GT Elections Dataset Text Statistics

Elections Dataset	Content		Title	
	Fake	Real	Fake	Real
Avg. number of words	794.513	**997.601**	**14.132**	11.507
Avg. word length	4.819	**4.824**	**5.251**	5.205
Avg. number of sentences	36.541	**45.109**	**1.226**	1.123
Avg. number of words per sentence	**23.316**	23.118	**12.452**	10.699
Avg. number of stop words per sentence	**10.853**	10.664	**3.857**	3.332
Avg. number of symbols per sentence	2.707	**2.759**	**0.983**	0.839

Table 4.3: NELA-GT Covid Dataset Text Statistics

Covid Dataset	Content		Title	
	Fake	Real	Fake	Real
Avg. number of words	769.058	**803.085**	**14.165**	11.719
Avg. word length	**4.843**	4.787	**5.285**	5.202
Avg. number of sentences	35.041	**36.846**	**1.209**	1.117
Avg. number of words per sentence	**23.381**	22.996	**12.562**	10.936
Avg. number of stop words per sentence	**10.910**	10.708	**3.958**	3.389
Avg. number of symbols per sentence	2.734	**2.789**	**0.978**	0.856

and title portions in both datasets. Fake news articles are often written to be more sensational and attention-grabbing. This might lead to use of longer sentences in fake news. In content based labeling functions, we utilize these features along with many others to distinguish fake and real news.

4.2 Evaluation Metrics

We evaluate the pipeline at the end of multiple stages as listed below.

- Training score evaluation: We evaluate the performance of models trained for model-based labeling functions.

- Application Performance Evaluation: We measure the effectiveness of applying models trained for model-based labeling functions to the unlabeled large-scale dataset.
- Evaluation after Weak Label Aggregation: We evaluate the quality of the generated aggregated labels by comparing them against actual labels.
- Fake News Detection Classifier Evaluation: We assess the effectiveness of the trained fake news detection classifier on a test subset.

For each of these evaluations, the same metrics are utilized. These metrics can be enumerated as follows.

- **Accuracy** is a measure of how often a classification model correctly predicts the class of an instance. It is calculated by dividing the number of correctly predicted instances by the total number of instances. A high accuracy indicates that the model is good at classifying instances correctly.
- **Precision** is a measure of how often a classification model correctly predicts a certain class. It is calculated by dividing the number of correctly predicted instances of a class by the total number of instances predicted as that class.
- **Recall** is a measure of how often a classification model correctly predicts a class, given that the instance is actually of that class. It is calculated by dividing the number of correctly predicted instances of a class by the total number of actual instances of that class.
- **F1-score** is a measure that combines precision and recall into a single value. It is calculated by taking the harmonic mean of precision and recall. A high F1-score indicates that the model is good at both avoiding false positives and false negatives.

We perform all our experiments on the balanced dataset, which allows us to employ commonly used text classification metrics in our study without the need to look for specialized metrics. This is also beneficial because studies in the text classification field commonly use these metrics. Using the same measures makes it is easier to compare and evaluate our results against existing studies in the field. In our study, we focus on a binary classification problem. To calculate single precision, recall or F1-score metrics for both classes, we take the average of their respective measures.

For instance, to obtain a combined precision score, we take the average of precision measures for data instances identified as both real and fake. Since our dataset is balanced, we do not need to take a weighted average.

4.3 Semi-Supervision Setup

The first setup we explore is the case where we have a small amount of labeled data that we use to programmatically label a large scale dataset. To be able to use and experiment our weak labeling module with this setup, we extract 2000 data instances from the Covid dataset and create a separate dataset. The original dataset is left with 295,518 data instances, resulting in a labeled to unlabeled ratio of 0.67%. The instances for the subset dataset are randomly selected in a balanced fashion, resulting in 1000 data instances for each label type. Then, both of the resulting datasets are fed into the weak labeling module of WeSTeC where the dataset with 2000 instances have their actual labels attached to them and the other dataset does not.

We present our results for this setup in three subsections. Firstly, we discuss the training and testing results of models used as part of model-based labeling functions. Then, we share outcomes obtained when applying all labeling functions to the data instances of unlabeled dataset and aggregating the generated weak labels. Finally, we share the outcomes of our RoBERTa text classifier trained with aggregated weak labels and tested against actual labels to validate two hypotheses: to determine if our fake news classifier can generalize beyond the performance of the aggregated weak labels, and to validate if our models can achieve comparable performances to the exact same model trained with actual labels.

4.3.1 Weak Labeling for Semi-supervision

4.3.1.1 Model-Based Labeling Functions

We train a number of models using the dataset with ground truth labels. These models are then used as part of labeling functions, where each labeling function utilizes a single trained model. During the application of labeling functions, the label for each data

instance in the unlabeled dataset is predicted using the corresponding trained model. In the following subsections, we first present the setup for model training. Then, we present our results of the training phase for the models, excluding the involvement of the unlabeled dataset. Finally, we provide the results of model application, where the trained models are applied to and tested against the unlabeled dataset.

Setup We apply a number of preprocessing steps to make both title and content parts of the news articles ready for training. These steps are shared by all 5 different model trainings. We use the gensim library [1] to create and execute the preprocessing pipeline. Pipeline involves the following steps.

- Strip any HTML tags that exist in text.
- Strip punctuation symbols from the text
- Convert multiple whitespaces into a single one
- Remove any numeric symbols
- Remove all stopwords
- Remove words with less than 3 letters
- Stem the text

After the preprocessing steps, the text is fed into the TF-IDF vectorizer for tokenization. TF-IDF stands for term frequency-inverse document frequency, which is a statistical measure that quantifies the importance of a word in a document within a collection of documents. [46] It is calculated by multiplying the term frequency and inverse document frequency metrics. Term frequency represents how often a particular term appears within document. Inverse document frequency, on the other hand, assesses the uniqueness of a term across a collection of documents. We make use of TF-IDF vectorizer in scikit-learn library,[2] which tokenizes textual documents in a corpus using the TF-IDF approach. Initially, TF-IDF vectorizer requires all documents in the corpus in order to generate the inverse document frequency. Then, any document in the dataset can be converted into tokens using the TF-IDF vectorizer.

