Automatic Fake News Detection in Political Platforms – A Transformer-based Approach

Shaina Raza
Department of Computer Science, Ryerson University
shaina.raza@ryerson.ca

Abstract

The dynamics and influence of fake news on Twitter during the 2020 US presidential election remains to be clarified. Here, we use a dataset related to 2020 U.S Election that consists of news articles and tweets on those articles. Therefore, it is extremely important to stop the spread of fake news before it reaches a mass level, which is a big challenge. We propose a novel fake news detection framework that can address this challenge. Our proposed framework exploits the information from news articles and social contexts to detect fake news. The proposed model is based on a Transformer architecture, which can learn useful representations from fake news data and predicts the probability of a news as being fake or real. Experimental results on real-world data show that our model can detect fake news with higher accuracy and much earlier, compared to the baselines.

1 Introduction

Fake news refers to false or misleading information that appears as real news (Zhou & Zafarani, 2020). Fake news can be broadly categorized as either misinformation (unintentional false information) or disinformation (deliberate false information). Recent social and political events, such as 2020 United States presidential election, have seen an increase in fake news (E. Chen et al., 2021). According to a report by First Draft News [1], America's current disinformation crisis is the result of more than two decades of corruption in country's information ecosystem. There are many factors to blame for this social and political misinformation. For example, the role of social media that is unregulated, lack of investment in public media, downfall of local news outlets, and emergence of hyper-partisan online outlets.

An information (news) ecosystem consists of publishers (news media that publish the news article), information (news content) and users (Anderson, 2016). Initially, the news comes from the publishers. Then, it goes to the news websites, from where it goes to the users who share news on different platforms (blogs, social media, etc.). If the news is fake, some users may find it more sensational and interesting to comment on and share over their networks. The existence of the bots in social media makes it even worse, who spread misinformation through multiple channels to urge people believe the fake news. Therefore, it becomes crucial to stop the fake news before it reaches to a broad audience. In this paper, we aim to effectively detect the fake news.

Generally, the content of fake news is vague and misleading (C. Liu et al., 2019). According to a research (Horne & Adalı, 2017), the content of fake news consists of certain patterns, such as excessive use of capital letters, punctuations, or emotion-bearing words, which gives us clues about a news being fake or real. However, if the content of news is not sufficient, then the social contexts may be useful to assess the veracity (truthfulness) of news. The social contexts (Shu et al., 2019) refers to users' interactions, such as, comments, shares, likes, followers-followees relations etc., that are helpful to determine if a news fake or real. Sometimes, even the verified accounts in social media are involved in the propagation of fake news

[1] https://firstdraftnews.org/latest/fake-news-complicated/

Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) , pages 68–78
August 5–6, 2021, ©2021 Association for Computational Linguistics

(Shahi et al., 2021). In this work, we plan to consider both news content and social contexts to detect fake news.

Generally, a news item is represented by a news ID or news title, which is not sufficient to capture the patterns of fake news. There are many important pieces of information that may be more useful. For example, a news body or news source could be (at times) more convincing in persuading readers to believe something, so, we need to pay closer attention to such information. We refer to such auxiliary information as side (metadata) information. The side information associated with a news article can be news body, source, time of publication, topics etc. In this work, we plan to consider different side information related to news. We also consider embedded tweets on news articles, which provide us additional information to determine the veracity of news.

According to a research, the fake news spreads within minutes once planted (Vosoughi et al., 2018). For example, the fake news that Elon Musk's Tesla team is inviting people to give any amount ranging from 0.1 to 20 bitcoins in exchange for double the amount, resulted in a loss of millions of dollars within the first few minutes[2]. So, it is critical to detect fake news early on before it spreads. In this work, we plan to early detect the fake news within few minutes after its propagation.

In recent years, the Transformer-based models (Vaswani et al., 2017) have gained significant popularity in different NLP tasks, such as text classification, detection methods. These models usually input whole lexical data as one piece of information or document, without considering any side information (Wu et al., 2020). In addition, the temporal information is not considered (by default) in these models. To better utilize the strengths of Transformer-based models for fake news detection, it is important to include heterogenous information (main, side and temporal information) to build a classification model. In this work, we build a novel Transformer model that considers heterogenous information for the task of fake news detection. Throughout this paper, we refer to the main information as the news headline, and we refer to side information as consisting of news-related features, social contexts (tweets), and temporal information.

We summarize our contributions as:

- We propose a novel Transformer model that considers news content and associated side information for the fake news detection task.
- We incorporate heterogenous side information in our model. In addition to only lexical data (as in typical Transformers), we also consider the non-lexical (numeric, categorical) data. We use the multi-head attention mechanism to attend to different parts of such information.
- We propose to detect fake new early within few minutes after it is planted. For that, we utilize the position encoding (Devlin et al., 2018) in the Transformer model that helps us to achieve our goal of early detection. The position encoding represents the words' order in a model, i.e., the value of a word (content) and its temporal position in a sentence.

