QATAR UNIVERSITY

COLLEGE OF ENGINEERING

DETECTING USERS PRONE TO SPREAD FAKE NEWS ON TWITTER

BY

ZIEN SHEIKH ALI

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computing

January 2023

COMMITTEE PAGE

The members of the Committee approve the Thesis of

Zien Sheikh Ali defended on 27/12/2022.

Dr. Abdulaziz Al-Ali
Thesis Supervisor

Dr. Tamer Elsayed
Thesis Co-Supervisor

Dr. Paolo Rosso
Committee Member

Dr. Junaid Qadir
Committee Member

Dr. Uvais Qidwai
Committee Member

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

Sheikh Ali, Zien, A., Masters : January: 2023, Master of Science in Computing

Title: Detecting Users Prone to Spread Fake News on Twitter

Supervisor of Thesis: Dr. Abdulaziz Al-Ali.

The spread of misinformation has become a major concern to our society, and social media is one of its main culprits. Evidently, health misinformation related to vaccinations has slowed down global efforts to fight the COVID-19 pandemic. Studies have shown that fake news spreads substantially faster than real news on social networks. One way to limit this fast dissemination is by assessing information sources in a semi-automatic way. In this thesis, we aim to identify users who are prone to spread fake news in Arabic Twitter. These users play an important role in misinformation propagation and identifying them is beneficial in controlling the spread of misinformation on social media. To identify users that are prone to spread misinformation, we need to have examples of previously spread fake news. Thus, we collected Arabic verified claims from Arabic fact-checking websites, then we collected tweets relevant to the claims and manually annotated them. Lastly, we identified unique users and labeled them as prone to spread fake news or not prone to spread fake news. Since a user is deemed prone to spread fake news either by tweeting or retweeting fake claims, two datasets were created. The first dataset is annotated based on verified tweets shared by the user, it consists of 1,317 users, of which 272 are prone to spread fake news. The second dataset is annotated based on verified tweets and retweets shared by the user, it consists of 1,546 users, of which 541 are prone to spread fake news. We use features extracted from users' recent

tweets (e.g., linguistic, statistical, and profile features) to predict whether they are prone to spread fake news or not. We introduced three new statistical features: average days between tweets, number of quoted retweets and average user engagement. To tackle the classification task, multiple machine learning models are employed (XGBoost, Logistic Regression, Random Forests, and Neural Networks) and evaluated. Empirical results reveal promising detection performance where an F1 score of 0.73 was achieved by the logistic classification model. When tested on a publicly available English benchmark dataset, our approach has outperformed the current state-of-the-art for this task. Lastly, we show that our model can be deployed as a real-time API service to predict user credibility.

DEDICATION

This thesis is dedicated to my family, for their continuous support and encouragement.

ACKNOWLEDGMENTS

First and foremost, ultimate thanks to Allah for all the opportunities, blessings and strength that have been showered on me to work on this thesis.

I would like to express my gratitude and sincere thanks to my supervisors Dr. Abdulaziz and Dr. Tamer, for their continuous support, guidance and feedback. This thesis would not have been completed without their valuable supervision and mentorship.

I also want to thank Dr. Walid Magdy for his valuable feedback and advice on this thesis. I extend my appreciation to my colleagues, BigIR team members, for their support and assistance.

Lastly, my deepest appreciation goes to all my family members. My dear parents, siblings, and husband. Thank you for your continuous love and prayers. Thank you for always believing in me and encouraging me to pursue my master's degree.

This work was made possible by NPRP grant No.: NPRP11S-1204-170060 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ArPFN_TRT Arabic dataset variant that uses tweets and retweets

ArPFN_T Arabic dataset variant that uses tweets

EN_PAN English Dataset used in PANs shared task

PAN_2020+ improved version of PANs winning participation

PAN_2020 PANs winning participation

CHAPTER 1: INTRODUCTION

Over the past two decades, social media has gained significant popularity around the world. According to a report on Digital Trends, the number of active social media users has reached 4.1 billion in 2020. [1] The Middle East region has also experienced rapid and constant growth in social media consumption. A study performed by Radcliffe et al. [1] has shown that social media users in the Middle East and Africa spend an average of 3.5 hours per day on social networks, which is the longest time in comparison with America, Europe, and Asia Pacific. The study states that given the large dependency on social media, it has become the most common news source for Arab youth. In fact, 79% of Arab youth in 2020 got their news sources from social media, while in 2015, only 25% of Arab youth used social media as their news source.

People find social networks an easier and more accessible method of consuming information compared to traditional media sources. Social networks provide a platform for news outlets, governments, and public figures to communicate the latest news in a brief manner and engage with their followers [2]. Social networks can be especially useful for seeking real-time information during a risk or crisis, allowing any user or witness on the scene to share updates. Twitter has specifically evolved into a popular social media platform for news sharing. It allows tweets to reach a larger audience quickly through retweets and likes. While Twitter can be an effective tool to express thoughts and engage with authorities and organizations, it is also misused to generate fabricated information and occasionally to manipulate the public opinion.

Misinformation is defined as false or inaccurate information. While disinformation

[1] https://thenextweb.com/news/digital-trends-2020-every-single-stat-you-need-to-know-about-the-internet

is defined as the false information that maliciously intends to mislead the readers.[2] In this thesis, the focus is primarily on misinformation, rather than disinformation. Misinformation can spread faster, deeper and wider in social networks compared to traditional media sources [2]. This wide spread of misinformation causes a serious impact on society and individuals. In the past few years, Arabic social media has been utilized to spread state propaganda, attack political parties, and mislead the society [3].

Moreover, with the recent COVID-19 outbreak, there has been a rapid dissemination of health-related messages. The World Health Organization (WHO) has consequently coined the term 'infodemic,' defined as an excess of information, including false and misleading information, during a disease outbreak.[3] Health related misinformation is also proliferating on Arabic social media [4]. Spreading anti-vaccine misinformation has contributed to large vaccination hesitancy, which is now hindering the national global efforts to fight the pandemic. Misinformation nowadays is not only used as a political weapon, but it also poses a serious risk to society and public health.

Previous studies have targeted misinformation on Arabic social media from a content-based perspective by verifying the content of a single post or tweet [5]–[8]. However, only a few studies explore this task from a source-based perspective. The spread of misinformation can be effectively mitigated by identifying the credibility of the source of the information [9]. In social media, users are contributing to the spread of fake news by retweeting and engaging with the information. It was found by Shao et al. [10] that fake news tends to attract both malicious and normal users. The goal of malicious users is to achieve personal benefits, while normal users often spread misinformation unintentionally without knowing the veracity of the information. Moreover, people tend

[2] https://www.unhcr.org/innovation/wp-content/uploads/2022/02/Factsheet-4.pdf
[3] https://www.who.int/health-topics/infodemic

to share fake news that mimics the style of contemporary news to gain more popularity or engagement on social media. According to Pennycook et al. [11] people often spread misinformation because they wrongly believe it is accurate.

Contrary to previous studies that target malicious users that intentionally spread misinformation (e.g. bots [12] and trolls [13]), our work is concerned with users that are prone to spread fake news. We define these users as *users that contribute to the diffusion and amplification of misinformation on Twitter, either intentionally or unintentionally*. Recognizing that type of users on Twitter is an important task that can be employed in combating the spread of fake news. For example, a tool to identify fake news spreaders can be an explicit addition to fake news detection systems. Moreover, identifying credible users will help social media consumers in finding trustworthy information.

1.1. Problem Statement

In this thesis, we aim to identify users prone to spread fake news on Arabic Twitter. Given a Twitter user, our objective is to identify whether they are likely to be prone to spread Arabic fake news or not using their recent tweets and Twitter profile information.

Due to the lack of Arabic datasets for this task, we propose a data collection pipeline to collect claims, tweets and users for this task. We explored a range of different features extracted from the user timeline such as textual, profile, statistical and emotional features. We evaluated the performance of multiple learning algorithms on our Arabic dataset, as well as a publicly available English benchmark dataset.

1.2. Research Questions

To gain a comprehensive understanding of the problem at hand, we focus on the following research questions:

RQ1 How effective are traditional machine learning methods in automatically detecting users that are prone to spread Arabic fake news?

RQ1.1 How effective are the existing baselines for the task?

RQ1.2 Which feature category combination exhibits the best performance?

RQ1.3 How does the classifier perform when contextualized embeddings are used instead of word n-grams?

RQ2 What is the effect of increasing the number of considered user tweets for feature extraction on the performance of the classifier?

RQ3 What is the effect on classification performance when considering users who retweet claims in the training set?

RQ4 How effective is our methodology on an English dataset?

1.3. Contributions

The contributions of this thesis are summarized as follows:

- We collect and publicly share the first large Arabic dataset of naturally occurring claims, AraFacts.[4] We used AraFacts as the seed to generate our dataset of users.

[4] https://gitlab.com/bigirqu/AraFacts

- We propose a method for generating a Twitter user dataset using a set of previously verified claims.
- We propose the first model to detect users prone to spread Arabic fake news.
- We deploy an Application Programming Interface (API) service to classify if a user is prone to spread fake news in real-time.

The rest of this thesis is organized as follows; Chapter 2 discusses related work. Chapter 3 describes our data collection process. Chapter 4 outlines our proposed methods. Chapter 5 presents our experimental setup, results and discussion. Chapter 6 describes the web API creation and chapter 7 concludes the thesis and discusses ethical considerations.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we review the literature for related work on the task of detecting misinformation spreaders on social media. We organize our literature review into three main sections. We start by covering different user profiling tasks to identify malicious users (Section 2.1). Second, we discuss efforts on profiling and investigating users that spread misinformation (Section 2.2). Third, we review the different approaches to collect Twitter users and identify their role in spreading fake news (Section 2.3).

2.1. User Profiling in Social Media

Credibility of social media users is concerned with user profiling. Researchers have attempted to classify users for different purposes. Some efforts focused on identifying malicious users such as paid trolls [14]. These types of users are typically paid to manipulate others to support certain political parties or PR agencies. The research conducted by Mihaylov et al. [14] uses Support Vector Machines (SVM) to classify users in community forums as either troll or non-troll using a combination of features derived from their profiles, comments and the type of engagement they received on their comments. Other malicious accounts could be operated by unpaid trolls [13], Sockpuppets [15], water army [16] or hateful users [17].

Moreover, Darwish et al. [18] aim to identify seminar users in Arabic Twitter. They define seminar users as Twitter users that are politically oriented and dedicate their accounts to support a specific entity. Unlike trolls, seminar users show their real identity on social media instead of using a fake account.

Numerous studies were done on identifying social bots on Twitter [12], [19], [20]. Social bots are accounts that are partly controlled by a software. Malicious bot accounts

are known for engaging in opinion manipulation campaigns and spreading information with low credibility [21].

The studies presented thus far aim to classify a particular kind of malicious users. They are mostly users that intentionally try to manipulate public opinion for personal and political purposes. Unlike the work above, our aim is to detect users that are prone to spread misinformation without specifying their political orientation or intentions.

2.2. Classifying Misinformation Spreaders on Social Media

The task of classifying misinformation spreaders remains under-explored. Recent attention has focused on identifying social media users that spread fake news. Rangel et al. organized an Author Profiling Shared task at CLEF 2020 [22]. The goal of the shared task is twofold; first, to verify the viability of automatically classifying potential fake news spreaders on Twitter. Second, to show the difficulty of the task when users are sharing tweets other than misinformation. The task is defined as follows, given a Twitter user's recent tweets, determine whether they are keen to spread fake news or not. The authors provide a corpus of Twitter users and their recent 100 tweets. The languages covered are English and Spanish only. The task received 66 participants, and the highest Accuracy scores achieved were 75% on the English dataset and 82% on the Spanish dataset.

It is worth mentioning that the highest performance was achieved using a stacked ensemble classifier of five machine learning algorithms; four of the base models use character n-grams as features, while the fifth model uses features based on statistics of the tweets such as the average length of the tweets [23]. All the highest six participants in the task used a combination of n-grams and traditional machine learning approaches.

Since the organizers provide only the text of the recent 100 tweets for each user, the features extracted by the participants were limited to a combination of textual features, stylistic features, personality features and emotions.

Giachanou et al. [24] explored the role of psycholinguistic features in distinguishing between fake news spreaders and fact-checkers. The psycholinguistic features include sentiment, emotions, linguistic features, personality traits, readability, and communication style. Their analysis suggests that spreaders of fake news use more informal language than checkers. Moreover, checkers of fake news use more positive words in comparison to spreaders. Personality features derived from the user's tweets have been effectively utilized in related work [24], [25]. However, since no personality features extraction tools exist for Arabic, they were not used in this thesis.

Rath et al. [26] proposed a fake news spreader detection model using an inductive representation framework. Users that are more likely to spread misinformation are identified, given a tweet and a directed social network. They built a social graph of Twitter users and define modular communities using Community Health Assessment (CHA) model. The graph embeddings are learned using GraphSAGE framework. The approach proposed by Rath et al. identifies fake news spreaders related to a given tweet, while our approach identifies users independently using linguistic and profile features of the user. We do not include social network features such as the following and follower networks of the user.