In our case, TF-IDF vectorizer is initially constructed by feeding all text in the Covid and Elections datasets to make sure it can tokenize any subset of these datasets includ-

[1] https://radimrehurek.com/gensim/
[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

ing both content and title parts. This is done only once and the TFIDF vectorizer is saved to the file system. Whenever we require tokens for the title or content portion of any news articles in each dataset, we simply pass it into this ready-to-use vectorizer.

The tokens obtained from the vectorizer are directly used as features for the trained models. We have 5 different models with different hyperparameters. For most of the classifiers, we employ grid search to identify the best combination of hyperparameters. However, for the logistic regression classifier and multinomial naive Bayes classifier, no hyperparameter search is conducted. Throughout the grid search for hyperparameter tuning, we use 80/20 training test split. We provide the final set of hyperparameters used in our work for each distinct model. Even though separate models are trained for title and content parts of the news articles, the same hyperparameters are used for both.

- **XGBoost**
  - **Learning rate**: 0.15
  - **Max depth**: 9
  - **Number of estimators**: 1000
- **Adaboost**: Boosted ensemble is built using a base estimator. In our work, the random forest estimator is selected as the base estimator.
  - **Base estimator max depth**: 10
  - **Base estimator minimum number of samples required to be at a leaf node**: 1
  - **Learning rate**: 1.0
  - **Number of estimators**: 50
- **Random Forest**
  - **Number of estimators**: 100
  - **Max depth**: 12
  - **Minimum number of samples required to be at a leaf node**: 1
  - **Minimum number of samples required to split an internal node**: 5
  - **Number of features to consider when looking for the best split**: Square root of number of total features

Table 4.4 shows the training results. An 80/20 train test split is used, resulting in

Table 4.4: Semi-Supervision model training results

		XGBoost	Adaboost	LR	RF	NB
Content	Accuracy	0.73	0.69	0.76	0.74	**0.78**
	F1 Score	0.73	0.69	0.76	0.74	**0.78**
Title	Accuracy	0.61	0.58	0.65	0.66	**0.68**
	F1 Score	0.61	0.58	0.65	0.66	**0.68**

1600 training and 400 testing data instances. The accuracy and F1 scores for each model training, obtained from both content and title portions of news articles, are given separately.

Due to the limited number of data instances, the results fall below the 0.8 threshold. Although none of these models can be solely used for programmatically labeling data instances in a large-scale dataset, we use them as one of the weak labeling sources and we aim to generalize beyond any weak labeling source used.

Another observation from the training scores show that the simpler statistics-based models provide better results compared to boosting algorithms, which can also take advantage of a larger number of training instances. This is expected due to the low number of training instances.

Table 4.5 shows results of the accuracy and F1 scores obtained when the trained models are tested against the instances of large-scale Covid dataset without labels, having 295,518 data instances in total. As expected, the results are similar to the testing results provided in Table 4.4 since both labeled and unlabeled datasets contain news articles from the same domain. The number of testing instances in Table 4.4 is 400 whereas the number of instances in the unlabeled dataset is 295,518, which makes the testing results more accurate.

With the trained models ready and saved, model-based labeling functions are ready for labeling function application and weak label aggregation steps. Similarly, there are also a couple of requirements when it comes to content-based labeling functions. Both content and model based labeling functions need to be ready for the weak labeling module to move onto the next step. Firstly, content features are calculated

Table 4.5: Semi-Supervision model apply results

		XGBoost	Adaboost	LR	RF	NB
Content	Accuracy	0.74	0.68	**0.77**	0.74	0.75
	F1 Score	0.74	0.68	**0.77**	0.74	0.75
Title	Accuracy	0.63	0.61	**0.66**	0.64	0.65
	F1 Score	0.62	0.61	**0.66**	0.64	0.65

and appended to both labeled and unlabeled datasets. Then, threshold selection algorithm is run for each content feature, using the ground truth labels of the labeled dataset. In the end, thresholds are saved to be used in the next steps. The features for which no thresholds are found, are eliminated. After the framework completes all required steps for the preparation of both content and model-based labeling functions, it proceeds with the labeling function application and aggregation steps.

4.3.1.2 Labeling Function Aggregation

By using the thresholds identified from the labeled dataset, the number of labeling functions can vary. There are cases where the threshold selection algorithm can fail to find a satisfactory threshold for a particular combination of feature and side. This results in the elimination of that combination from further consideration in the rest of the pipeline. In the case of using the labeled Covid subset dataset with 2000 instances, out of 134 possible labeling functions, only 35 are created when the constant C in threshold selection algorithm is set to 5. Selecting a bigger constant can allow the algorithm to find more labeling functions but we prioritize higher accuracies over a greater number of labeling functions in this case.

Tables 4.6 and 4.7 show best performing 10 and worst performing 5 labeling functions when applied to the unlabeled dataset. Looking at the worst performing labeling functions, it can be seen that even the worst performing functions achieve higher than random guessing, showing the effectiveness of the threshold search algorithm. All identified labeling functions, except one, show higher accuracy than random guessing when evaluated individually without combination with other labeling functions.

Table 4.6: Semi-Supervision best performing LFs

Labeling Function Name	Empirical Accuracy
lf_model_logistic_regression_content	0.768
lf_model_naive_bayes_content	0.753
lf_model_xgboost_content	0.744
lf_model_random_forest_content	0.738
lf_content_exclamation_point_ratio_upper_fake	0.736
lf_content_flesch_kincaid_index_upper_fake	0.728
lf_title_noun_ratio_upper_real	0.719
lf_content_automated_readability_index_upper_fake	0.717
lf_title_proper_noun_ratio_upper_fake	0.711
lf_title_word_count_upper_fake	0.708

Table 4.7: Semi-Supervision worst performing LFs

Labeling Function Name	Empirical Accuracy
lf_title_cardinal_number_ratio_upper_fake	0.490
lf_title_punctuation_ratio_lower_real	0.510
lf_content_existential_ratio_lower_fake	0.512
lf_content_existential_ratio_upper_real	0.535
lf_title_pronoun_ratio_upper_fake	0.542

Looking at the top 10 labeling functions, we can see that both model and content based labeling functions are included. This highlights the effectiveness of both of our approaches in generating accurate labeling functions.