We evaluate our system by running experiments on real-world data, which consists of news articles from various sources and social contexts from Twitter. Using an ablation study, we find that including both news content and social contexts is beneficial in detecting fake news patterns. The inclusion of more side information proves very useful as indicated in the results. We also show that our proposed model can detect fake news earlier and with greater accuracy than baselines.

The rest of the paper is organized as follows. Section 2 is about Methodology. Section 3 is for Experiment, and Section 4 is for Experimental Results and Analysis. The Related Work is covered in Section 5. Section 6 is the Conclusion and recommendations for the future work.

2 Methodology

Problem Definition: Given news and associated side information (news-related, social contexts and temporal information), the task is to determine if a news item is fake or real.

We consider the fake news detection task as a binary classification problem (news as fake or real). We also consider a multiclass classification (news as fake, real or mixed) in the experiments.

Proposed Model: In our work, we modify the structure of pre-trained Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) to add side information (in addition to main information). The same methodology can also be applied on other Transformers (RobertA, XLNet, BART, T5 etc.).

[2] https://www.bbc.com/news/technology-56402378

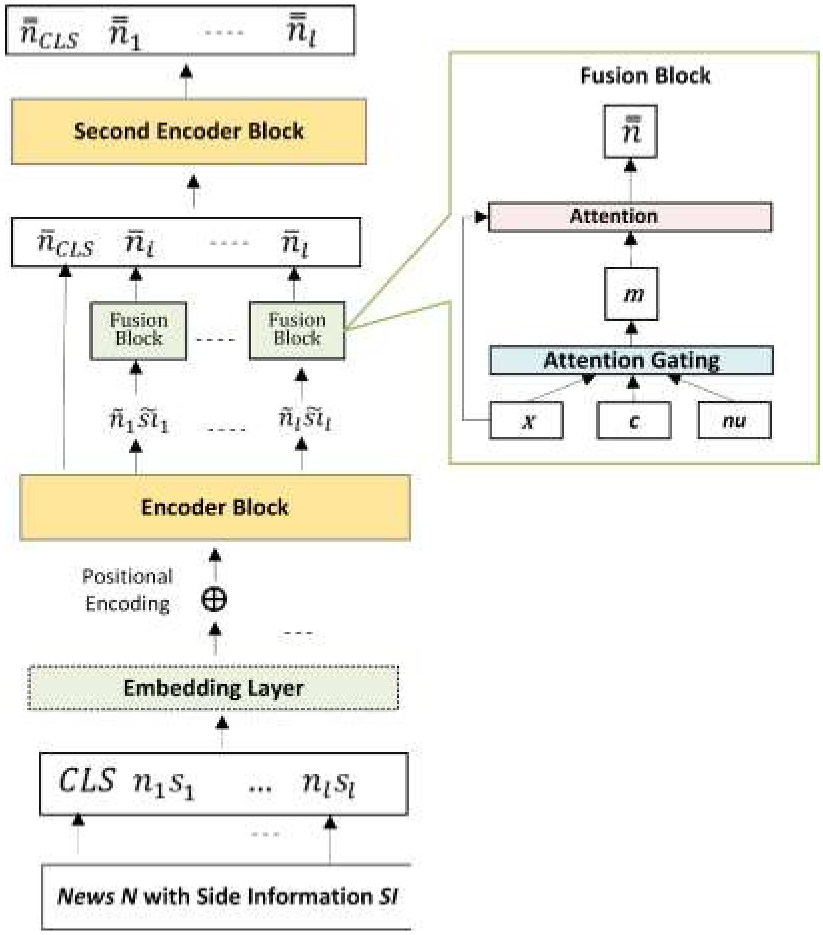
$\bar{\bar{n}}_{CLS}$ $\bar{\bar{n}}_1$ ---- $\bar{\bar{n}}_l$
Second Encoder Block
$\bar{n}_{CLS}$ $\bar{n}_i$ ---- $\bar{n}_l$
Fusion Block
Fusion Block
$\tilde{n}_1\tilde{si}_1$ ---- $\tilde{n}_l\tilde{si}_l$
Encoder Block
Positional Encoding
Embedding Layer
CLS n_1s_1 ... n_ls_l
News N with Side Information SI
Fusion Block
$\bar{\bar{n}}$
Attention
m
Attention Gating
x
c
nu

Figure 1: Proposed Model Faker

We represent each news item *N* by its title (main information) and side information. (Temporal, news-related, and social contexts). We believe that having more information is always beneficial. For instance, the author and source provide us with partisan information (political party as belonging to right or left wing). The temporal information is useful for determining whether a fake news is already spread or just released. Similarly, social contexts (tweets) give us additional information about users' reactions on news.