Del Tredici and Fernandez [27] proposed a model for fake news detection by utilizing the tweets of users spreading the news. Their idea is that users prone to spread fake news exhibit unique linguistic trends. The results of Del Tredici and Fernandez show that fake news detection can be improved by leveraging user representations derived using

linguistic features only.

A number of studies have begun to examine the characteristics of users that are likely to spread fake news. Shu et al. [28] investigate the role of user profiles for fake news detection, they characterize Twitter users that are more likely to spread fake news or real news and study multiple implicit and explicit profile features to discover the user characteristics that are useful for fake news detection. Their experiments show that user features such as registration time, account verification, political bias and personality type could make a significant impact in detecting fake news.

Moreover, Shao et al. [10] performed a study on Twitter super-spreaders; these are Twitter accounts that spread low-credibility content related to the 2016 U.S. presidential campaigns and elections. One of the main findings of the study was that super-spreaders amplify misinformation by continuously spreading it and tagging influential users with high number of followers. Moreover, the study has shown that super-spreaders are more likely to be social bots and suggests mitigating the spread of online misinformation by limiting social bots on Twitter.

2.3. Annotated User Datasets

Current Twitter fake news datasets are catered for verification of tweets (tweet-level verification) [8]. Limited datasets identify the role of users in the spread of fake news. We summarize the different approaches to collect Twitter users next.

Rangel et al. [22] constructed a corpus of 500 English users and 500 Spanish for PAN 2020 shared task to detect users keen to spread fake news on Twitter. The corpus was constructed as follows; first, false claims debunked by fact-checking websites (e.g. PolitiFact and Snopes) are collected. Then, Twitter is searched to find tweets relevant to

these claims, where the tweets are labeled as supporting a claim or not. After annotating the tweets, the users are labeled as keen to spread fake news or not based on whether they shared at least one tweet supporting a fake claim. Finally, users with the most annotated tweets were selected.

Labelling users based on annotated tweets was similarly adopted by Shao et al. [10], where users who are super-spreaders are identified as users that continuously spread misinformation related to the 2016 US elections. Another contribution by Shu et al. [28] used verified tweets from FakeNewsNet dataset [29] to label users as likely to spread fake-news or likely to spread real-news.

Other than automated datasets, other datasets are manually annotated through crowd-sourcing, such as the collected dataset of Seminar Users [18] and BotOrNot dataset [30]. Furthermore, there are independent services that work on identifying unreliable news sources. For example, PropOrNot is a propaganda identification service that manually identifies propaganda outlets and their corresponding social media accounts. The list of unreliable sources is made publicly available on their website.[5]

Twitter Safety has also made an initiative to disclose suspended accounts that were involved in spam, coordinated activity or harmful activity.[6] Twitter publicly shared the suspended accounts tweets, account information and media.

The studies presented thus far provide solutions for profiling users who try to spread misinformation. There is however insufficient research on addressing users spreading Arabic fake news on Twitter. To fill this gap, our study focuses on identifying Arab users that are prone to spread fake news. While previous work has focused on misinformation

[5] http://www.propornot.com/p/the-list.html

[6] https://blog.twitter.com/en_us/topics/company/2018/enabling-further-research-of-information-operations-on-twitter

datasets for the task of tweet-level verification, very few studies worked on constructing datasets for user-level verification.

CHAPTER 3: DATA COLLECTION

In this section, we describe the user data collection and annotation methodology. Our goal is to collect a set of users that are prone to spread fake news and users that are likely to spread real news. To build the dataset, we modified the method used in the shared task at PAN 2020 for profiling fake news spreaders on Twitter described in Section 2.3. We constructed the dataset in three main stages. First, we collected sets of previously verified Arabic claims from multiple resources. We then used those claims to find tweets that are spreading them. Finally, we identify users associated with the tweets from the previous step and label them based on tweet or retweet frequencies. These stages are detailed in the next three subsections.

3.1. Claim Collection

In this stage, we aim to collect real claims from the Arab world and then search for tweets that are spreading the claims. To do so, we utilized an existing Arabic rumors dataset; ArCOV19-Rumors [8]. ArCOV19-Rumors is an Arabic COVID-19 Twitter dataset for misinformation, the dataset contains a set of 138 verified claims from fact-checking websites, and 9,414 relevant tweets to those claims. The tweets were manually annotated by the authors of ArCOV19-Rumors. The claims in the dataset are not limited to health misinformation, it also covers claims related to COVID-19 from multiple topical categories such as social, political, sports and entertainment. However, the limitation with ArCOV19-Rumors dataset, is that it only covers claims between January 2020 and April 2020. For our claim collection, we expect to cover a larger number of claims within a wider time-frame. Therefore, to address the limitations of ArCOV19-Rumors, we constructed AraFacts [31].

AraFacts is the first large collection of Arabic naturally-occurring claims from 5 different Arabic fact-checking websites. The claims are annotated and verified by professional fact-checkers. We collected 8,958 claims that were posted between 2016 and 2021. From each fact-checking website, we crawled the claim, its factual label, description and 10 additional meta-data. The meta-data contains information parsed from the fact-checking article in addition to information derived by us, such as normalized category, normalized label and claim type. We proposed a normalized claim rating to achieve a standard rating for all claims in AraFacts. We set our own claim rating scheme consisting of four labels *(False, Partly-False, Sarcasm, True)* and mapped 27 distinct claim labels to them. Similarly, the topical category labels were normalized from 35 distinct category labels to 8 normalized categories. We summarize the dataset fields in table 3.1.

We selected claims from AraFacts that have labels *True* and *False* only; since claims with labels *Sarcasm* and *Partly-False* contain a mixture of misleading claims and that are not completely True or False. Overall, we have collected 5,371 claims from both datasets with 299 of them being True and 5,072 being False.

Table 3.1: Information available in AraFacts dataset and their description.

Column name	Description
Claim-ID	ID of the claim
Claim	Text of the claim
Source	Name of the fact-checking website from which the claim was crawled
Description	Detailed description of the claim
Source-label	The veracity label of the claim as it appears in the fact-checking website
Normalized-label	Normalized claim label
Source-category	Topical category of the claim as it appears in the fact-checking website
Normalized-category	Normalized topical category of the claim
Date	Article publication date
Source URL	URL of the article
Claim URLs	URLs to web pages spreading the claim
Evidence URLs	URLs referenced by the fact-checker to justify their annotation
Claim type	Indicates whether the claim refers to text, an image or a video

3.2. Tweet Collection

After collecting claims, the next step is to find relevant tweets to them. We utilized the manually-annotated tweets from ArCOV19-Rumors dataset where only tweets labeled as True or False were kept, and the rest were discarded, resulting in 3,025 tweets. In the AraFacts dataset, we used the claim URLs data field, which contains URLs to Web

pages that spread each claim. We identified URLs pointing to tweets and obtained their tweet IDs. The tweets were then crawled using the Twitter API, yielding 2,981 tweets that are related to 1,213 claims.

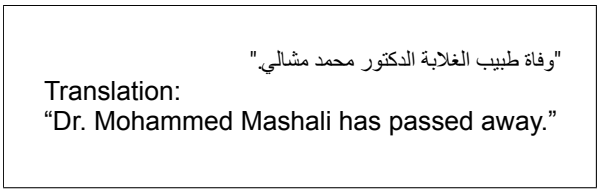
After collecting the tweets, we manually inspected a subset of 100 tweets to verify that the tweets are indeed relevant to the claim. We discovered two shortcomings with the tweets derived from AraFacts. First, 10 of the tweets were not in Arabic language. Thus, we decided to eliminate non-Arabic tweets, since the scope of this work is on Arabic misinformation. We used langdetect library[7] to detect the tweet language and removed 233 non-Arabic tweets from the Tweet collection. The second shortcoming was that some tweets were not associated with their claims. In some cases, the tweet was not spreading the same claim. Figure 3.1 shows one such example. Arguably, a user that is questioning the correctness of a claim is neutral towards it and not spreading it.

Out of the 100 tweets that we inspected, 9 were found to be irrelevant to their claims. This has prompted us to manually annotate all tweet-claim pairs to verify their relevancy to the claim. We chose to not annotate ArCOV19-Rumors tweets since they were already manually annotated.

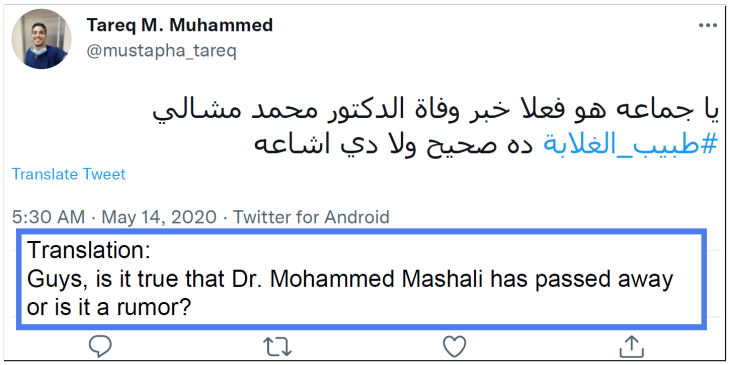
The annotation task was performed by one annotator who was asked to read the tweet and the claim, then label the tweet as:

- Expressing the claim: if the tweet is sharing, restating or rephrasing the same claim
- Negating the claim: if the tweet is debunking or denying the claim

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



(a) Claim text of a claim from AraFacts dataset



(b) Tweet corresponding to the claim

Figure 3.1: Example of a tweet and claim pairs obtained from AraFacts. The example above shows that the tweet is not spreading the claim, it is questioning whether the claim is true or not.

- Other: if the tweet is questioning the claim or the tweet is irrelevant to the claim

The detailed annotation guidelines can be found in Appendix A. The results of the annotation task are presented in Table 3.2. Evidently, 95% of the tweet-claim pairs were labeled correctly by the fact-checkers and 4.5% of tweets were labeled Other; meaning they are not relevant or not spreading the claim. We unexpectedly identified 7 tweets negating the claim, these could have been added erroneously by the fact-checkers.

Table 3.2: Results for tweet-claim annotation task.

Annotation	Number of tweets
Expressing the claim	2,474
Other	125
Negating the claim	7

Once annotation was complete, we eliminated the tweets that are labeled as *Other*

and changed the label of tweets that are labeled as *Negating* (i.e. a tweet that negates a True claim is labeled False and vice versa). Finally, we collected retweets to all the verified tweets from AraFacts using Twitter API.[8] Unlike AraFacts, the retweets for ArCOV19-Rumors were publicly available.[9] The total number of collected retweets is 35,698.

3.3. User Collection

Since our annotated tweet collection is limited to only ArCOV19-Rumors tweets and a small subset of AraFacts claims (only 1,213 claims have annotated tweets), this step aims to capture more associated claims to each user by searching the users' timelines for occurrences of other claims from our collection.

Thus, we needed to capture more user-claim associations by searching user timelines for occurrences of other claims from our collection. We started by using the collected tweets to identify unique users with at least 1 tweet in ArCOV19-Rumors or AraFacts. Consequently, 4,176 unique users were found. For each user, we used Twitter API to collect their timelines. The maximum number of tweets that can be crawled per user is 3,200 tweets. We then searched users' timelines for claims using all 5,371 claims from our collection.

For each user timeline, we used the ElasticSearch engine[10] to retrieve tweets that have high similarity with the claims' text or description. The retrieved tweets, with BM25 similarity score above 15, were manually annotated using the same annotation

[8] https://developer.twitter.com/en/docs/twitter-api/tweets/retweets/introduction

[9] https://gitlab.com/bigirqu/ArCOV-19/-/blob/master/ArCOV19-Rumors/tweet_verification

[10] https://www.elastic.co/elasticsearch/

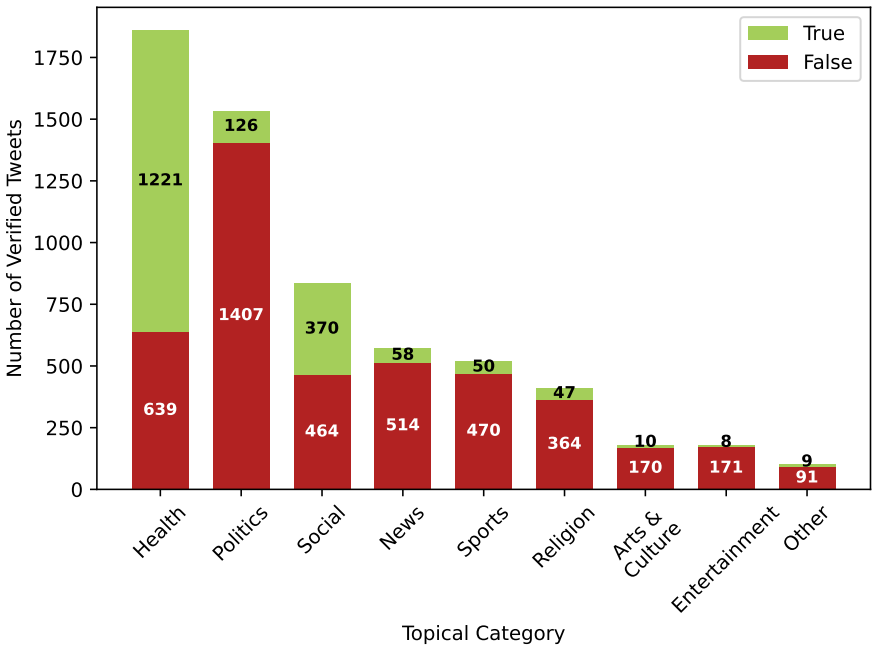


Figure 3.2: Distribution of the verified tweets and their topical categories and veracity labels.

guidelines mentioned in Section 3.2, and then appended to the tweet collection.