We perform complementary analysis to gain a better insight on how lower scoring labeling functions affect the overall accuracy. We use the simplest labeling function aggregator for this experiment, which is the majority vote strategy, and vary the number of top labeling functions selected to see how the accuracy and coverage of the aggregated weak labels change. It is important to note that this analysis uses the actual ground truth labels of the unlabeled dataset to calculate labeling function ac-

curacies and select top K accordingly. This analysis is not part of the overall pipeline and only conducted to gain additional insight to weak label aggregation step. In the overall pipeline all identified labeling functions are used and no labeling function selection is conducted. Figure 4.1 illustrates this analysis, providing insights into the relationship between the number of top labeling functions selected and the resulting accuracy and coverage of the aggregated weak labels.

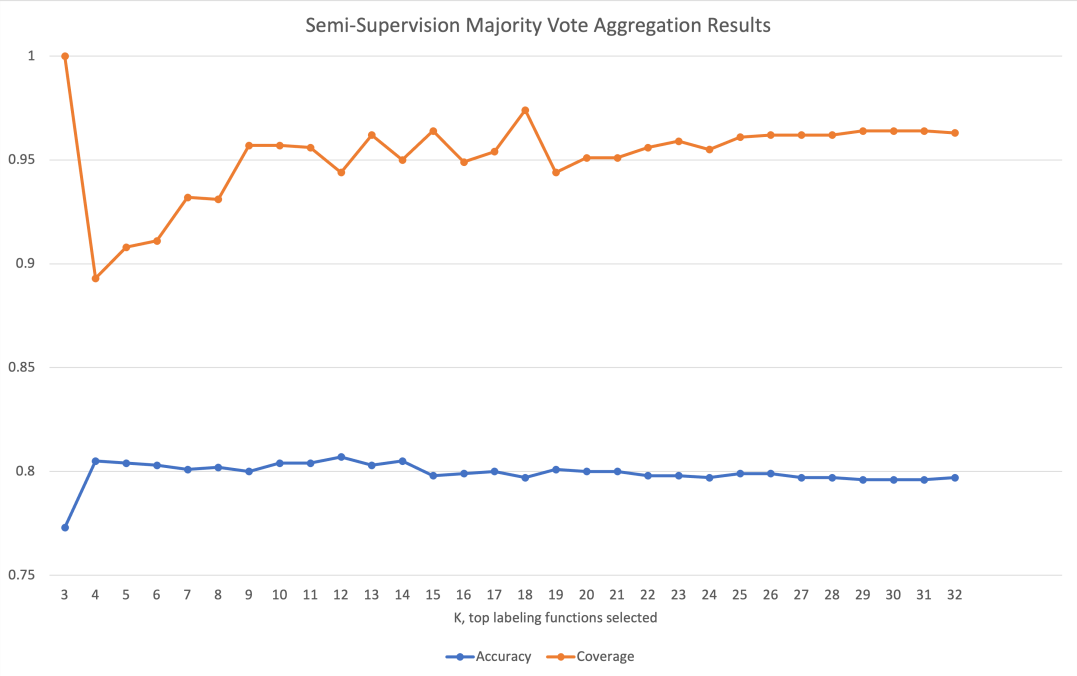

Figure 4.1: Majority vote aggregation results with changing top k labeling functions selected, semi-supervision

Looking at the Figure 4.1, the accuracy remains stable when K is below 14, but then it starts to gradually decrease. On the other hand, the coverage increases as the number of labeling functions increases, since the likelihood of having an equal number of votes on each side decreases. When the total number of labeling functions is an even number, it increases the chances of cases where the aggregator assigns abstain value, resulting in a zigzag-like line graph. We see that the labeling functions with lower accuracy affects the overall aggregated weak label accuracy as K increases, although not significantly.

WeSTeC supports data instance selection as part of the weak labeling module. Snorkel label model generates probabilistic labels without having access to ground truth la-

Table 4.8: Semi-Supervision Performance of Weak Label Aggregation Strategies

	Accuracy	Precision	Recall	F1 score	Coverage
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

bels. Using the probabilistic labels, the framework can find the data instances for which the aggregator shows higher confidence compared to others. If the application using WeSTeC does not need to utilize all data instances of the unlabeled dataset, target number of data instances can be specified to the framework. WeSTeC selects specified number of instances from the unlabeled dataset where the Snorkel label model shows greater confidence. The rest of the data instances are ignored in the rest of the pipeline.

WeSTeC supports three different weak label aggregation strategies. These are majority vote, Snorkel label model and Hyper label model. More information on how these strategies work can be found in 3.3.2.1. Majority vote and Hyper label model strategies directly output discrete labels, -1, 0 and 1 corresponding to abstain, real and fake respectively. On the other hand Snorkel label model strategy outputs probabilistic label in the range from 0 to 1, showing the possibility of data instance being fake. We perform an additional step to convert these probabilities to discrete labels.

On top of this, we also utilize the probabilistic labels generated by Snorkel label model to select 50,000 data instances where the aggregator is most confident. We present our results using the accuracy, precision, recall, F1-score and coverage metrics in Table 4.8. Both Snorkel label model results are given in the table, with or without the data selection process conducted.

Looking at the results, we can see that all strategies achieve higher accuracy compared to the performances of individual labeling functions shared in Table 4.6. The best performing labeling function achieves 0.768 accuracy. This shows the strength of weak supervision and the importance of being able to combine various weak labeling

sources.

When it comes to comparing different labeling function aggregation strategies, without data selection, all strategies output similar results, however, the majority vote strategy slightly outperforms the other approaches which are more complex in nature. This result is similar to the results obtained by Wu et. al. [39], who created the hyper-label model aggregation strategy. However, the majority vote strategy abstains when the number of votes are the same, resulting in slightly less coverage than 100%.

In addition, the data selection process applied to Snorkel label model strategy significantly improves accuracy, showing the effectiveness of probabilistic labels. We compare the aggregated weak label accuracies with the state-of-the-art weakly supervised fake news detection studies in section 4.5.1, together with the performance of domain adaptation setup. Weak labeling module terminates by assigning aggregated weak labels to the unlabeled dataset obtained by each aggregation strategy.