We present a novel Transformer-based model, **Faker**, as shown in Figure 1. The input to model is news items and associated side information. Each news item *N* has a sequence of words, i.e., $N = \{n_1, n_2, \dots, n_l\}$ where l is length. For each news, we have the accompanying side information, i.e., $SI = \{si_1, si_2, \dots, si_l\}$. In our work, we consider different types (lexical and non-lexical) of side information, whereas our main information is textual. We use 'word' as a general term to represent any word from *N* or feature from *SI.*

The first layer in Faker is the embedding layer. The input to the embedding layer is the sequence of words from each input *N* or *SI.* The [CLS] token is added at the start of the sequence and is later used for the class label prediction. We utilize the token and segment embedding from the BERT model to represent the syntax and semantics of each word.

Similar to (Q. Chen et al., 2019), we also assume that temporal order exists in sequences. So, we use position encoding (Vaswani et al., 2017) to capture the chronological information in the sequences. In our case, the position value of each word is decided by the timestamp of news publication.

The output from the embedding layer is then fed into the next twelve layers in the first Encoder block. After the encoding process, we get the output vector for each word from news. The contextualized representation after the first Encoder block is $\tilde{N} = \{\tilde{n}_1, \tilde{n}_2 \ \dots, \tilde{n}_l\}$ for the news and $\widetilde{SI} = \{\tilde{si}_1, \tilde{si}_2, \dots, \tilde{si}_l\}$ for the side information ($\tilde{si}$ comes from si, the dot above i under $\tilde{si}$ is hidden under the tilde ~). Each word vector from $\widetilde{N}$ and $\widetilde{SI}$ is then passed to a **F**usion **B**lock.

Fusion Block: Inside the Fusion Block, we represent each piece of information (lexical or non-lexical) from $\widetilde{N}$ and $\widetilde{SI}$ with a token (word). The *x* is a textual word, *nu* is numeric word (feature) and *c* is a categorical word.

Inspired by the gating mechanism introduced in (Wang et al., 2018), we first take each feature from the non-lexical data (*nu* and *c*) and combine them using a gating mechanism to produce a new non-lexical vector h, as shown in Equation (1):

$$h = g_c \odot (W_c c) + g_{nu} \odot (W_{nu} nu) + b_h \quad (1)$$

where c is categorical feature, *nu* is numerical feature, W denotes a weight matrix, b denotes a scalar bias, and g_c and g_{nu} are the gating vector for *c* and *nu* respectively. We may refer to g_i as a gating vector for a non-lexical feature *i.* The g_i is fused with *x* using an activation function *R*. Then it goes into *h*. The g_i is defined in Equation (2):

$$g_i = R\left(W_{g_i}[i \,||\, x] + b_i\right) \quad (2)$$

Once, we get the *h,* we use a weighted summation between the lexical vector *x* and the combined non-lexical vector h to produce a fused sequence *m*, as shown in Equation (3):

$$m = x + \alpha h \quad (3)$$

where *x* is text feature, α is a normalizing factor to dampen the magnitude of *h* representation within a range. The α is shown in Equation (4):

$$\alpha = \min\left(\frac{||x||_2}{||h||_2} * \beta, 1\right) \quad (4)$$

where the $||x||_2$ and $||h||_2$ denote the l_2-norms of x and h, and hyperparameter β is selected during the validation process. Subsequently, an attention is applied over the lexical and non-lexical vectors to produce the final fused representation$\bar{n}$. The output from each Fusion Block is $\bar{n}_i$ and is calculated for each word from the input sequence. The new sequence $\bar{N} = \{\bar{n}_{CLS}, \bar{n}_1, \bar{n}_2, \dots, \bar{n}_l\}$ is then fed as input to the next Encoder block. We apply the Encoder layers of our model on this sequence $\bar{n}$. At the end of the second Encoder block, we get the sequence $\bar{\bar{n}} = \{\bar{\bar{n}}_{CLS}, \bar{\bar{n}}_1, \bar{\bar{n}}_2, \dots, \bar{\bar{n}}_l\}$. The first element in $\bar{\bar{n}}$ is the [CLS] token that has the necessary information to predict the class {real, fake} label. Therefore, the $\bar{\bar{n}}_{CLS}$ goes through a final transformation to produce a value which can be used to predict a class label.