Table 3.3 shows the number of verified tweets from each source and their veracity label. As shown in the table, the number of False tweets obtained from AraFacts dataset is larger than the number of True tweets. This class imbalance can be attributed to the fact that fact-checking websites prioritize debunking False news over verifying True news. We also visualize our collection of verified tweets (tweets related to verified claims) in Figure 3.2 by demonstrating all 9 topical categories and their label distributions. Notably, the majority of tweets are related to Health or Politics and most fake claims are political.

Table 3.3: Summary of the verified tweets collection.

Tweet source	True Tweets	False tweets
ArCOV19-Rumors	1,625	1,948
AraFacts	191	2,431
Manually annotated tweets	133	114
All	1,949	4,493

Next, we investigate the users collection after adding the newly annotated tweets. We count the number of users in terms of their number of verified tweets or retweets. Table 3.4 summarizes the statistics. In the first two rows, we only consider verified tweets in our statistics and ignore the retweets of the verified tweets. In the third and fourth rows, we consider both tweets and retweets in our statistics.

To construct our final *labeled* user dataset, we investigate two different variants. First, we investigate users by considering their verified tweets and ignoring their retweets. We label the user as **prone to spread fake news** if they *shared at least 2 false tweets*, regardless of whether they also spread true tweets or not. On the other hand, a user is **not prone to spread fake news** if they have *shared at least one true tweet and have no record of spreading false tweets* (i.e., we did not detect any evidence that this user is prone to spread fake news, but have only some evidence of the user spreading true news.).

For the second variation of the dataset, we consider the user's tweets and retweets. We identify users that are **prone to spread fake news** as users who have *shared at least two false tweets or retweets*. Similarly, **users not prone to spread fake news** are users that have *shared at least one true tweet or retweets and no false tweets or retweets*.

Table 3.4: Number of users in terms of the number of verified tweets and retweets (RTs) they shared.

Users that shared ...	X			
	1	2	3	4
at least X true tweets and 0 false tweets	**1,045**	186	67	34
at least X false tweets	3,131	**272**	66	26
at least X true tweets or retweets and 0 false tweets or RTs	**1,005**	204	71	35
at least X false tweets or RTs	3,171	**541**	166	73

The assumption is that users associated with frequent false tweets (or retweets) are more likely to be prone to spread fake news than other users. Although we chose a threshold of two false tweets or retweets for users prone to spread fake news, this threshold can be adjusted by the practitioner to suit the task at hand. The threshold for users not prone to spread fake news was set to at least one true tweet. Admittedly, this criterion may introduce noise to this class, as we do not have enough evidence that these users did not spread any fake news that are not included in our verified set of claims.

CHAPTER 4: METHODOLOGY

In this section, we describe the features and models used to automatically identify users that are prone to spread fake news on Twitter.

4.1. Feature Extraction

For each user, we obtain recent tweets and user meta-data using Twitter API. Features that capture information about the user's activity, popularity and linguistic style are extracted. These features can be classified into the following five main categories:

1. **Profile Features**: for each user, we used some meta-data from the user's JSON object as features, and we derived 10 additional features related to the user from the meta-data. The features used have been implemented in previous studies to profile users [12], [28], [32]. Table 4.1 summarizes the extracted profile features, their type and description.

Table 4.1: Profile features extracted from users' profiles. Features marked with $\star$ are the 10 features derived using fields from the user's JSON meta-data, while the remaining features are fields from the user's JSON object without modifications.

Feature	Type	Description
default_profile	Boolean	If the user has changed the default theme or background of their profile or not.
verified	Boolean	If the user has a verified account or not.
followers_count	Integer	Number of followers the account has.
following_count	Integer	Number of users that the account is following.
favourites_count	Integer	Number of tweets that were liked by the user.
listed_count	Integer	Number of lists the user has been added to.
statuses_count	Integer	Number of tweets posted by the user.
tweet_frequency$\star$	Float	Frequency of the user's tweets, calculated as *tweets_count* divided by *account_age*.
follower_growth_rate$\star$	Float	Rate of followers growth, calculated as *followers_count* divided by *account_age*.
following_growth_rate$\star$	Float	Rate of following growth, calculated as *following_count* divided by *account_age*.
listed_ growth_rate$\star$	Float	Rate of lists growth, calculated as *lists_count* divided by *account_age*.
followers_following_ratio$\star$	Float	Number of followers compared to following, calculated as *followers_count* divided by *following_age*.
len_screen_name$\star$	Integer	Number of characters in the users screen name.
digits_in_screen_name$\star$	Integer	Number of digits in the users screen name.
name_length$\star$	Integer	Number of characters in the name of the user.
digits_in_name$\star$	Integer	Number of numerical digits in the name of the user.
description_length$\star$	Integer	Number of characters in the user's description (biography).

2. **Textual Features**: to obtain textual features, the user's recent tweets are first concatenated as one "document". We then performed the following pre-processing steps on the text:

   (a) Removed all non-alphanumeric characters from the text, except emojis. Motivated by the preprocessing performed in [23].

   (b) Replaced any URL or any media link with the tokens #URL# and #MEDIA#.

   (c) For the Arabic dataset, we used tashaphyne library[11] to clean the text by removing any figuration and normalize elongated words.

   From each user's document, we derived tf-idf word n-grams and eliminate words that appear in less than 50 documents (across all users). Additionaly, we tested multiple n-gram ranges (unigrams, bigrams, unigrams and bigrams) as a hyper-parameter for each trained model.

3. **Contextualized Embeddings**: Contextualized embeddings represent words or sentences as a dense vector in a low-dimensional space. In contextualized embeddings, words are represented based on the static and semantic meaning of the word. We used contextualized embeddings to represent each user's recent 100 tweets.

   The use of contextualized embeddings as features is motivated by the work of An et al. to predict hateful users on Twitter [17]. They obtain a user-level representation by computing Sentence-BERT (S-BERT) [33] embeddings for each user's tweet, then averaging all tweet embeddings into one 768-dimensional vector.

[11] https://github.com/linuxscout/tashaphyne/

In our experiments, we used transformer models to generate embeddings, namely, we used different variations of Bidirectional Encoder Representations from Transformers (BERT) to compute embeddings. We used three different BERT-based models that support Arabic language:

(a) **AraBERT [34]**: AraBERT is the first Arabic text based BERT transformer model. It is pre-trained on manually scraped Arabic Wikipedia pages, the Open Source International Arabic News Corpus [35] and the 1.5 billion words Arabic Corpus [36].

(b) **MARBERT [37]**: MARBERT is an Arabic BERT model that is pre-trained on a dataset of 1 Billion Arabic tweets created by the authors. The tweets involved in the dataset are tweets written in Modern Standard Arabic (MSA) and diverse Arabic dialects.

(c) **S-BERT**: Sentence-BERT is a modification of the original BERT model. It derives semantically meaningful sentence embeddings using siamese and triplet network. S-BERT is multilingual and supports 100 languages, including Arabic.

4. **Statistical Features**: we derived features from the users' recent 3,200 tweets. We used features that describe the user impact, motivated by the work proposed by Lampos et al. [38], in addition to timeline features that describe the users' activities. The proposed statistical features are listed below. The last three features are newly proposed in this work.

    - Proportion of tweets with hashtags.
    - Average number of hashtags per tweet.

- Proportion of tweets with mentions.
- Number of unique mentions in the user's timeline.
- Proportion of tweets that are replies to other users.
- Proportion of tweets that contain URLs.
- Proportion of tweets that contain media, e.g., images or videos.
- Proportion of tweets that are retweets.
- Proportion of tweets that are quote retweets. $*$
- Average engagement of the user, computed as the average number of retweets and likes per tweet. $*$
- Average days between tweets. $*$

5. **Emotional Features:** Several researchers have utilized emotional signals for credibility assessment [39]–[41]. Moreover, multiple participants in PANs author profiling task (i.e., to detect users keen to spread fake news) used emotional signals to address the task [22], [42], [43].

   For the Arabic dataset, we used the emotion functionality in ASAD tool [44]. The extracted 11 features are; anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise and trust.

4.2. Classification Models

Traditional machine learning algorithms are an effective tool for text classification tasks. Studies have shown that the performance of traditional machine learning methods outperforms the performance of deep learning methods for small datasets [45]. For

our experiments, we selected popular supervised algorithms that are used in different text classification tasks, and we trained these models over a combination of the features described in Section 4.1. The models are detailed below:

1. **XGboost:** XGboost (eXtreme Gradient Boosting) is an ensemble algorithm based on distributed decision trees that use gradient tree boosting algorithm. XGboost is known for achieving high performance in different machine learning tasks [46]. We used XGBoost classifier library[12] and tuned the following hyperparameters: eta, gamma and maximum depth of a tree.

2. **Random Forest:** This classifier is composed of set of decision trees where each decision tree is trained on a random subset of the features. The prediction is obtained by majority voting of the predictions of all trees in the forest. Random Forest is an effective algorithm that has been successfully used in text classification tasks [47]. We used Random Forest implementation by scikit-learn[13] and tuned its associated hyperparameters; number of decision trees and maximum depth.

3. **Feed-Forward Neural Network:** Feed forward neural networks are artificial neural networks with multiple layers. They are effective for complex non-linear tasks. We used a simple feed forward neural network implemented by scikit-learn[14] with 1 hidden layer and 100 neurons. We tuned the optimizer of the neural network to reach optimal results.

4. **Logistic Regression:** Logistic Regression is a statistical model that is used to

[12] https://xgboost.readthedocs.io/en/latest/

[13] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[14] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

predict the probability of a binary event occurring. Logistic Regression is efficient to train and can be easily interpreted. We used the scikit-learn Logistic Regression implementation[15] and tuned regularization strength as a hyperparameter.

5. **Support Vector Machine:** Support Vector Machines is a powerful algorithm for text classification tasks. The objective of the algorithm is to distinctly classify data points by creating a hyperplane in an N-dimensional space when N is the number of features. We used the scikit-learn implementation for Support Vector Machines[16] and tuned the kernel and regularization strength as hyperparameters.

[15] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[16] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

CHAPTER 5: EXPERIMENTAL EVALUATION

In this section, we conduct experiments to answer the research questions. We begin by describing the experimental setup used in our experiments. Then, we present the results of our experiments to answer each research question, and discuss the results. We aim to answer the following research questions:

RQ1 How effective are traditional machine learning methods in automatically detecting users that are prone to spread Arabic fake news?

RQ1.1 How effective are the existing baselines for the task?

RQ1.2 Which feature category combination exhibits the best performance?

RQ1.3 How does the classifier perform when contextualized embeddings are used instead of word n-grams?

RQ2 What is the effect of increasing the number of considered user tweets for feature extraction on the performance of the classifier?

RQ3 What is the effect on classification performance when considering users who retweet claims in the training set?

RQ4 How effective is our methodology on an English dataset?

5.1. Datasets

We conducted our experiments on three user datasets. We used the two Arabic user datasets collected in Section 3.3. In addition to the Arabic datasets, we used the English dataset of users proposed in PANs author profiling task to predict users keen to

spread fake news. The English dataset was chosen due to the availability of existing text preprocessing tools.

For simplicity, we refer to the first Arabic dataset that considers user verified tweets only as **ArPFN_T**, the second Arabic dataset that considers both users verified tweets and retweets as **ArPFN_TRT** and the English dataset as **EN_PAN**. Table 5.1 summarizes the statistics of each dataset.

Table 5.1: Datasets used in our experiments and their statistics. PFN/NPFN denotes the number of users that are prone/not prone to spread fake news.

Dataset	PFN	NPFN	Total Users
ArPFN_T	272	1,045	1,317
ArPFN_TRT	541	1,005	1,546
EN_PAN	250	250	500

5.2. Experimental Setup

For our experiments on the Arabic datasets **ArPFN_T** and **ArPFN_TRT**, we performed nested 10-fold cross validation to tune the hyperparameters of the models. For that, we optimized for Positive-F_1 (F_1^+) score, where users prone to spread fake news represent the positive class. Since both datasets are imbalanced and the positive class is the minority, we over-sampled the positive class in training folds only. The reported results on **ArPFN_T** and **ArPFN_TRT** are the average over the 10-folds used in cross validation.