4.3.2 Semi-Supervised Text Classification

The aim of the weak labeling module is to be able to programmatically label large scale datasets. Subsequently, one can use the large-scale labeled dataset to train state-of-the-art supervised models that are known to work best in fake news detection tasks. We select RoBERTa text classification as our text classification model. [13] Authors of RoBERTa argue that it performs better than state-of-the-art models including its predecessor BERT on three different tasks. They achieve this by improving over the BERT pretraining by using bigger batches over more data, training the model longer, removing the next sentence predition objective, training on longer sequences and dynamically changing the masking pattern applied to the training data. When it comes to fake news detection, Ozgobek et al. [28] compare five different text classification models and show that RoBERTa gives the best results. Looking at all these results, we select RoBERTa as our text classification model.

When it comes to ground truth labels used in text classification model fine tuning, the aggregated weak labels obtained through the Snorkel label model aggregation strategy, combined with the data selection layer, are used. For the training process,

Table 4.9: Semi-Supervision Text Classification Results

Label Type	Tested Against	Accuracy	Precision	Recall	F1 score
Weakly supervised	Weak Labels	0.993	0.993	0.993	0.993
	Actual Labels	0.952	0.952	0.953	0.952
Supervised	Actual Labels	0.968	0.968	0.968	0.968

we select a dataset size of 50,000 instances, ensuring a balanced distribution. The training is run for 3 epochs with default model hyperparameters with the exception of the learning rate, which is tuned by running sweeps using WandB.

Table 4.9 shows the results for the trained RoBERTa text classifier where an 80/20 train-test split is used. Three different test results are provided in the table. The first two results are obtained using the weak labels as the ground truth labels during training. However, their testing procedures differ. The first row shows the results when the trained model is tested against weak labels. The second row reveals the results for the case when the trained model is tested against the actual labels. For comparison, the same training setup is run for the case where actual ground truth labels are used in both training and testing time. The corresponding results for this setup are displayed in the third row of the table.

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. The accuracy scores of the fake news classifier trained with weak labels and actual labels are within a range of 1-2%, proving the effectiveness of the aggreageted weak labels. In addition, the accuracy score of the aggregated labeling function is reported as 0.931 in Table 4.8. These labels are used to train the text classification mode, which achieves an accuracy of 0.952 when tested against the actual labels. This shows the fact that classification models trained on aggregated weak labels can generalize beyond the initial labels and achieve higher accuracies. We compare the text classification results with state-of-the-art weakly supervised fake news detection techniques in section 4.5.2, together with evaluating the performance of the domain adaptation setup.

4.4 Domain Adaptation Setup

The second setup we consider is domain adaptation. The semi-supervision setup shows the base case where no labeled data is available even from other domains. One can manually label a small amount of data and use that to programmatically label large-scale dataset. Although we achieve satisfactory results in the semi-supervision setup, it still requires some labeled news articles related to the emerging topic.

In real world scenarios, enterprises accumulate labeled data accross various domains, including older topics discussed over time. We explore the idea of using these labeled data sources from past events to label a large-scale dataset related to an emerging topic where no labeled data at the beginning of the dissemination. This makes the system to function without relying on any labeled data specifically from the target domain.

WeSTeC is used in a fashion similar to the semi-supervision setup, with the only variation being the labeled dataset provided to the framework. In this setup, we use the elections dataset as the labeled data, which contains 80,369 labeled data instances. This dataset is completely on the politics domain and includes news articles that mention the US elections in 2019. The dataset without labels, provided to the weak labeling module, is the same as the semi-supervision setup, which is the covid dataset with 297,518 data instances. This dataset contains news articles that discuss the Covid-19 pandemic.

4.4.1 Weak Labeling for Domain Adaptation

4.4.1.1 Model-Based Labeling Functions

The models and hyperparameters used in each model training remain the same as in the small amounts of labeled data setup. We present the training results of the models used in model-based labeling functions in Table 4.11. The training is done using the labeled elections dataset, with an 80/20 train-test split resulting in 64,295 training instances and 16,074 testing instances.

Table 4.10: Domain adaptation model training results

		XGBoost	Adaboost	LR	RF	NB
Content	Accuracy	**0.89**	0.74	0.86	0.74	0.78
	F1 Score	**0.89**	0.74	0.86	0.74	0.77
Title	Accuracy	**0.76**	0.72	0.75	0.70	0.74
	F1 Score	**0.76**	0.72	0.75	0.70	0.74

Table 4.11: Domain Adaptation model apply results

		XGBoost	Adaboost	LR	RF	NB
Content	Accuracy	**0.85**	0.70	0.82	0.68	0.73
	F1 Score	**0.85**	0.70	0.82	0.67	0.72
Title	Accuracy	**0.72**	0.67	0.71	0.65	0.70
	F1 Score	**0.71**	0.67	0.71	0.65	0.70

The results show that the more complex algorithms like XGBoost show significantly better results compared to the semi-supervision setup. However, simpler algorithms such as random forest did not benefit as much from the larger number of data instances.

Table 4.11 shows the results obtained when the trained models are tested against the Covid dataset. These results are more important than the internal training results on the elections dataset alone, since the objective of our work is to use the trained models in another domain.

Looking at the results, we observe a slight decrease in the scores when testing against the Covid dataset compared to testing against the elections dataset. However, this decrease is not substantial, and the accuracy scores when tested against unlabeled dataset are higher than those obtained in the semi-supervision setup. These findings show the effectiveness of the proposed system in the domain adaptation setup.

Table 4.12: Domain Adaptation top performing LFs

Labeling Function Name	Empirical Accuracy
lf_model_xgboost_content	0.849
lf_model_logistic_regression_content	0.820
lf_model_naive_bayes_content	0.729
lf_content_exclamation_point_ratio_upper_fake	0.723
lf_model_xgboost_title	0.715
lf_model_logistic_regression_title	0.712
lf_title_proper_noun_ratio_upper_fake	0.711
lf_model_naive_bayes_title	0.705
lf_model_adaboost_content	0.696
lf_content_parantheses_close_ratio_upper_fake	0.689

4.4.1.2 Labeling Function Aggregation

In the domain adaptation setup, 34 labeling functions are generated by the framework where 24 of them are content-based and the rest is model-based. Tables 4.12 and 4.13 show best performing 10 and worst performing 5 labeling functions. Similar to the semi-supervision setup, we can see that even the worst performing content-based labeling functions score better than random guessing, making all labeling functions identified by the threshold search algorithm valid except one. In addition, looking at the top performing labeling functions, the best scoring labeling functions are model-based, showing the effectiveness of the model-based approach in the settings where a larger number of training data instances are available. Although the training data is in another domain, model-based labeling functions in domain adaptation setup achieve higher accuracies compared to the content-based labeling functions.