Feature	Description	Format
Article ID*	Article identifier	Integer
News title	Headline of news	Text
News source *	News Source (e.g., CNN, theonion)	Categorical
News content *	News Body	Text
Author *	Author of article	Categorical
URL *	URL of the article	Text
Publication timestamp*	Publication time as unix timestamp	Integer
Tweet ID *	ID of tweet	Integer
Embedded tweet*	Raw data from tweets (on news)	Text

Table 1: Dataset features, * is side information

3 Experiment

Fake news data: We use the NELA-GT-2020 dataset (Horne, Benjamin; Gruppi, 2021), which covers a broad set of events, including the COVID-19 pandemic and the 2020 U.S. Presidential Election. In this work, we only consider the 2020 U.S. Election event-based data, which consists of 294,504 related news articles across 403 sources between January 1st, 2020 and December 31st, 2020. The source-level ground truth labels are collected from the Media Bias/Fact Check (MBFC)[3] website.

The dataset also includes over 400,000 embedded Tweets found in news articles, which we also employ in our research. Table 1 shows the features of US Elections data that we use.

We use article IDs to create sequences based on available features (in Table 1). The embedded tweet text is also included in the sequence. Each sequence record is grouped by article ID and is sorted according to publication timestamp. The actual news articles are not labeled.

The dataset only provides us the ground truth labels (0- reliable, 1- mixed, 2- unreliable) at source-level. These source-level labels are obtained from MBFC, which considers the dimensions of veracity based on a factuality (credibility) and on conspiracy sources. We use the distant supervision (Mintz et al., 2009) to assign a label to each news story. In that, first we take the distant (weak) labels provided to each news source and use a weighted scheme to label each news article. The intuition of distant labeling is that the training labels at source-level may be imprecise and partial but can be used to create a strong

[3] https://mediabiasfactcheck.com/

predictive model. This approach is also suggested in the NELA-GT-18 paper (Nørregaard et al., 2019) and has shown promising results in a recent work (Horne et al., 2019).

After doing the labeling, we get around 37k labels as 'fake', 12.5k labels as 'real' and 32k labels as 'mixed'. To handle the data imbalance problem in in the dataset, we use the under-sampling technique (Drummond et al., 2003), in which the majority class is made closer to the minority class, by removing records from the majority class. Initially, we tried the SMOTE technique, in which the distribution of minority class is increased by replicating some records, but due to limited memory, we opt for under-sampling.

Evaluation Metrics: To assess model perform, we use accuracy **ACC**, precision **Prec**, recall **Rec** and F1-score **F1**, and area under curve **AUC**. Compared to ACC, AUC is usually better at ranking predictions because AUC evaluates model performance across all possible thresholds. We treat the fake news detection as a binary classification problem using labels {'Real', 'Fake'}, and as multiclass classification using labels {'Real', Fake' and 'Mixed'}.

Comparison Methods: For the baselines, we use:

Fake-news detection methods

- TriFN (Shu et al., 2019): A matrix factorization methods that exploits user, news and publisher relationships for fake news detection.
- Declare (Popat et al., 2020): A neural network that assesses the credibility of claims on news.
- Grover (Zellers et al., 2019): a neural framework to detect fake news.

Transformer-based methods

- BERT (Devlin et al., 2018): Bidirectional Encoder Representations from Transformers.
- GPT-2 (Radford et al., 2019): Generative Pre-trained Transformer model.
- VGCN-BERT (Lu et al., 2020): Transformer-based model that uses BERT with Graph Convolutional Network for text classification

Other methods

- SVM (Chang & Lin, 2011): Support Vector Machine model for text classification.
- DeepWalk (Perozzi et al., 2014): Embedding-based deep neural model for text classification.

Experimental Settings and Hyperparameters: All the experiments are conducted on the GPUs provided by Google Colab Pro. We implemented our model using TensorFlow. The sequences are created in a chronological order of a news publication timestamp. We temporally split the time-ordered data (by timestamps) for model training. We use the last 15% of the chronologically sorted data as the test set, the second to last 15% of the data as the validation set and the initial 75% of the data as the train set. The known labels are used as the ground truth for model training and evaluation.

In the final settings, we choose the following hyperparameters: the news stories and tweets are on average 500 words, so we choose a sequence length of 500 token. We use padding if the length is shorter and truncation if it is greater. The dimensionality is set to be 768. Larger batch sizes did not work at our end due to memory limitation. So, we choose the batch size to be 8. The dropout rate is set be 0.25, epochs 10, learning rate 1e-3 and Adam optimizer is chosen for optimization.

4 Experimental Results and Analysis

We present the results of binary and multiclass classification using ACC, F1-score (harmonic mean precision and recall) and AUC in Table 2.

4.1 Binary Classification Results

According to the results shown in Table 2, our proposed method Faker consistently outperforms all other methods in inferring binary classification labels (for the evaluation metrics ACC, F1-score, and AUC). For example, our proposed model Faker's accuracy score in inferring news articles is 20-30% higher than that of the state-of-the-art fake news detection models (TriFN, Declare, and Grover), as well as Transformer-based models (BERT, GPT-2, and VGCN-BERT), and other methods (SVM and DeepWalk).