For our experiments on the English dataset **(EN_PAN)**. We maintained the data splits provided by PAN for easy comparisons. The dataset is balanced and consists of 500 users. The size of the training and test splits are 300 and 200 users, respectively. Additionally, we evaluated our models using 10-fold cross validation to be able to

perform statistical significance tests.

5.2.1. Baselines

We compare the performance of our models against the following baselines:

1. **Majority Baseline** A classifier that always predicts the label of the majority class. The majority class in both datasets is class 0 (users not prone to spread fake news) with 79% and 65% of the labels in datasets **ArPFN_T** and **ArPFN_TRT** respectively.

2. **PAN_2020** [23]: The winning participation at PANs author profiling task. The participants proposed an ensemble of five machine learning models. They replaced the typical majority voting with a logistic regression classifier that takes the outputs of the ensemble models as the input vector. The first four models (Logistic Regression, Support Vector Machines, Random Forest and XGBoost) use word n-grams as features while the fifth model (XGBoost) uses statistical features. All features are derived from the user's recent 100 tweets only. We used the authors' implementation.[17]

3. **PAN_2020+**: An improved version of **PAN_2020** that we proposed. First, we eliminated the XGBboost model from the ensemble of models as it was shown by Buda and Bolonyai that it has the least impact on the performance as per the Logistic Regression coefficients. Additionally, for the remaining models that use only tf-idf as features, we expand the feature vector by including emotional signals. We trained four models individually with the same feature vector of word

[17] `https://github.com/pan-webis-de/bolonyai20`

n-grams and emotions, then we stacked the four models into a Logistic Regression ensemble as done in **PAN_2020**.

5.2.2. Training and Evaluation Measures

For our experiments on Arabic datasets **ArPFN_T** and **ArPFN_TRT**, we evaluated our models using F_1^+ score. We additionally report Precision, Recall, Macro-F_1 and Accuracy scores. Moreover, for the experiments on dataset **EN_PAN**, we evaluate our models using F_1^+ score and report Macro-F_1 score on the test folds.

To identify statistically significant results, we performed two-tailed paired t-test on F_1^+ score, using the scores over the 10 folds, with 95% confidence.

5.3. Classification of Users Prone to Spread Arabic Fake News (**RQ1**)

To address **RQ1**, we trained our baselines and individualized models to predict if a user is prone to spread fake news or not. We used datasets **ArPFN_T** and **ArPFN_TRT** for this task, where the textual features are extracted from each user's recent 100 tweets.

Tables 5.2 and 5.3 summarize our results for the baseline models on datasets **ArPFN_T** and, **ArPFN_TRT** respectively, where bold numbers indicate the best scores for each metric. From the tables, it can be noted that **PAN_2020** and **PAN_2020+** models have successfully outperformed the majority class baseline for both datasets. The performance of the majority class classifiers is expectedly 0, in terms of F_1^+, since the negative class represents the majority in both datasets. Moreover, **PAN_2020+** performed slightly better than **PAN_2020**. However, the improvement was not statistically significant.

To answer **RQ1.1**, the baselines **PAN_2020** and **PAN_2020+** are able to identify

users prone to spread fake news in dataset **ArPFN_TRT** (F_1^+ = 0.63). However, it is harder to identify users prone to spread fake news in dataset **ArPFN_T** (F_1^+ = 0.39).

Table 5.2: Performance on dataset **ArPFN_T**. Bold numbers indicate the best scores for each metric.

Model	F_1^+	Precision	Recall	Accuracy	Macro-F_1
Majority	0.00±0.00	0.00±0.00	0.00±0.00	0.79±0.00	0.44±0.00
PAN_2020	0.38±0.09	0.60±0.10	**0.29**±0.08	**0.81** ±0.02	**0.64**±0.05
PAN_2020+	**0.39**±0.12	**0.61**±0.10	0.28±0.11	**0.81**±0.02	**0.64**±0.06

Table 5.3: Performance on dataset **ArPFN_TRT**. Bold numbers indicate the best scores for each metric.

Model	F_1^+	Precision	Recall	Accuracy	Macro-F_1
Majority	0.00±0.00	0.00±0.00	0.00±0.00	0.65±0.00	0.40±0.00
PAN_2020	0.61±0.05	0.65±0.04	0.58±0.07	0.75±0.03	0.71±0.04
PAN_2020+	**0.63**±0.06	**0.69**±0.06	**0.59**±0.06	**0.76**±0.04	**0.73**±0.04

Next, we perform an ablation study to evaluate the impact of different feature category combinations and to find the best combination for this task. We report the results for four models; Random Forest (RF), XGboost Classifier (XGB), Logistic Regression (LR) and Feed Forward Neural Network (NN). We report results for the best performing models and omit the results for Support Vector Machines (SVM) as it performed poorly in our preliminary experiments when compared to the others. We tried the following combinations:

- Textual features only.
- Non-textual-based features (profile and statistical).
- Textual, profile, and statistical features.

- All feature categories.

First, we compare the performance between textual features and non-textual-based features. Tables 5.4 and 5.5 summarize the results for these experiments on datasets **ArPFN_T** and, **ArPFN_TRT** respectively. We performed significance tests to compare the performance of each combination with respect to **PAN_2020+**. We use the $*$ symbol to denote a statistically significant difference improvement over that baseline. In addition to comparing our results to the baseline, we perform a statistical significance test to compare the performance of each classifier when using textual features only in comparison with profile and statistical features. We use the † symbol to denote statistically-significant difference between each learning algorithm when trained using textual features only, in comparison to the same algorithm trained on profile and statistical features.

From these results, it can be noted that in most cases, training textual features on a single machine learning algorithm produces similar or better results to baseline **PAN_2020+**. For both datasets, it can be seen that some models perform better on textual features only, while others perform better when the other non-textual features are used. Table 5.4 shows that LR classifier that is trained on **ArPFN_T** and uses textual features can outperform the baseline and the classifiers trained on non-textual features with a statistically-significant difference.

Similarly, for dataset **ArPFN_TRT**, LR trained on textual features achieves higher performance in comparison to the baseline and other classifiers. However, no statistically-significant improvement was achieved. Moreover, we note that although the performance on F_1^+ of most classifiers outperforms the baseline, the baseline achieves higher precision for all classifiers in tables 5.4 and 5.5. We tested more feature combinations in

attempt to improve the classification.

Table 5.4: Evaluating the effect of using different feature category combinations on dataset **ArPFN_T**. Bold numbers indicate the best scores for each metric. The asterisk ($*$) indicates a statistically-significant difference compared to the baseline, while the dagger (†) indicates a statistically-significant difference between using textual features and non-textual features per learning algorithm.

Features	Model	F_1^+	Precision	Recall	Accuracy	macro-F_1
	PAN_2020+	0.39 ±0.12	**0.61** ±0.10	0.28 ±0.11	**0.81** ±0.02	0.64 ±0.06
Textual	RF	0.22* ±0.09	0.39 ±0.15	0.15 ±0.07	0.78 ±0.02	0.54 ±0.05
	XGB	0.33 ±0.06	0.51 ±0.08	0.24 ±0.06	0.79 ±0.02	0.61 ±0.03
	LR	**0.47**$^{*\ \dagger}$ ±0.06	0.58 ±0.12	0.41 ±0.06	**0.81** ±0.03	**0.68** ±0.04
	NN	0.44 ±0.07	0.52 ±0.10	0.40 ±0.08	0.80 ±0.03	0.66 ±0.04
Profile and Statistical	RF	0.31 ±0.08	0.54 ±0.08	0.22 ±0.07	0.80 ±0.02	0.60 ±0.04
	XGB	0.37 ±0.09	0.47 ±0.10	0.31 ±0.10	0.79 ±0.03	0.62 ±0.05
	LR	0.37 ±0.02	0.23 ±0.01	**0.90** ±0.09	0.36 ±0.02	0.36 ±0.02
	NN	0.40 ±0.08	0.37 ±0.09	0.43 ±0.10	0.73 ±0.04	0.61 ±0.06

Table 5.5: Evaluating the effect of using different feature category combinations on dataset **ArPFN_TRT**. Bold numbers indicate the best scores for each metric.

Features	Model	F_1^+	Precision	Recall	Accuracy	Macro-F_1
	PAN_2020+	0.63 ±0.06	0.69 ±0.06	0.59 ±0.06	**0.76** ±0.04	0.73 ±0.04
Textual	RF	0.64 ±0.05	0.64 ±0.05	0.64 ±0.07	0.72 ±0.04	0.75 ±0.03
	XGB	0.63 ±0.05	0.62 ±0.05	0.64 ±0.07	0.71 ±0.04	0.74 ±0.03
	LR	**0.65** ±0.05	0.64 ±0.06	0.66 ±0.06	0.73 ±0.04	0.75 ±0.04
	NN	0.63 ±0.05	0.63 ±0.07	0.65 ±0.10	0.71 ±0.04	0.74 ±0.04
Profile and Statistical	RF	**0.65** ±0.05	**0.66** ±0.05	0.63 ±0.06	0.73 ±0.04	**0.76** ±0.03
	XGB	0.63 ±0.05	0.61 ±0.04	0.65 ±0.05	0.71 ±0.03	0.73 ±0.03
	LR	0.60 ±0.03	0.46 ±0.02	**0.85** ±0.04	0.59 ±0.03	0.59 ±0.03
	NN	0.63 ±0.04	0.51 ±0.04	0.80 ±0.05	0.66 ±0.04	0.66 ±0.04

Next, we experiment with combining textual features with other feature categories to establish the best combination of feature categories. We combine textual features with profile and statistical features and combine all feature categories. Table 5.6 summarizes the results obtained on dataset **ArPFN_T**. It can be demonstrated that LR and NN have outperformed the baseline with statistically-significant improvement. Moreover, emotional features did not show an improvement in classifying users prone to spread fake news.

Table 5.7 summarizes the results obtained on dataset **ArPFN_TRT**. Interestingly, it can be observed that classifiers XGB, LR and NN have outperformed the baseline for this dataset with a statistically-significant difference. Moreover, using emotional

features yields in a better F_1^+ score for two of the classifiers, however, there is no statistical improvement when using emotional signals.

To answer **RQ1.2**, we conclude that combining textual and non-textual features yields better results in general. More specifically, the best achieved performance is obtained on LR classifier on dataset **ArPFN_T** (F_1^+ = 0.51) and XGB classifier (F_1^+ = 0.70) on dataset **ArPFN_TRT** trained on all feature categories.

Table 5.6: Evaluating the effect of using different feature category combinations on dataset **ArPFN_T**. Bold numbers indicate the best scores for each metric. The asterisk ($*$) indicates statistically-significant difference compared to the baseline.

Features	Model	F_1^+	Precision	Recall	Accuracy	Macro-F_1
	PAN_2020+	0.39 ±0.12	0.61 ±0.10	0.28 ±0.11	0.81 ±0.02	0.64 ±0.06
Textual, Profile, and Statistical	RF	0.20* ±0.08	**0.72** ±0.27	0.12 ±0.05	0.80 ±0.02	0.54 ±0.04
	XGB	0.37 ±0.09	0.57 ±0.12	0.28 ±0.08	0.81 ±0.02	0.63 ±0.05
	LR	**0.51*** ±0.06	0.51 ±0.08	**0.54** ±0.09	0.79 ±0.04	**0.69** ±0.03
	NN	0.44* ±0.07	0.55 ±0.09	0.39 ±0.08	0.80 ±0.03	0.67 ±0.04
Textual, Profile, Statistical, and emotions	RF	0.19* ±0.07	0.66 ±0.20	0.12 ±0.05	0.80 ±0.01	0.54 ±0.03
	XGB	0.38 ±0.10	0.62 ±0.10	0.29 ±0.11	**0.82** ±0.02	0.64 ±0.06
	LR	**0.51*** ±0.06	0.51 ±0.08	**0.54** ±0.09	0.80 ±0.04	**0.69** ±0.04
	NN	0.45* ±0.07	0.56 ±0.09	0.39 ±0.09	0.81 ±0.03	0.67 ±0.04

Table 5.7: Evaluating the effect of using different feature category combinations on dataset **ArPFN_TRT**. Bold numbers indicate the best scores for each metric. The asterisk ($*$) indicates statistically-significant difference compared to the baseline.