Similar to the previous setup, we perform a complementary analysis to gain a better understanding of how lower scoring labeling functions affect the overall accuracy. We use the accuracy and coverage metrics generated by the majority vote aggregator, which are recorded for varying top K labeling functions selected. This is complementary analysis is exclusive to our study and is performed using the actual labels of

Table 4.13: Domain Adaptation worst performing LFs

Labeling Function Name	Empirical Accuracy
lf_title_determiner_ratio_upper_real	0.491
lf_title_adverb_ratio_upper_fake	0.507
lf_title_verb_present_participle_ratio_upper_fake	0.516
lf_content_semicolon_ratio_upper_real	0.518
lf_title_adposition_ratio_lower_real	0.527

the unlabeled dataset. In the overall pipeline, no selection is applied and all identified labeling functions are used. Figure 4.2 shows this analysis.

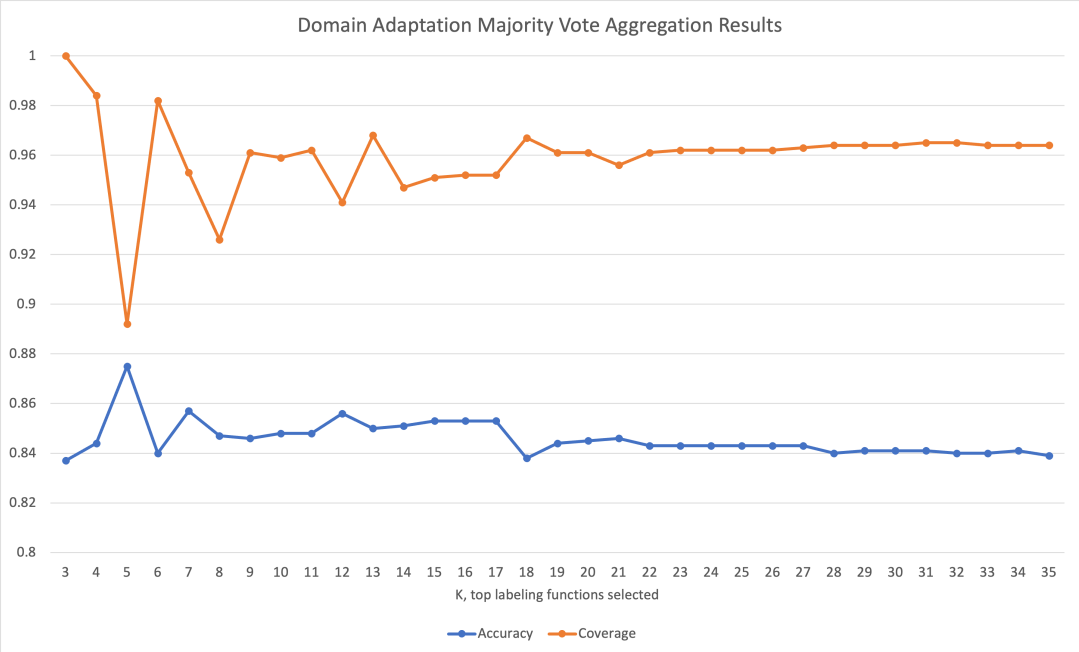

Figure 4.2: Majority vote aggregation results with changing top k labeling functions selected, domain adaptation

In domain adaptation setup, optimal aggregated accuracies are obtained at lower values of K compared to the semi-supervision setup. This is because the performance of the content-based labeling functions remains stable but the model-based labeling functions get significantly higher with a larger number of labeled data instances, resulting in higher accuracies with a smaller number of better-performing labeling functions. In contrast to the semi-supervision setup, the decrease in accuracy becomes

Table 4.14: Domain Adaptation Performance of Weak Label Aggregation Strategies

	Accuracy	Precision	Recall	F1 score	Coverage
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

more significant after reaching higher values of K, as there is a larger discrepancy in accuracy between the best and worst performing functions.

Similar to the previous setup, we compare three different weak label aggregation strategies. The Snorkel label model strategy originally generates probabilistic labels, which range from 0 to 1, showing the likelihood of data instances being fake. We round the probabilistic labels to the nearest discrete value to obtain real or fake labels. We also utilize the data instance selection capabilities of WeSTeC to select 50,000 data instances where the Snorkel label model is most confident. Approximately half of these instances are predicted as fake, close to 1, while the remaining instances are predicted as real, close to 0. Our results, including accuracy, precision, recall, F1-score and coverage metrics are presented in Table 4.14.

The results show improved accuracies for the aggregated labels compared to the previous setup. Regardless, the performance comparison among the weak label aggregation strategies remains consistent, with the majority vote strategy giving the best results and the hyper-label model outperforming the Snorkel label model, although the differences between all these strategies are negligible. Similar to the semi-supervision setup, the data elimination process significantly improves the accuracy of the aggregated weak labels, which shows the power of benefiting from the probabilistic labels.

4.4.2 Text Classification with Domain Adaptation

We use the aggregated weak labels obtained through Snorkel label model with data selection process to train the RoBERTa text classifier. The training process follows the same steps as in the previous setup, including hyperparameter tuning using WandB.

Table 4.15: Domain Adaptation Text Classification Results

Label Type	Tested Against	Accuracy	Precision	Recall	F1 score
Weakly supervised	Weak Labels	0.988	0.988	0.988	0.988
	Actual Labels	0.961	0.961	0.961	0.960
Supervised	Actual Labels	0.968	0.968	0.968	0.968

Table 4.15 shows the three way comparison similar to Table 4.9 to make the comparison between training with actual and weak labels easier.