TriFN outperforms other fake news detection baselines (Declare, Grover) in terms of overall performance. This is most likely because when we use both social contexts and news content, we get better patterns for detecting fake news.

Among the Transformer based models, the general performance of BERT and VGCN-BERT is better than GPT-2. The BERT model is more suited to generative (text generation) tasks, whereas the GPT-2 model is better suited to autoregressive (time-series) tasks. The fake news and Transformer-based baselines have outperformed the simple machine learning (SVM) and neural baseline (DeepWalk).

Model/ Metric	TriFN	Grover	Declare	BERT	VGCN-BERT	GPT2	SVM	Deep Walk	Faker
Binary Classification									
ACC	0.695	0.602	0.579	0.690	0.652	0.602	0.459	0.620	**0.824**
F1	0.660	0.598	0.552	0.612	0.635	0.609	0.468	0.610	**0.768**
AUC	0.698	0.678	0.577	0.619	0.632	0.648	0.430	0.542	**0.804**
Multiclass Classification									
ACC	0.675	0.582	0.559	0.660	0.650	0.582	0.400	0.519	**0.810**
F1	0.640	0.580	0.540	0.591	0.605	0.589	0.456	0.598	**0.750**
AUC	0.680	0.660	0.563	0.601	0.632	0.636	0.420	0.529	**0.780**

Table 2: Results of all models using Binary and Multiclass classification

4.2 Multi-label Classification Results

In addition to the simplified binary classification, we infer instance labels using the original 3-class label space, as shown in Table 2.

The results show that our proposed model Faker consistently outperforms all the models on multiclass classification on the quality metrics: ACC, F1-score and AUC. Similar to the results of binary classification, the general performance of TriFN is better than other fake news baselines. The BERT-based models (in general) performs better than GPT-2, which outperform simple baselines.

In terms of efficiency, the benefits of Faker are far more pronounced in the binary classification setting. This is most likely due to the fact that when the ‘mixed' label is removed, the models are better able to identify the instances as real or false.

4.3 Sampling Ratio

We sample the training set, which is controlled by a sampling ratio parameter θ ∈ {0.2, 0.4, 0.6, 0.8, 1.0}. Here, θ = 0.2 denotes 20% and θ = 1.0 means 100% of training instances used. We have shown the results with sample ratio of 1.0 in Table 2. For the other ratios, we show results in Appendices.

The results in Figure 4 in Appendix A show that our proposed model Faker consistently outperforms baselines in inferring binary labels by 5-30%. Figure 5 in Appendix B results also show that Faker’s scores during multiclass classification is consistently higher than other baselines for all values of θ. Overall, the F1-score and AUC of Faker is significantly higher in the multi-label classification compared to other approaches.

4.4 Precision-Recall of Binary Classification

We also test model perform on a small subset of 4000 instances for binary classification in Table 3.

	Actual Fake	Actual Real
Predicted Fake	2008	110
Predicted Real	37	1845

Table 3: Confusion Matrix of Sample data

The results in Table 3 show that Faker accuracy is 96.3%. We get the precision 94.8%, which means that we have a few false positives (news is real but predicted as fake) and we can correctly predict a large portion of true positives (i.e., news is fake and predicted as fake). We also get the recall value of 98.81%, which shows that we have much more true positives than false negatives. Generally, a false negative (news is fake but predicted as real) is a worse error than a false positive in fake news detection. In our experiment, we get less false negatives than the false positives (which are also fewer). Our F1-score is 96.46%, which is also high.

4.5 Effectiveness of Early Detection

In this experiment, we compare the performance of our model and baselines on early fake news detection. We follow the methodology of Liu and Wu (Y. Liu & Wu, 2018) to define a propagation path for each news story. The idea is that any observation data after the detection deadline cannot be used for training. According to the research in fake news detection, the fake news usually takes less than an hour to spread. Therefore, we choose minutes as the unit for the detection deadlines.

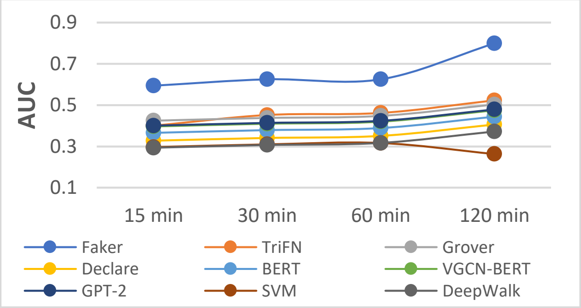

Figure 2: AUC of models on detection deadlines

The results in Figure 2 shows that, in general, the models perform better when the detection deadline delayed. This is shown with the overall better performance of those methods in later detection deadlines (except for SVM). This probably shows that more data obviously helps us to better classify the truth. Our proposed Faker model consistently achieves the best AUC for all the detection deadlines. Faker also achieve good performance even in the early stage after the news is released.