Features	Model	F_1^+	Precision	Recall	Accuracy	Macro-F_1
	PAN_2020+	0.63 ±0.06	0.69 ±0.06	0.59 ±0.06	0.76 ±0.04	0.73 ±0.04
Textual, Profile and Statistical	RF	0.66 ±0.06	0.67 ±0.06	0.66 ±0.08	0.74 ±0.04	0.76 ±0.04
	XGB	0.68* ±0.04	0.68 ±0.04	0.69 ±0.06	0.75 ±0.03	**0.78** ±0.03
	LR	0.68* ±0.04	0.65 ±0.04	0.70 ±0.06	0.75 ±0.03	0.77 ±0.03
	NN	0.67* ±0.05	**0.69** ±0.06	0.65 ±0.06	0.75 ±0.04	**0.78** ±0.04
Textual, Profile, Statistical, and Emotions	RF	0.67 ±0.05	0.68 ±0.07	0.67 ±0.06	0.75 ±0.04	0.77 ±0.04
	XGB	**0.70*** ±0.05	**0.69** ±0.06	0.70 ±0.07	**0.77** ±0.04	0.79 ±0.04
	LR	0.68* ±0.04	0.65 ±0.04	**0.71** ±0.05	0.74 ±0.03	0.76 ±0.03
	NN	0.64 ±0.06	0.66 ±0.06	0.62 ±0.08	0.73 ±0.04	0.75 ±0.04

Lastly, we investigate the performance of the classifiers when we use contextualized embeddings instead of word n-grams. We used the 768-dimensional embeddings vector that represents the average of the embeddings of each user's 100 tweets. Instead of using textual features, we concatenate the user embeddings vector to the profile, statistical, and emotional features.

Figures 5.1 and 5.2 show bar charts comparing the F_1^+ score of using *different* embeddings (i.e., generated from different pre-trained language models) in training our four models. The figures also illustrate the performance of the models trained with all feature categories when textual features are word n-grams. The baseline **PAN_2020+** performance was also added to the figure for the sake of comparison. Both figures show that

S-BERT embeddings yield the best performance among all other types of embedding. However, the models trained on embeddings are all outperformed by the models trained on the word n-grams. Our statistical significance tests have shown that the classifier that uses n-grams as textual features statistically outperformed most of the classifiers that use contextualized embeddings. Moreover, **PAN_2020+** has outperformed all RF classifiers, XGB classifiers that use AraBERT and MARBERT embeddings for **ArPFN_T** with a statistically-significant difference. While for dataset **ArPFN_TRT**, **PAN_2020+** only outperformed LR and NN classifiers that use MARBERT embeddings with a statistically-significant difference. Answering **RQ1.3**, the replacement is then deemed ineffective, at least in the way we generated the embeddings vector as the average of the embeddings vectors of each individual user's timeline tweets.

In summary, we conclude that we are able to classify Twitter users prone to spread Arabic fake news. We conducted experiments using only textual features and without using textual features. Second, we try different feature combinations. Our results demonstrate that textual features are essential for classifying users and that combining textual features with user profile features, statistical features and emotional signals yields the best results on our dataset. Moreover, we conclude that using contextualized embeddings instead of word n-grams does not improve the performance of the classifiers.

The results show that the classification performance on dataset **ArPFN_T** is less than the performance on dataset **ArPFN_TRT**. We speculate that this might be due to the fact that users prone to spread fake news are a minority in dataset **ArPFN_T**. The performance could be improved by increasing the number of users that are prone to spread fake news. We further investigate the effect of using users that have shared retweets of claims in Section 5.5.

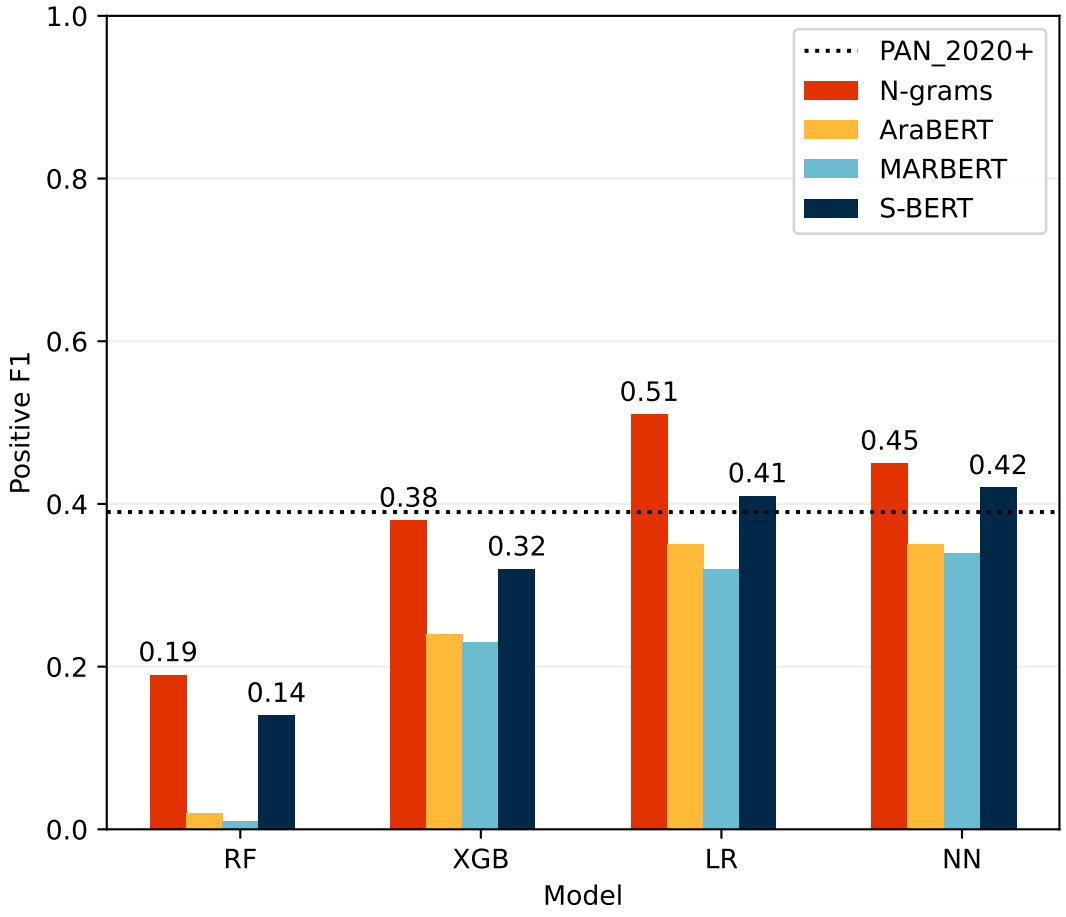


Figure 5.1: Performance of different models trained using mean-pooling BERT embeddings with profile, statistical and emotional features on dataset **ArPFN_T**.

5.4. Effect of Considering Longer User's Timeline (**RQ2**)

We explore the effect of using more tweets from the user's timeline on classifying the users. Identifying the ideal number of tweets is important in time-sensitive applications, as it determines the number of requests using Twitter API, which allows the retrieval of 100 tweets per request with a rate limit of 900 requests within a 15-minute window.[18] We conduct experiments by gradually increasing the number of tweets per user and

[18] https://developer.twitter.com/en/docs/twitter-api/rate-limits

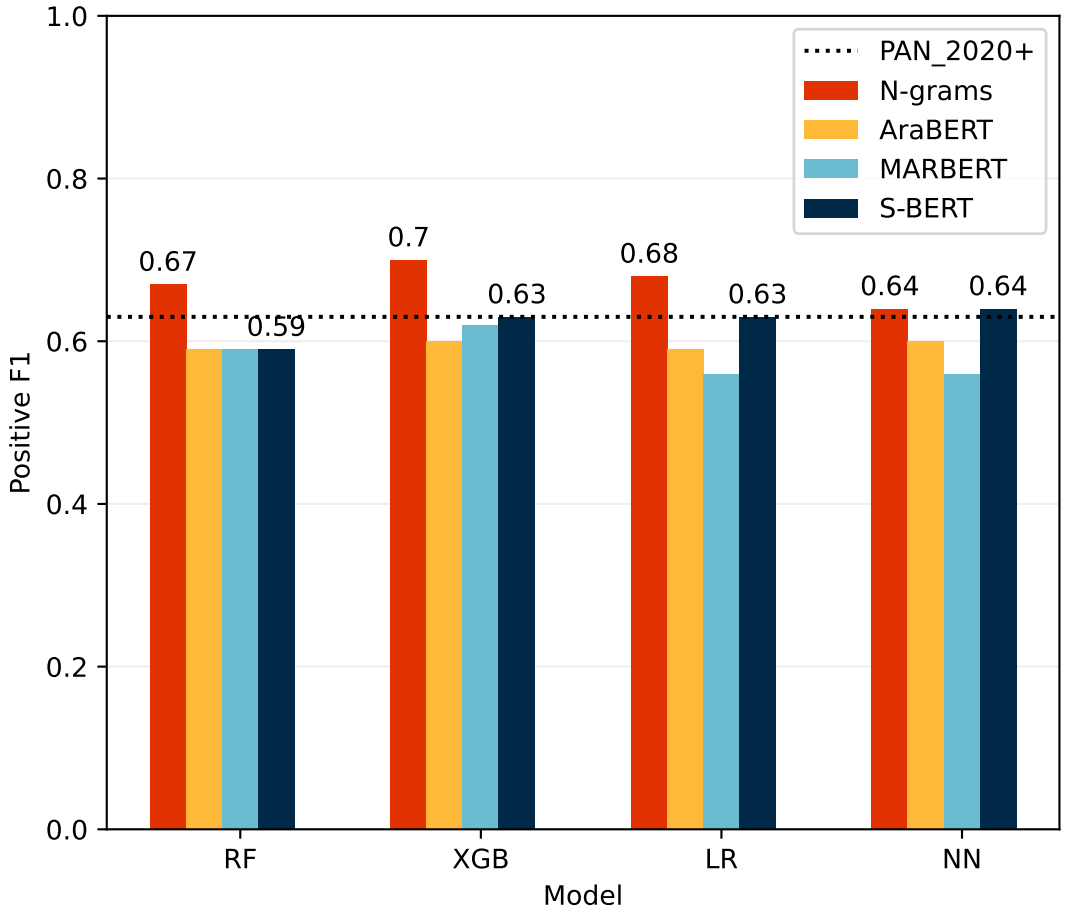


Figure 5.2: Performance of different models trained using mean-pooling BERT embeddings with profile, statistical and emotional features on dataset **ArPFN_TRT**.

evaluating the performance of each model. We test the model performance on 100, 500, 2000 and 3200 user tweets. The experiments are conducted on datasets **ArPFN_TRT** and **ArPFN_T** only, as dataset **EN_PAN** is only limited to 100 tweets and the usernames are hashed, so we were unable to expand it.

For these experiments, we chose the best models from tables 5.6 and 5.7. Namely, for dataset **ArPFN_TRT** we select XGB and LR classifiers trained on all features. For dataset **ArPFN_T** we select XGB and NN classifiers trained on all features.

Figures 5.4 and 5.3 show the performance after increasing the number of tweets

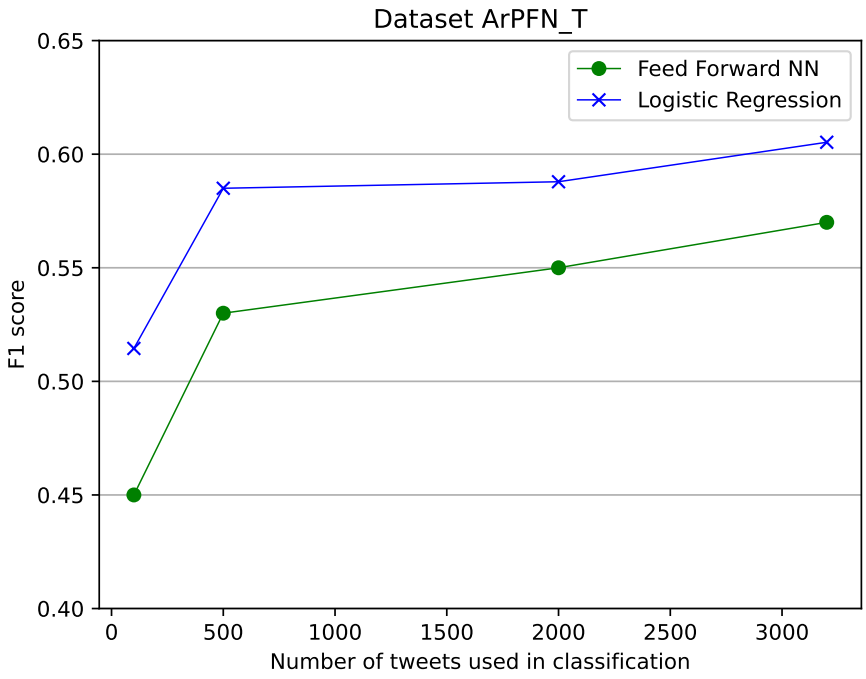


Figure 5.3: Performance on dataset **ArPFN_T** after gradually increasing the considered number of user tweets for classification.

for each model. The figures clearly show that increasing the number of considered tweets of the timeline results in performance improvements in all four models. The most notable improvement, which is also statistically-significant, was achieved by the LR model trained on **ArPFN_TRT** whose performance jumped from F_1^+ score 0.675 with 100 tweets to 0.732 with 3,200 tweets, yielding the highest performance in all our experiments. Similarly, in **ArPFN_T**, LR achieved an increase in F_1^+ from 0.514 with 100 tweets to 0.605 with 3,200 tweets.