Looking at the results we can see that the performances are significantly better than the semi-supervision setup. In addition, the comparison between models trained with actual labels and aggregated weak labels are also closer when both are tested against actual labels. Looking at the accuracy score, the difference between training on weak and actual labels is as small as 0.7%. This shows the power of our framework in domain adaptation scenarios. Similar to the previous setup, the text classification model achieves higher accuracies than the accuracy of the aggregated weak label, which is 0.948. This again proves that the classification models trained with weak labels can generalize beyond the aggregated weak labels provided to them, even when tested against actual labels.

4.5 Comparison with existing weakly supervised fake news detection studies

We compare the results obtained for both of our setups with existing weakly supervised learning approaches. We conduct two different analyses. In the first one, results of our aggregated weak labels are compared with existing weakly supervised fake news detection algorithms that run on the idea of labeling function application and aggregation. In the second analysis, we compare the fake news detection capability of our text classification models with existing fake news detection algorithms.

Table 4.16: Domain Adaptation Text Classification Results

	Accuracy	F1 score	Coverage
Ozgobek et al., manual Snorkel	0.700	0.720	0.860
Ozgobek et al., automated Snorkel, Acc >65%	0.710	0.740	0.860
Ozgobek et al., Snuba, DT, 3	0.765	0.765	0.902
WeSTeC, semi-supervision	0.794	0.798	0.963
WeSTeC, semi-supervision, w/ data elimination	0.931	0.931	1.000
WeSTeC, domain adaptation	0.839	0.835	0.964
WeSTeC, domain adaptation, w/ data elimination	0.948	0.948	1.000

4.5.1 Evaluation of Weak Labels

We analyze the performance of the weak labeling module of the proposed framework by comparing it with the similar state-of-the-art studies. Many of the studies that focus on weakly supervised fake news detection do not include metrics for how well the generated weak labels perform. Instead, they only provide the final classification accuracy of models trained with the generated weak labels. However, the study by Ozgobek et al. [28] which is one the baseline studies for our work, also shows the effectiveness of their weak labels. We compare our aggregated weak label performances with their results in Table 4.16. Ozgobek et al. conducted tests using three different strategies for generating and aggregating weak labels, two of which are based on Snorkel and one based on Snuba. They argue that Snuba outperforms the rest. We provide our results for both the semi-supervision and domain adaptation setups. In all calculations, weak labels generated are evaluated against the actual labels that were initially available.

In both the semi-supervision and domain adaptation settings, our framework outperforms the best results obtained by Ozgobek et al. [28], showing the superiority of our approach and the improvements made over the base idea. In the domain adaptation setup, the performance increases, showing the domain-independent capabilities of the weak labeling strategies we utilized. Furthermore, the introduction of data elimination layer significantly improves the accuracy of the weak labels for both setups.

4.5.2 Evaluation of Fake News Detection

We show the comparison between the proposed model and other weakly supervised fake news detection studies. We provide a brief description of each study we include in our evaluation.

- **TDSL** [31] Dong et al. introduce a semi-supervised learning framework for timely fake news detection through two paths CNN. Small amounts of labeled data is 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 ratio separately.
- **AA-HGNN** [33] Ren et al. 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 semi-supervision setup.
- **SSLNews** [32] Konkobo et al. developed a three path CNN based deep learning model for early detection of fake news. They mainly utilize user interactions through comments. Their experiments are conducted on a setup where labeled to unlabeled data ratio is 25%.
- **MDA-WS** [34] In their study, Li et al. 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 2-fold cross validation fashion. We take the average of all three results to include in our evaluation.
- **MWSS** [30] Shu 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.
- **FND-NS** [27] Raza and Ding 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.

Table 4.17: Fake News Detection Results

Method	Accuracy	F1 score
TDSL, 1% LDR [31]	0.798	0.886
TDSL, 30% LDR [31]	0.834	0.909
AA-HGNN [33]	0.675	0.639
SSLNews, 25% LDR [32]	0.695	-
MDA-WS [34]	0.769	0.768
CNN-MWSS [30]	0.795	0.805
RoBERTa-MWSS [30]	0.810	0.810
FND-NS, domain adaptation [27]	0.748	0.749
Ozgobek et al., <1% LDR [28]	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

- **Ozgobek et al.** [28] In their study, Ozgobek et al. proposed a weakly supervised fake news detection model using only content-based features. They utilized Snuba to weakly label fake news articles in 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 4.17. 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 all data instances.

The results show that our approach outperforms all state-of-the-art baselines in both the semi-supervision and domain adaptation setups. All compared baselines except the work by Raza and Ding [27] utilize a single weak supervision source. 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, our proposed framework allows for the integration of many content fea-

tures, 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 Ozgobek et al. [28] 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 provides, many content features can be included without compromising the overall performance of the aggregated weak labels.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, a framework for timely detection of fake news on emerging topics is proposed, utilizing weak supervision approaches. When a new topic emerges on the open web, traditional approaches such as supervised learning and fact-check based detection mechanisms fail to effectively address the early detection of fake news due to the lack of prior knowledge, labeled datasets, or fact-check articles. Our work demonstrate how to programmatically label large-scale datasets related to emerging topics and then utilize traditional supervised learning approaches using the automatically assigned weak labels.

We consider two essential setups for early detection of fake news, drawing inspiration from real-world use cases. The first one is the semi-supervision setup, where only a small amount of labeled data available. Using this limited number of labeled data, we propose ways to programmatically label large-scale datasets. The second setup is domain adaptation where we have labeled data from certain domains, and our objective is to use these data to effectively label large-scale dataset related to an emerging topic. Most of the time, enterprises have access to labeled data from different topics and past events, which is important to consider when dealing with the emerging fake news problem. The domain adaptation setup aims to cover this scenario.

In order to easily accommodate both setups, we introduce an end-to-end weakly supervised text classification framework, WeSTeC. Although there are various tools for weakly supervised learning and text classification, there is a need for a system that can perform weakly supervised text classification tasks end-to-end, using the already available tools. WeSTeC enables us to conveniently experiment with both semi-supervision and domain adaptation approaches for the text classification task at

hand.