The ability of Faker to detect early is attributed to its position-aware mechanism, which learns the hidden patterns from the sequences of news data and tweets, and then classify the news articles. Using position encoding, the ranking position of each data point in a time-ordered sequence is considered. The model, then, pays more attention to those data points that reflect the truthfulness of the news article with respect to a temporal pattern. For example, the ranking position of a data point might give us an important clue as to whether a concerned news article is fake in the recent time.

4.6 Ablation Study

In ablation study, we remove a key feature component from a model one a time and investigate its impact on performance. Due to limited space, we just show the AUC performance of reduced variants with binary classification. In our experiments, we tested many variants of Faker but mention the important ones below:

Faker: Original model with news, tweets and side information.

Faker(n): Faker with only news-related information - removing social contexts (tweets).

Faker(s) Faker with only social contexts.

Faker(h-): Original Faker with headline removed from news content

Faker(b-): Original Faker with body removed

Faker(so-): Faker with news source removed.

The results are shown as in Figure 3:

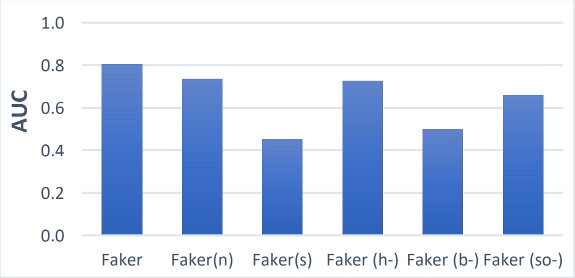

Figure 3: AUC of Faker's variants

From the results in Figure 3, we see that when we remove the social contexts as in Faker(n), the model performance is impacted but the model performance is impacted more when we remove the news content as in Faker (s). This probably shows that news related information is very important to learn the patterns of fake news. However, together both news and social contexts gives us more accurate results, as demonstrated by highest AUC of Faker. The Faker(n) variant also appears to indicate that the body text is not entirely responsible for the overall performance, but it is pretty close to the default system with all features.

The results also show that model performance is impacted more when we remove the news body, compared to the removal of the headline or the source of the news. This is seen with the lower accuracy of Faker(b-) compared to both the Faker(h-) and Faker(so-). This shows that headline and source are important, but news body alone carries more information. Between source and headline, the source seems to be more informative, this is perhaps related to the partisan information.

We also test different setting, for example, number of layers, dropout rate, number of heads, batch size, and removing certain embedding, such as positional embeddings. With all these experiments, we find that our current setup is the best for achieving our goals.

5 Related Work

Following the 2016 election, Google, Twitter, and Facebook all took steps to combat fake news. Facebook and Twitter also allow users to mark news stories as fake. A marked news story usually then goes through a manual fact-checking process. Manual fact-checking is inefficient for detecting fake news early because it is a time-consuming process, and it is also not scalable to handle a large volume of fake news online. In this paper, we look at automated methods for detecting fake news.

The automatic fake news detection methods can be broadly categorized as: content-based and social contexts-based methods. Most of the existing content-based detection methods (Horne & Adalı, 2017; Przybyla, 2020; Zellers et al., 2019) use style-based features (e.g., sentence segmentation, tokenization, bag-of-words, latent topics, and POS tagging) or linguistic features (e.g., frequencies of words, case schemes, context-free grammar and syntax etc.,) from news articles to detect fake news.

One challenge of content-based techniques is that fake news style, platform, and topics are changing constantly. Models trained on one dataset may perform poorly on a new dataset with a different content, style, or language. Furthermore, because the target variables in fake news change over time, certain labels become obsolete, while others must be re-labeled. These algorithms also necessitate a massive amount of training data to detect fake news. By the time these methods gather enough data, fake news has spread far enough.

To solve the issues of content-based methods, the researchers begin focusing on social contexts to detect fake news. The existing social contexts-based approaches are categorized into two types: (i) stance-based methods, and (ii) propagation-based methods. The stance-based approaches exploit the users' viewpoints from social media posts to determine the truth (De Maio et al., 2020; Y. Liu & Wu, 2020; Nakamura et al., 2020; Shu et al., 2019). The propagation-based methods (Huang et al., 2020; Jiang et al., 2019; Y. Liu & Wu, 2018; Qian et al., 2018) utilize the information related to the dissemination of fake news, e.g., how users spread it. These methods use techniques such as graphs and multi-dimensional points for fake news detection (Huang et al., 2020; Y. Liu & Wu, 2018).