Answering **RQ2**, considering more tweets in extracting the textual features yield better performance; however this requires more API requests which could introduce additional time delays during prediction.

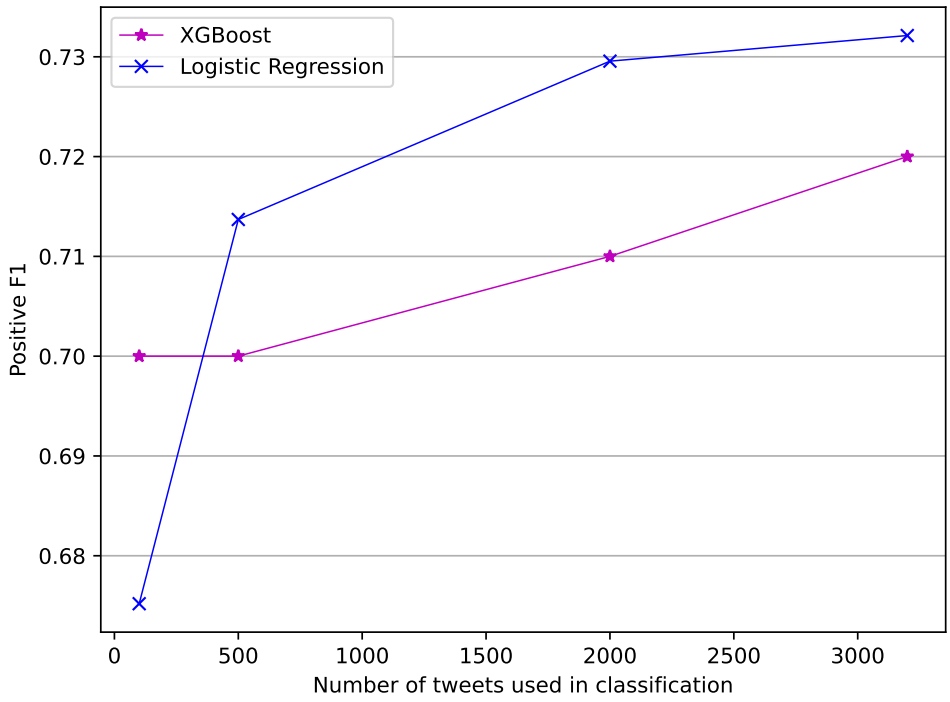


Figure 5.4: Performance on **ArPFN_TRT** after gradually increasing the number of considered user tweets for classification.

5.5. Effect of Considering Users Who RT Claims(**RQ3**)

In this section, we investigate the effect of using users from dataset **ArPFN_TRT** on training data on the performance of the classifiers. Datasets **ArPFN_T** and **ArPFN_TRT** have a large overlap. However, there is a notable difference in performance, as shown in Section 5.3. Figure 5.5 shows that there are 1,282 users in common between dataset **ArPFN_T** and **ArPFN_TRT**.[19] The difference between the two datasets is that dataset **ArPFN_TRT** considers the verified **retweets** of the users in addition to verified tweets. Therefore, our aim is to investigate whether there are any classification performance gains when we include users who retweeted claims in our training data.

[19]Out of the 1,282 users in common, 7 users that are labeled as not prone to spread fake news in **ArPFN_T** and labeled as prone to spread fake news in **ArPFN_TRT**. The mismatch in labeling is because all the users have one true tweet, 0 false tweets and more than one false retweet. We eliminate these users from our experiments.

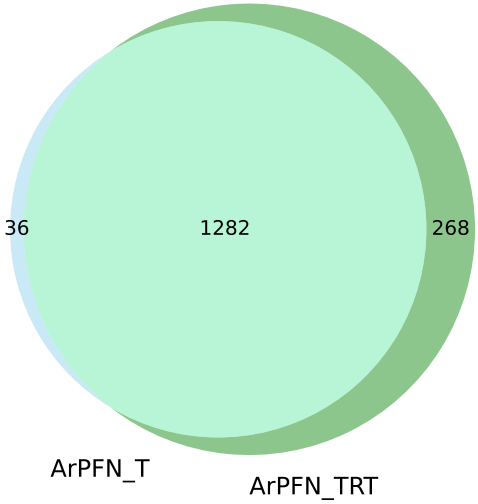


Figure 5.5: The overlap of users between datasets **ArPFN_T** and **ArPFN_TRT**.

We perform two experiments and test them on the same dataset. This allows us to investigate the effect of adding 268 users from dataset **ArPFN_TRT** to the training data and the effect of adding 36 users from dataset **ArPFN_T** to the training data. To do so, we perform 10-fold cross validation as follows. First, we use the 1,274 users that are in common in both datasets and have the same labels to split the data to 10 stratified folds. For testing, we use the same folds described above. While for training, we perform the following:

- For the first experiment, we add the 36 unique users in dataset **ArPFN_T** to the training folds.
- For the second experiment, we add the 268 unique users in dataset **ArPFN_TRT** to the training folds.

With the updated data splits, we perform our experiments to compare the performance

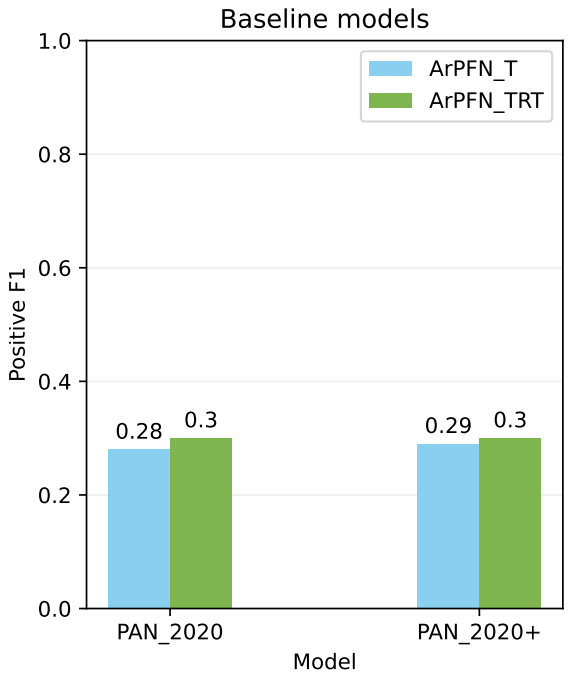


Figure 5.6: Comparison of average F_1^+ score between baseline models when trained on **ArPFN_T** versus when trained on **ArPFN_TRT**.

of both datasets. We report the average F_1^+ score of 10-fold cross validation. For textual features, we used user's recent 100 tweets only. Figure 5.6 shows bar charts comparing the F_1^+ score between the classifier trained on the baseline models using dataset **ArPFN_T** and dataset **ArPFN_TRT**. Figures 5.7, 5.8, 5.9 shows bar charts comparing the F_1^+ scores between the two datasets on different feature category combinations.

From the figures below, it can be clearly seen that performance of the classifier using dataset **ArPFN_TRT** is better in most cases. The addition of users with retweets to the training data yields in higher performance. We note that all 268 users added to datasets **ArPFN_TRT** training data are labeled as prone to spread fake news. In contrast, all 36 users added to dataset **ArPFN_T** are labeled as not prone to spread fake news. The increased performance achieved when the classifier is trained on **ArPFN_TRT** in comparision with **ArPFN_T** is most likely attributed to the fact that **ArPFN_TRT**

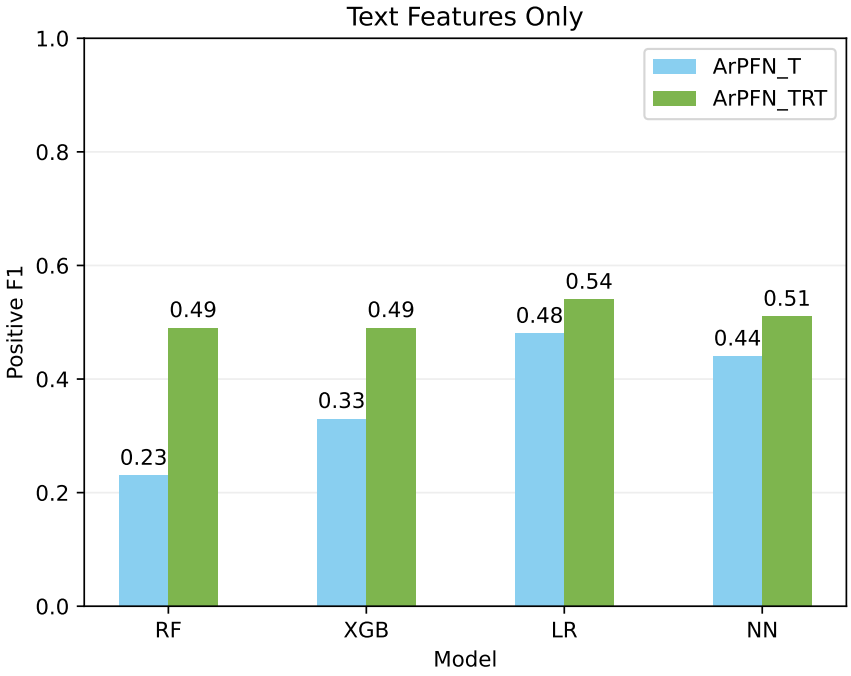


Figure 5.7: Comparison of average F_1^+ score between classifiers using word n-grams when trained on **ArPFN_T** versus when trained on **ArPFN_TRT**.

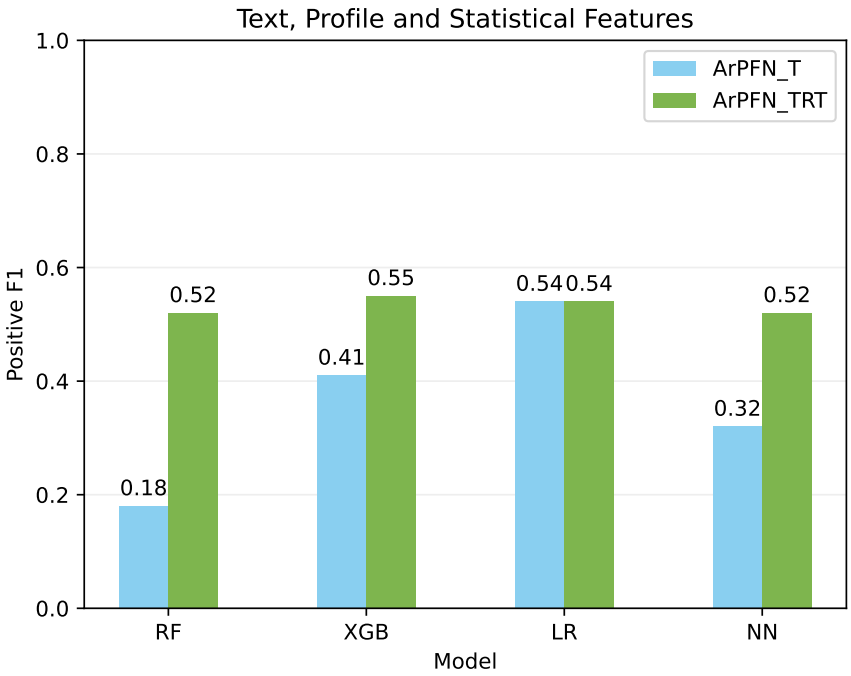


Figure 5.8: Comparison of average F_1^+ score between classifiers using word n-grams, profile and statistical features when trained on **ArPFN_T** versus when trained **ArPFN_TRT**.

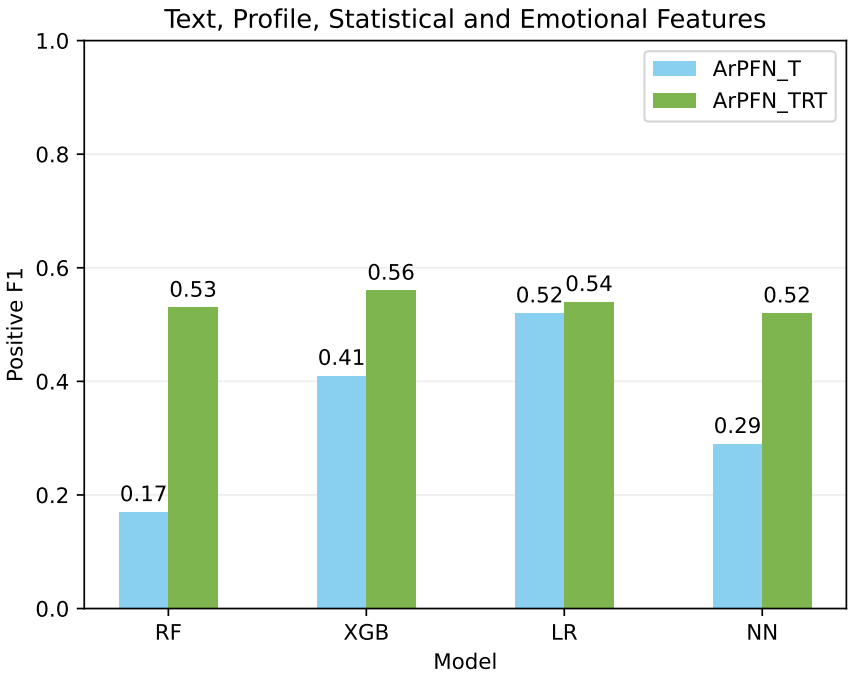


Figure 5.9: Comparison of average F_1^+ score between classifiers using all feature combinations when trained on **ArPFN_T** versus when trained on **ArPFN_TRT**.

contains more positive training data.