The proposed framework consists of two main modules: weak labeling and text classification. The weak labeling module is responsible for programmatically labeling a large-scale unlabeled dataset, by learning from a second provided labeled dataset. In the semi-supervision setup, we utilize the two-input structure of WeSTeC to provide small amounts of labeled data to the system, while in domain adaptation setup, we supply a labeled dataset from another domain. This enables us to use the proposed framework in multiple scenarios. The text classification module uses the generated and aggregated weak labels to train supervised machine learning models that have been successful in many text classification use cases. The framework outputs a ready-to-use trained text classification model specialized for the large-scale unlabeled dataset.

The weak labeling module combines multiple weak labeling strategies, which can be listed as content-based and model-based labeling functions. Content-based labeling functions utilize widely adopted content features, including stylistic, complexity and readability measures. WeSTeC automatically generates content-based labeling functions for each text feature in the dataset. It then learns feature thresholds using the labeled dataset provided as input to the framework, eliminating features that are identified as not-distinctive during the threshold selection phase.

Model-based labeling functions take advantage of machine learning models trained with labeled dataset. When applied to the unlabeled dataset, the assigned labels cannot be utilized as single weak labels. However, the framework leverages them as one of many weak supervision sources, aiming to generalize beyond them. After generating up to 144 labeling functions from both approaches, the framework automatically applies them to the large-scale unlabeled dataset.

This is followed by the weak label aggregation step, aiming to combine all weak labels generated by each labeling function into a single weak label per data instance. WeSTeC supports multiple weak label aggregation strategies. On top of this, the framework leverages the probabilistic aggregated labels generated by the Snorkel label model to perform data selection before moving onto the text classification phase.

The dataset with aggregated weak labels is passed onto the text classification stage, where a supervised text classification model is trained using the aggregated weak labels. WeSTeC supports RoBERTa text classification, fine-tuning the pre-trained text classification model. We evaluate our work by measuring the performance of our aggregated weak labels, as well as quantifying the performance of our trained fake news classifier. We perform evaluations in both semi-supervision and domain adaptation setups. Our weak labeling pipeline outperforms all baseline studies in both setups.

We attribute the superiority of our approach to two qualifications of our weak labeling module: the ability to combine multiple weak labeling sources and seamless utilization of many content features without worrying about degrading the overall performance of aggregated weak labels, thanks to automatic feature selection layer. Our fake news classification model outperform all state-of-the-art weakly supervised fake news detection studies. In both setups, our models trained with weak labels achieve accuracies as close as 1-2% to the models trained with actual labels under the exact same conditions. This reassures the quality of our weak labeling process and validates the effectiveness of our weakly supervised learning techniques.

5.1 Important Achievements

We have introduced an end-to-end weakly supervised text classification framework, which is not only suitable for fake news detection but also applicable to many other text classification tasks. The learning curve for adapting various weakly supervised learning tools and libraries is high, and similar steps are required when utilizing different tools and libraries for text classification tasks. With WeSTeC, we enable users to easily execute fully automated weakly supervised text classification pipelines by providing only three inputs: necessary configuration parameters, a labeled dataset and an unlabeled dataset. The versatile structure of WeSTeC enables its use on different setups, including but not limited to the two setups that we experimented on: semi-supervision and domain adaptation. We believe that WeSTeC fills an important gap in the weak supervision technology landscape.

The proposed framework combines some of the important text classification and fake news detection techniques with novel improvements such as combining multiple weak labeling approaches and automatic content feature elimination. These additions have enabled us to outperform all weakly supervised fake news detection baselines. In addition, WeSTeC allows us to experiment with different alternatives of same steps to make sure highest performing alternative is selected. For example, having access to three different weak label aggregation strategies, we are able to visualize how each approach performs in our case. Using all these advancements, we have developed a weakly supervised fake news detection approach that outperforms all baselines in both aggregated weak label and fake news detection classifier performances.

5.2 Future Work

We aim to make WeSTeC available as an open-source framework for everyone. We initially want to structure the codebase and add documentation. This will enable both public use of the framework and encourage contributors to improve its functionality. Currently, the framework supports two different weak labeling strategies, three different weak label aggregation approaches and a single text classification alternative. Our goal is to expand the available options for many aspects of the framework, including but not limited to the ones mentioned. We have an intention to increase the number of content features, starting with psychological and LIWC features. In addition, we aim to improve the threshold selection algorithm in two areas. The algorithm currently supports the threshold selection for binary classification only. We plan to enhance it to also support multi class classification. The second area of improvement is to use better statistics to determine the thresholds. We want to explore the use of more complex statistics to better understand and optimize the threshold selection process.

We aim to validate the effectiveness of the proposed framework in text classification tasks other than fake news detection. For the fake news detection task, we seek access to manually labeled datasets that are not publicly available, to better understand how the framework performs with large-scale datasets containing manual labels. Also, we want to further analyze the potency of the two different types of labeling functions we explored in this study, when used with different datasets. Based on our analysis,

our objective is to enhance the framework to better accommodate different types of datasets including those with different labels, topics or domains. This will allow the framework to be more adaptable and effective in handling diverse text classification tasks.

REFERENCES

[1] A. J. Ratner, S. H. Bach, H. R. Ehrenberg, J. A. Fries, S. Wu, and C. Ré, "Snorkel: rapid training data creation with weak supervision," *The VLDB Journal*, vol. 29, pp. 709 – 730, 2017.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, p. 1–47, mar 2002.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, (Red Hook, NY, USA), p. 3111–3119, Curran Associates Inc., 2013.

[4] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[5] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1746–1751, Association for Computational Linguistics, Oct. 2014.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of

deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[10] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[11] G. Lample and A. Conneau, "Cross-lingual language model pretraining," in *Neural Information Processing Systems*, 2019.

[12] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Neural Information Processing Systems*, 2019.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[14] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, pp. 44–53, 2018.

[15] P. Varma and C. Ré, "Snuba: Automating weak supervision to label training data," *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, vol. 12 3, pp. 223–236, 2018.

[16] B. D. Horne and S. Adali, "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*, vol. abs/1703.09398, 2017.

[17] J. Pennebaker, M. Francis, and R. Booth, "Linguistic inquiry and word count (liwc)," 01 1999.