While social context methods are useful when there is a lack of news content, they also introduce additional challenges. Gathering social contexts, for example, is a broad topic. The data for social contexts is not only large, but also incomplete, noisy, and unstructured, which may render existing detection algorithms ineffective.

Fake news detection is a subtask of text classification (C. Liu et al., 2019), which is solved by various machine learning and deep learning methods. Some work (Y. Liu & Wu, 2018) uses RNN and CNN networks to build propagation paths for detecting the fake news. Some other work (Shu et al., 2019) uses matrix factorization methods to detect fake news. A few works (Zellers et al., 2019) use LSTM networks on users' comments to explain if a news is real or fake. A few works (Nguyen et al., 2020) also uses graph networks to propose an explainable fake news detection system.

In recent years, there has been a greater focus in NLP research using the Transformer models, such as BERT (Devlin et al., 2018). BERT is used in some fake news detection models (Jwa et al., 2019; C. Liu et al., 2019; Vijjali et al., 2020) to classify the news as real or fake. Despite the robust design proposed in these models, a few limitations are noted. First, these models do not consider a richer set of features from the news items and social contexts. Second, the focus in these methods is not on early fake news detection.

The inclusion of temporal information is important to early detect fake news (Y. Liu & Wu, 2020). Also the inclusion of side (meta-data) information related to news or social contexts is important to understand the nature of fake news data (Shu et al., 2019). Recently, an exploratory study (Shahi et al., 2021) on fake news gives us more new insights about the timeline of misinformation. In our work, we consider both the temporal and side information to detect fake news.

The existing works on fake news focus either on news content or social contexts to detect fake news, we consider both in our work. Compared to some previous works (Nguyen et al., 2020; Popat et al., 2020; Shu et al., 2019) that consider both these aspects, we include a wider set of news-related as well as social context (tweets). A few works (Y. Liu & Wu, 2020; Shu et al., 2019) propose early detection of fake news. Compared to these methods, we can detect fake detect the fake news much earlier (i.e., after a few minutes of news propagation). Compared to the previous works, we consider the latest state-of-the-art neural architectures (Transformers).

6 Conclusion and Recommendations

In our work, we propose a Transformer-based architecture for fake news detection. We utilize the news content and social contexts to detect the patterns of fake news. We also early detect the fake news through a position-aware encoding. We achieve higher performance compared to the baselines, which shows the usefulness of our proposed approach. In addition to fake news detection, this model can also serve for general classification tasks.

To further improve the proposed method, a recommendation is to consider more social contexts, such as friends' networks, propagation paths and implicit users' feedbacks. It would also be very useful to consider malicious social media users' profiles and their activities. Another recommendation is to combat data and concept drifts. It would also be very useful to understand the tactics of fake news producers in real-time scenarios. Furthermore, data labelling scheme can be investigated because of the possibility of incorrectly labelled data, which may lead to data biases (Kishore Shahi, 2020). A possible extension of this work is to mitigate those biases. We also want to break filter bubbles and burst echo chambers created due to the spread of fake news.

References

Anderson, C. W. (2016). News ecosystems. *The SAGE Handbook of Digital Journalism*, 410–423.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 1–27.

Chen, E., Chang, H., Rao, A., Lerman, K., Cowan, G., & Ferrara, E. (2021). COVID-19 misinformation and the 2020 US presidential election. *The Harvard Kennedy School Misinformation Review*.

Chen, Q., Zhao, H., Li, W., Huang, P., & Ou, W. (2019). Behavior sequence transformer for E-commerce recommendation in Alibaba. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

De Maio, C., Fenza, G., Gallo, M., Loia, V., & Volpe, A. (2020). Cross-relating heterogeneous Text Streams for Credibility Assessment. *IEEE Conference on Evolving and Adaptive Intelligent Systems*, *2020-May*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *ArXiv Preprint ArXiv:1810.04805*.

Drummond, C., Holte, R. C., & others. (2003). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. *Workshop on Learning from Imbalanced Datasets II*, *11*, 1–8.

Horne, Benjamin; Gruppi, M. (2021). NELA-GT-2020: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles. *ArXiv Preprint ArXiv:2102.04567*.

Horne, B. D., & Adalı, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *ArXiv Preprint ArXiv:*1703.09398

Horne, B. D., NØrregaard, J., & Adali, S. (2019). Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology*, *11*(1).

Huang, Q., Zhou, C., Wu, J., Liu, L., & Wang, B. (2020). Deep spatial–temporal structure learning for rumor detection on Twitter. *Neural Computing and Applications*, *August*.

Jiang, S., Chen, X., Zhang, L., Chen, S., & Liu, H. (2019). User-characteristic enhanced model for fake news detection in social media. *CCF International Conference on Natural Language Processing and Chinese Computing*, 634–646.