5.6. Applicability on an English Dataset (**RQ4**)

We aim to validate the effectiveness of our methodology by testing it on datasets of other languages. To this end, we used the **EN_PAN** dataset to conduct our experiments on English. Dataset **EN_PAN** is limited to the text of the recent 100 tweets from each user, and the usernames of the users were hashed to maintain their privacy. Thus, we were unable to extract all the features we described in Section 4.1. In this experiment, we compare the performance of the main baseline **PAN_2020** and the improved baseline **PAN_2020+**.

Table 5.8 reports the results of our experiment on the same PAN data splits. It is shown that our methodology of combining textual features with emotional signals has

improved the F_1^+ scores by 5 points. To validate our results, we also perform 10-fold cross validation. The results of that setup are summarized in table 5.9, showing that our improved baseline **PAN_2020+** outperforms **PAN_2020**, which constitute the current state-of-the-art. However, the improvement was not statistically-significant.

Table 5.8: Performance on **EN_PAN** using PAN train-test splits. Bold numbers indicate the best scores for each metric.

Model	F_1^+	Macro F_1^+
PAN_2020	0.74	0.73
PAN_2020+	**0.79**	**0.77**

To gain a better understanding, we report the performance of each sub-model in **PAN_2020** and **PAN_2020+** on the testing data in table 5.10 and using 10-fold cross validation on table 5.11. When comparing Tables 5.8 with 5.10 and also 5.9 with 5.11, we infer that individual classifiers trained on textual features alone or combined with emotional signals perform less than ensemble models **PAN_2020** and **PAN_2020+**. Unlike the performance on datasets **ArPFN_T** and **ArPFN_TRT**, on **EN_PAN**, the ensemble methods achieve better predictive results than individual models.

Table 5.9: Results for 10-fold cross validation performed on **EN_PAN**. Bold numbers indicate the best scores for each metric.

Model	F_1^+	Macro F_1
PAN_2020	0.73 $\pm$0.05	0.73$\pm$0.05
PAN_2020+	**0.75** $\pm$0.03	**0.75**$\pm$0.05

Table 5.10: Performance of the sub-models of the ensemble classifiers **PAN_2020** and **PAN_2020+** on the test dataset. Bold numbers indicate the best scores for each metric.

Model	Sub-model	Features	F_1^+	Macro F_1^+
PAN_2020	LR	Text	0.70	0.70
	SVM	Text	0.72	0.71
	RF	Text	0.72	**0.73**
	XGB	Text	0.69	0.70
	XGB	Statistical	0.56	0.61
PAN_2020+	LR	Text + emotions	0.71	0.70
	SVM	Text + emotions	0.72	0.72
	RF	Text + emotions	**0.73**	**0.73**
	XGB	Text + emotions	0.69	0.70

Table 5.11: Performance of the sub-models of the ensemble classifiers **PAN_2020** and **PAN_2020+** using 10-fold cross validation. Bold numbers indicate the best scores for each metric.

Model	Sub-model	Features	F_1^+	Macro F_1^+
PAN_2020	LR	Text	0.72±0.06	**0.73**±0.05
	SVM	Text	0.69±0.04	0.69±0.4
	RF	Text	0.72±0.05	**0.73**±0.05
	XGB	Text	0.67±0.06	0.67±0.05
	XGB	Statistical	0.61±0.05	0.61 ±0.03
PAN_2020+	LR	Text + emotions	0.73±0.06	**0.73**±0.05
	SVM	Text + emotions	0.70±0.04	0.70±0.04
	RF	Text + emotions	**0.74**±0.05	**0.73**±0.06
	XGB	Text + emotions	0.68±0.06	0.69±0.05

5.7. Further Analysis on the Classifier

In this section, we perform analysis on the predictions of our best classifier. We investigate the users that are prone to spread fake news, but have shared both True and False tweets. Our aim is to identify whether users that have shared both True and False tweets are harder to classify by our classifier than users that have only shared False tweets.

We also investigate the category of claims shared by the users. As shown in figure 3.2, most False tweets are related to politics and most True tweets are related to health. We want to examine if our classifier is biased towards users who tweet political tweets and labels them as prone to spread fake news, and whether our classifier is biased towards users who tweet health claims by labeling them as not prone to spread fake news.

For our analysis, we used the predictions of the Logistic Regression classifier trained on **ArPFN_TRT** using textual features (based on a user's 3,200 tweets), profile features, statistical features and emotional signals as it achieves the highest F_1^+ score (0.73).

First, we investigate users that shared both True and False tweets or retweets. In dataset **ArPFN_TRT**, 64 users out of 540 users prone to spread fake news have shared both True and False tweets. Figure 5.10 demonstrates the number of true positives and false positives from these users, in addition to the true positives and false positives for users that shared only False tweets or retweets. Note that these metrics were chosen since we are considering only examples of the positive class (i.e. users prone to spread fake news). The figure shows that our classifier correctly classified 41 users (64%) with both True and False tweets or retweets and 374 (78%) users with only False tweets or retweets. This indicates that detecting users that only shared False news is easier than detecting users that shared both True and False news. Future research could examine multi-class classification of users, where those that share both categories of tweets can be labeled differently.

Second, we investigate the categories of tweets shared by users, we limit our investigation to Health and Politics categories as they are the most dominant categories of tweets. Table 5.12 summarizes the percentage of false positive and false negative users that have shared political or health related verified tweets. From table 5.12, it can be

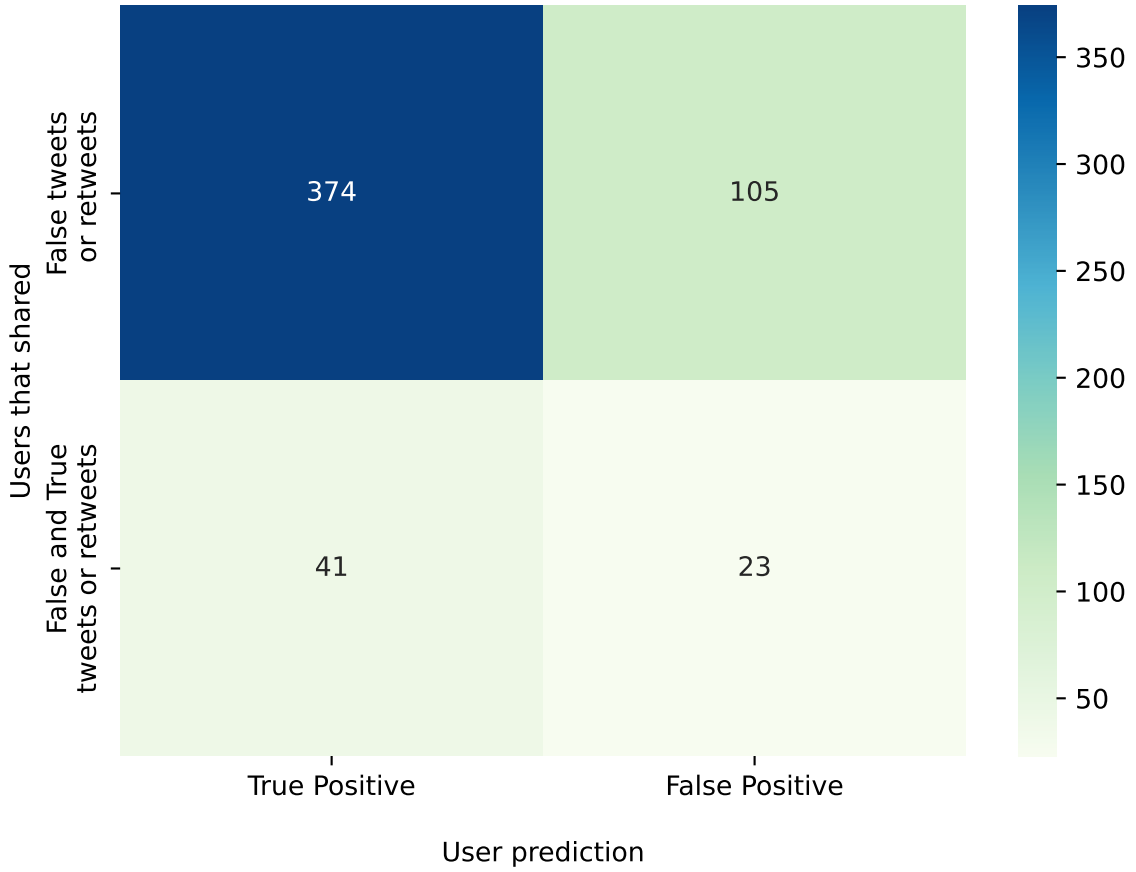


Figure 5.10: Heatmap displaying number of true positives and false positives and whether they have shared False claims only or shared both False and True claims.

noted that only 14% of false positive users have shared tweets related to politics. As for false negatives, 46% of the users sharing tweets related to health have been misclassified as not prone to spread fake news.

Table 5.12: Investigation of false positive and false negative users that shared verified tweet related to politics or health on dataset **ArPFN_TRT**.

Tweet Category	Percentage of False Positives	Percentage of False Negatives
Health	32%	46%
Politics	14%	20%

We conclude that our classifier is capable of identifying users prone to spread fake

news even if they have shared other true news. Moreover, our classifier is not biased toward political tweets. However, one concern is that the classifier considers 46% of users who spread health related tweets as not prone to spread fake news. This is expected as the majority of True Tweets are related to health. To address this limitation in future studies, the dataset should include more users and verified tweets from different categories. We note that similar conclusions where observed for dataset **ArPFN_T**. To avoid redundancy, we only show the results for dataset **ArPFN_TRT**.

CHAPTER 6: WEB API

We select the best model from dataset **ArPFN_TRT** and train it on the whole dataset. We selected XGB model trained on 100 tweets. Then we develop a function that accepts a Twitter username as an input, searches for user tweets using Twitter API, then it extracts features, performs the prediction and returns the user prediction as an output. We deploy the model as an API service. We used Flask micro-web framework[20] to implement the functionality as an API service. The API service allows our functionality to be accessible using any programming language.

Figure 6.1 demonstrates our functionality called using Postman API platform. The username is provided to the function as an input (userID) and the function responds with the confidence score of the prediction and the binary predicted label. The figure shows the prediction of the function for the provided username, the returned result states that the user is not prone to spread fake news with confidence of 0.74 within 2.64 seconds only.

[20] https://flask.palletsprojects.com/en/2.0.x/

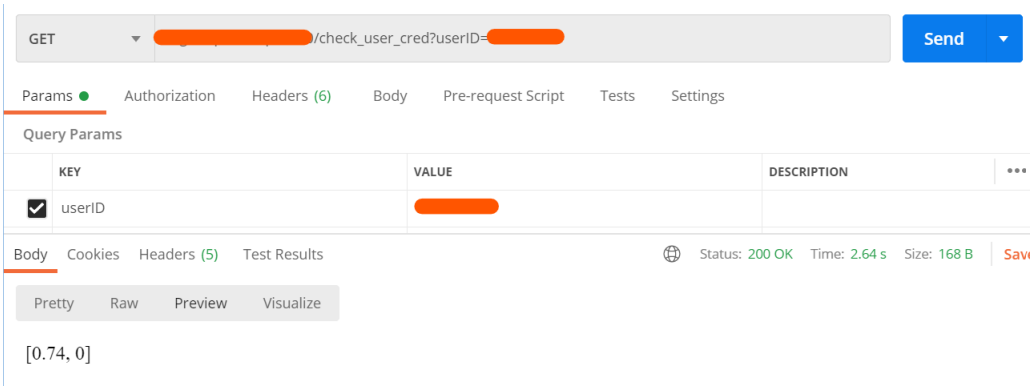


Figure 6.1: Testing of our web API service on Postman API Platform.