[18] O. Ngada and B. Haskins, "Fake news detection using content-based features and machine learning," *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pp. 1–6, 2020.

[19] Y. Qin, D. Wurzer, V. Lavrenko, and C. Tang, "Spotting rumors via novelty detection," *ArXiv*, vol. abs/1611.06322, 2016.

[20] V. L. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," 2016.

[21] S. Castelo, T. G. Almeida, A. Elghafari, A. Santos, K. Pham, E. F. Nakamura, and J. Freire, "A topic-agnostic approach for identifying fake news pages," *Companion Proceedings of The 2019 World Wide Web Conference*, 2019.

[22] S. Rastogi and D. Bansal, "A review on fake news detection 3t's: typology, time of detection, taxonomies," *International Journal of Information Security*, vol. 22, pp. 177 – 212, 2022.

[23] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Inf. Sci.*, vol. 497, pp. 38–55, 2019.

[24] X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Inf. Process. Manag.*, vol. 57, p. 102025, 2020.

[25] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in *International Conference on Computational Linguistics*, 2017.

[26] V. K. Singh, I. Ghosh, and D. Sonagara, "Detecting fake news stories via multimodal analysis," *Journal of the Association for Information Science and Technology*, vol. 72, pp. 17 – 3, 2020.

[27] S. Raza and C. Ding, "Fake news detection based on news content and social contexts: a transformer-based approach," *International Journal of Data Science and Analytics*, vol. 13, pp. 335 – 362, 2022.

[28] Ö. Özgöbek, B. Kille, A. R. From, and I. U. Netland, "Fake news detection by weakly supervised learning based on content features," in *NAIS*, 2022.

[29] A. R. From and I. U. Netland, "Fake news detection by weakly supervised learning," Master's thesis, Høgskoleringen 1, 7034 Trondheim, Norwa, 2021.

[30] K. Shu, G. Zheng, Y. Li, S. Mukherjee, A. H. Awadallah, S. W. Ruston, and H. Liu, "Leveraging multi-source weak social supervision for early detection of fake news," *ArXiv*, vol. abs/2004.01732, 2020.

[31] X. Dong, U. Victor, and L. Qian, "Two-path deep semisupervised learning for timely fake news detection," *IEEE Transactions on Computational Social Systems*, vol. 7, pp. 1386–1398, 2020.

[32] P. M. Konkobo, R. Zhang, S. Huang, T. T. Minoungou, J. A. Ouedraogo, and L. Li, "A deep learning model for early detection of fake news on social media*," *2020 7th International Conference on Behavioural and Social Computing (BESC)*, pp. 1–6, 2020.

[33] Y. Ren, B. Wang, J. Zhang, and Y. Chang, "Adversarial active learning based heterogeneous graph neural network for fake news detection," *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 452–461, 2020.

[34] Y. Li, K. Lee, N. Kordzadeh, B. D. Faber, C. Fiddes, E. Chen, and K. Shu, "Multi-source domain adaptation with weak supervision for early fake news detection," *2021 IEEE International Conference on Big Data (Big Data)*, pp. 668–676, 2021.

[35] M. Samadi, M. Mousavian, and S. Momtazi, "Deep contextualized text representation and learning for fake news detection," *Inf. Process. Manag.*, vol. 58, p. 102723, 2021.

[36] Z. Dai, G. Lai, Y. Yang, and Q. V. Le, "Funnel-transformer: Filtering out sequential redundancy for efficient language processing," *ArXiv*, vol. abs/2006.03236, 2020.

[37] N. Kant, R. Puri, N. Yakovenko, and B. Catanzaro, "Practical text classification with large pre-trained language models," *ArXiv*, vol. abs/1812.01207, 2018.

[38] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification algorithms: From text to predictions," *Inf.*, vol. 13, p. 83, 2022.

[39] R. Wu, S.-E. Chen, J. Zhang, and X. Chu, "Learning hyper label model for programmatic weak supervision," in *The Eleventh International Conference on Learning Representations*, 2023.

[40] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[41] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[42] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.

[43] A. D'Ulizia, M. C. Caschera, P. Grifoni, and F. Ferri, "Fake news detection: A survey of evaluation datasets," *PeerJ Computer Science*, vol. 7:e518, 06 2021.

[44] L. Hu, S. Wei, Z. Zhao, and B. Wu, "Deep learning for fake news detection: A comprehensive survey," *AI Open*, vol. 3, pp. 133–155, 2022.

[45] J. Nørregaard, B. D. Horne, and S. Adali, "Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles," *ArXiv*, vol. abs/2102.04567, 2019.

[46] H. Schutze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press, 2008.

APPENDIX A

SPACY PART OF SPEECH TAGGING

The part of speech tagging is a natural language processing (NLP) task that assigns grammatical information to each work in a sentence. Some of part of speech examples can be listed as nouns, verb, adjectives. We use Spacy library in our work to assign POS tags to sentences of news articles to then extract POS tagging and punctuation features. More information on this process can be seen in 3.3.1.1.

The library assigns two identifier to each word in a sentence after applying, which are identified as POS and TAG. POS identifies the high level part of speech of a word. Each POS has a set of fine-grained tags, which are identified by TAG. An example to POS can be given as adjective and an example TAG can be superlative adjective, which is more fine-grained. We provide list of POS and TAG identifiers we utilized and their description in A.1.

Table A.1: Spacy POS and TAG descriptions

POS	POS description	TAG	TAG Description
ADJ	Adjective		
ADP	Adposition		
ADV	Adverb	RBR	Comparative Adverb
ADV	Adverb	RBS	Superlative Adverb
DET	Determiner		
INTJ	Interjection		
NOUN	Noun		
NUM	Numeral	CD	Cardinal Number
PRON	Pronoun	EX	Existential Pronoun
PRON	Pronoun	PRP	Personal Pronoun
PRON	Pronoun	WDT	WH-Determiner
PRON	Pronoun	WP	Personal WH-Pronoun
PROPN	Proper Noun		
PUNCT	Punctuation		
SYM	Symbol		
VERB	Verb	VBG	Gerund or Present Participle Verb
VERB	Verb	VBN	Past Participle Verb
VERB	Verb	VBZ	Third Person Singular Present Verb
AUX	Auxiliary	MD	Modal