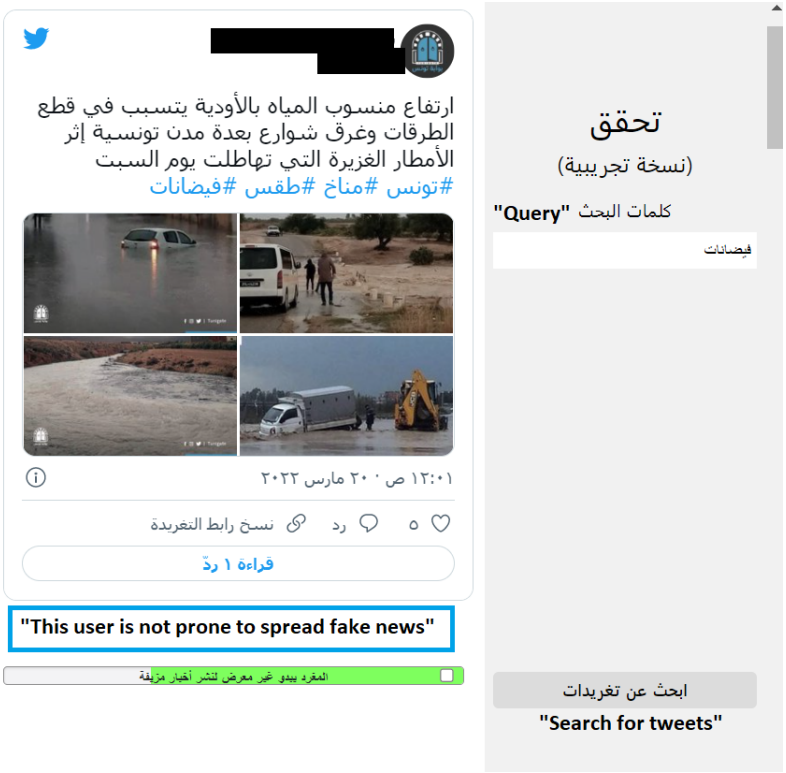


Figure 6.2: Integration of our API functionality in an online user interface.

Moreover, the API functionality can be integrated into an online user interface. Figure 6.2 demonstrates an implementation of our functionality in a real-time system. The query is searched and tweets are retrieved from Twitter in real-time, the bar below the tweet indicates if the user is prone to spread fake news or not and prediction confidence.

Note that we intend to make our functionality available with limited access to

journalists and authorities. We believe our model should be considered as a first step in identifying users prone to spread fake news where human judgement is assisted before characterizing the user.

CHAPTER 7: CONCLUSION AND RESEARCH CONTRIBUTIONS

7.1. Conclusion

In this thesis, we explored the task of identifying users who are prone to spread Arabic fake news on Twitter. While most related work on fake news detection systems focus on tweet verification, we instead explore the source of the tweet and take into account the user's recent activity, profile features, statistical features and emotional signals. We constructed the *first* Arabic users datasets, **ArPFN_T** and **ArPFN_TRT**, for this task by leveraging two Arabic misinformation datasets, ArCOV19-Rumors and AraFacts. We also proposed the *first* Arabic-specific classifier to identify users prone to spread fake news on Arabic Twitter. Our methodology exploits all user features and uses machine learning models to classify the user. Our experiments showed that combining all feature categories yields the best classification performance. Moreover, we established that increasing the number of considered user tweets increases detection accuracy. Finally, we investigated the abilities of our classifier by investigating the predictions of the best model. The best model has achieved an average F_1^+ score of 0.73 using 10-fold cross validation on our Arabic dataset. We also showed that our method is effective even on the publicly available English dataset and has outperformed the current state-of-the art by achieving an F_1^+ score of 0.79.

This study offers important insights on the subject of user credibility on Twitter, a topic that undoubtedly has ethical consequences. As a result, the use of any such prediction system to assess an individual's credibility must be done with caution. We would like to emphasize that the user labeling heuristic in this thesis was established by taking the opinion of multiple individuals rather than one. Ultimately, the choice of

heuristics to label users is subjective and may differ based on the use case of the target application.

7.2. Research contributions

The following papers are contributions made by this thesis:

1. **Z. Sheikh Ali**, A. Al-Ali, and T. Elsayed, "Detecting Users Prone to Spread Fake News on Arabic Twitter," in Proceedings of The 5th Workshop on Open-Source Arabic Corpora and Processing Tools, European Language Resources Association, 2022.[48]

2. **Z. Sheikh Ali**, W. Mansour, T. Elsayed, and A. Al-Ali, "Arafacts: The First Large Arabic Dataset of Naturally-Occurring Professionally-Verified Claims," in Proceedings of the Sixth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, 2021. [31]

REFERENCES

[1] D. Radcliffe and H. Abuhmaid, "How the middle east used social media in 2020," *Available at SSRN 3826011*, 2021.

[2] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[3] M. O. Jones, "The gulf information war| propaganda, fake news, and fake trends: The weaponization of twitter bots in the gulf crisis," *International journal of communication*, vol. 13, p. 27, 2019.

[4] M. O. Jones, "Disinformation superspreaders: The weaponisation of covid-19 fake news in the persian gulf and beyond," *Global Discourse: An interdisciplinary journal of current affairs*, vol. 10, no. 4, pp. 431–437, 2020.

[5] R. El Ballouli, W. El-Hajj, A. Ghandour, S. Elbassuoni, H. M. Hajj, and K. B. Shaban, "Cat: Credibility analysis of arabic content on twitter.," in *WANLP@ EACL*, 2017, pp. 62–71.

[6] P. Nakov, G. Da San Martino, T. Elsayed, *et al.*, "Overview of the clef–2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, 2021, pp. 264–291.

[7] F. Harrag and M. K. Djahli, "Arabic fake news detection: A fact checking based deep learning approach," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 4, pp. 1–34, 2022.

[8] F. Haouari, M. Hasanain, R. Suwaileh, and T. Elsayed, "ArCOV19-Rumors: Arabic COVID-19 Twitter Dataset for Misinformation Detection," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Association for Computational Linguistics, 2021.

[9] K. Shu, A. Bhattacharjee, F. Alatawi, *et al.*, "Combating disinformation in a social media age," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 6, e1385, 2020.

[10] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature communications*, vol. 9, no. 1, pp. 1–9, 2018.

[11] G. Pennycook, Z. Epstein, M. Mosleh, A. A. Arechar, D. Eckles, and D. G. Rand, "Shifting attention to accuracy can reduce misinformation online," *Nature*, vol. 592, no. 7855, pp. 590–595, 2021.

[12] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1096–1103.

[13] T. Mihaylov, G. Georgiev, and P. Nakov, "Finding opinion manipulation trolls in news community forums," in *Proceedings of the nineteenth conference on computational natural language learning*, 2015, pp. 310–314.

[14] T. Mihaylov, I. Koychev, G. Georgiev, and P. Nakov, "Exposing paid opinion manipulation trolls," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, 2015, pp. 443–450.

[15] S. K. Maity, A. Chakraborty, P. Goyal, and A. Mukherjee, "Detection of sockpuppets in social media," in *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 243–246.

[16] C. Chen, K. Wu, V. Srinivasan, and X. Zhang, "Battling the internet water army: Detection of hidden paid posters," in *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, IEEE, 2013, pp. 116–120.

[17] J. An, H. Kwak, C. S. Lee, B. Jun, and Y.-Y. Ahn, "Predicting anti-asian hateful users on twitter during covid-19," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 4655–4666.

[18] K. Darwish, D. Alexandrov, P. Nakov, and Y. Mejova, "Seminar users in the arabic twitter sphere," in *International Conference on Social Informatics*, Springer, 2017, pp. 91–108.

[19] V. S. Subrahmanian, A. Azaria, S. Durst, *et al.*, "The darpa twitter bot challenge," *Computer*, vol. 49, no. 6, pp. 38–46, 2016.

[20] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016.

[21] A. Deb, L. Luceri, A. Badaway, and E. Ferrara, "Perils and challenges of social media and election manipulation analysis: The 2018 us midterms," in *Companion proceedings of the 2019 world wide web conference*, 2019, pp. 237–247.

[22] F. Rangel, A. Giachanou, B. Ghanem, and P. Rosso, "Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter," in *CLEF*, 2020.

[23] J. Buda and F. Bolonyai, "An ensemble model using n-grams and statistical features to identify fake news spreaders on twitter," in *CLEF*, 2020.

[24] A. Giachanou, B. Ghanem, E. A. Rıssola, P. Rosso, F. Crestani, and D. Oberski, "The impact of psycholinguistic patterns in discriminating between fake news spreaders and fact checkers," *Data & Knowledge Engineering*, vol. 138, p. 101 960, 2022.

[25] A. Giachanou, E. A. Rıssola, B. Ghanem, F. Crestani, and P. Rosso, "The role of personality and linguistic patterns in discriminating between fake news spreaders and fact checkers," in *International Conference on Applications of Natural Language to Information Systems*, Springer, 2020, pp. 181–192.

[26] B. Rath, A. Salecha, and J. Srivastava, "Detecting fake news spreaders in social networks using inductive representation learning," *arXiv preprint arXiv:2011.10817*, 2020.

[27] M. Del Tredici and R. Fernández, "Words are the window to the soul: Language-based user representations for fake news detection," *arXiv preprint arXiv:2011.07389*, 2020.

[28] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu, "The role of user profiles for fake news detection," in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 436–439.

[29] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," *Big Data*, vol. 8, no. 3, pp. 171–188, 2020.

[30] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: A system to evaluate social bots," in *Proceedings of the 25th international conference companion on world wide web*, 2016, pp. 273–274.

[31] Z. Sheikh Ali, W. Mansour, T. Elsayed, and A. Al-Ali, "Arafacts: The First Large Arabic Dataset of Naturally-Occurring Professionally-Verified Claims," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Association for Computational Linguistics, 2021.

[32] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 675–684.

[33] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[34] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020, pp. 9–15.

[35] I. Zeroual, D. Goldhahn, T. Eckart, and A. Lakhouaja, "Osian: Open source international arabic news corpus-preparation and integration into the clarin-infrastructure," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, 2019, pp. 175–182.

[36] I. A. El-Khair, "1.5 billion words arabic corpus," *arXiv preprint arXiv:1611.04033*, 2016.

[37] M. Abdul-Mageed, A. Elmadany, *et al.*, "Arbert & marbert: Deep bidirectional transformers for arabic," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 7088–7105.

[38] V. Lampos, N. Aletras, D. Preoţiuc-Pietro, and T. Cohn, "Predicting and characterising user impact on twitter," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 405–413.

[39] B. Ghanem, P. Rosso, and F. Rangel, "An emotional analysis of false information in social media and news articles," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, no. 2, pp. 1–18, 2020.

[40] A. Giachanou, P. Rosso, and F. Crestani, "Leveraging emotional signals for credibility detection," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 877–880.

[41] X. Zhang, J. Cao, X. Li, Q. Sheng, L. Zhong, and K. Shu, "Mining dual emotion for fake news detection," in *Proceedings of the Web Conference 2021*, 2021, pp. 3465–3476.

[42] E. Fersini, J. Armanini, and M. D'Intorni, "Profiling fake news spreaders: Stylometry, personality, emotions and embeddings.," in *CLEF (Working Notes)*, 2020.

[43] L. G. Moreno-Sandoval, E. A. P. Del Puertas, A. P. Quimbaya, and J. A. Alvarado-Valencia, "Assembly of polarity, emotion and user statistics for detection of fake profiles.," in *CLEF (Working Notes)*, 2020.

[44] S. Hassan, H. Mubarak, A. Abdelali, and K. Darwish, "Asad: Arabic social media analytics and understanding," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2021, pp. 113–118.

[45] Q. Li, H. Peng, J. Li, *et al.*, "A survey on text classification: From traditional to deep learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 2, pp. 1–41, 2022.

[46] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, 2022.

[47] M. Z. Islam, J. Liu, J. Li, L. Liu, and W. Kang, "A semantics aware random forest for text classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1061–1070.

[48] Z. Sheikh Ali, A. Al-Ali, and T. Elsayed, "Detecting Users Prone to Spread Fake News on Arabic Twitter," in *Proceedings of The 5th Workshop on Open-Source Arabic Corpora and Processing Tools*, European Language Resources Association, 2022.

APPENDIX A: TWEET ANNOTATION GUIDELINES

The annotation guidelines that we constructed to annotate AraFacts tweets are described below:

Given a claim X, label the tweet T as:

- **Expressing the same claim**: if the author of the tweet is sharing, restating or rephrasing the same claim X. In other words, the author is believing the claim and participating in sharing it. (i.e., T = X).
- **Negating the same claim**: if the author of the tweet is disagreeing or denying the claim. In other words, the author is debunking the claim and stating that it is incorrect. (i.e., C = not X).
- **Other**: if it is not one of the above, for example:
  - Author of the tweet is sharing the claim and questioning whether it is true or fake
  - Author of the tweet is sharing multiple claims including the main claim
  - The tweet is referring to a deleted image or video and the text of the tweet is insufficient to annotate the claim

Annotation steps:

1. Read claim text
2. Read the tweet text
3. Determine if tweet is expressing the claim, negating the claim or neither

Notes:

- If the claim is related to an image or video, we recommend checking the URL of the claim and the URL of the tweet to compare if both links refer to the same image or video.

- We recommend considering the claim publication date and tweet posting date into considerations. If the tweet is posted after the claim has been verified, make sure that the tweet is still relevant to the same claim and that the claim is still holding the same label when it was verified.