Jwa, H., Oh, D., Park, K., Kang, J. M., & Lim, H. (2019). exBAKE: Automatic fake news detection model based on Bidirectional Encoder Representations from Transformers (BERT). *Applied Sciences (Switzerland)*, *9*(19), 4062.

Kaliyar, R. K., Goswami, A., Narang, P., & Sinha, S. (2020). FNDNet – A deep convolutional neural network for fake news detection. *Cognitive Systems Research*, *61*, 32–44.

Kishore Shahi, G. (2020). *AMUSED:* An Annotation Framework of Multi-modal Social Media Data. *arXiv preprint arXiv:2010.00502.*

Liu, C., Wu, X., Yu, M., Li, G., Jiang, J., Huang, W., & Lu, X. (2019). A Two-Stage Model Based on BERT for Short Fake News Detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 172–183.

Liu, Y., & Wu, Y. F. B. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 354–361.

Liu, Y., & Wu, Y. F. B. (2020). FNED: A Deep Network for Fake News Early Detection on Social Media. *ACM Transactions on Information Systems*, *38*(3).

Lu, Z., Du, P., & Nie, J. Y. (2020). VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12035 LNCS*.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1003–1011.

Mohammadrezaei, M., Shiri, M. E., & Rahmani, A. M. (2018). Identifying Fake Accounts on Social Networks Based on Graph Analysis and Classification Algorithms. *Security and Communication Networks*, *2018*.

Nakamura, K., Levy, S., & Wang, W. Y. (2020). r/Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. In *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*.

Nguyen, V.-H., Nakov, P., & Kan, M.-Y. (2020). FANG: Leveraging Social Context for Fake News Detection Using Graph Representation. *Proceedings of the 29th ACM International*

*Conference on Information & Knowledge Managemen*t, 1165-1174

Nørregaard, J., Horne, B. D., & Adalı, S. (2019). NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. *Proceedings of the 13th International Conference on Web and Social Media, ICWSM 2019*, *Icwsm*, 630–638.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710.

Popat, K., Mukherjee, S., Yates, A., & Weikum, G. (2020). Declare: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*.

Przybyla, P. (2020). Capturing the Style of Fake News. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(01), 490–497.

Qian, F., Gong, C., Sharma, K., & Liu, Y. (2018). Neural user response generator: Fake news detection with collective user intelligence. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 3834–3840.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Shahi, G. K., Dirkson, A., & Majchrzak, T. A. (2021). An exploratory study of COVID-19 misinformation on Twitter. *Online Social Networks and Media,* 100104

Shu, K., Wang, S., & Liu, H. (2019). Beyond news contents: The role of social context for fake news detection. *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, *9*, 312–320.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.

Vijjali, R., Potluri, P., Kumar, S., & Teki, S. (2020). Two stage transformer model for covid-19 fake news detection and fact checking. *ArXiv Preprint ArXiv:2011.13253*.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, *359*(6380), 1146–1151.

Wang, Y., Shen, Y., Liu, Z., Liang, P. P., Zadeh, A., & Morency, L. P. (2018). Words Can Shift: Dynamically Adjusting Word Representations Using Nonverbal Behaviors. *In Proceedings of the AAAI Conference on Artificial Intelligence,* 7216-7223.

Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., & Zhou, M. (2020). MIND: A Large-scale Dataset for News Recommendation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3597–3606.

Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., & Liu, H. (2019). Unsupervised fake news detection on social media: A generative approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 5644–5651.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. In *arXiv*preprint arXiv:1905.12616.

Zhou, X., & Zafarani, R. (2020). A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Computing Surveys*, *53*(5).

Appendix A. Binary Classification Sampling Ratios

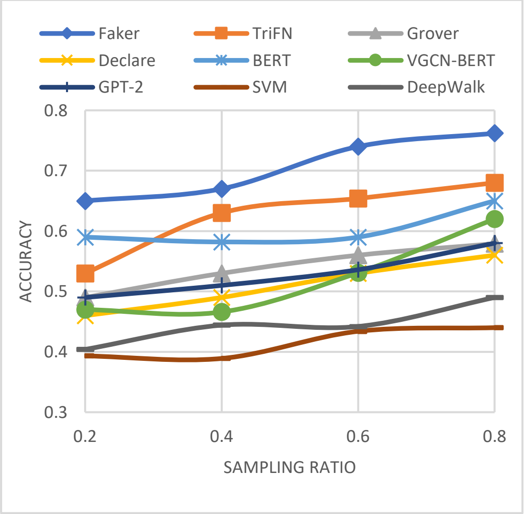

Figure 4 (a): Binary Classification Accuracy

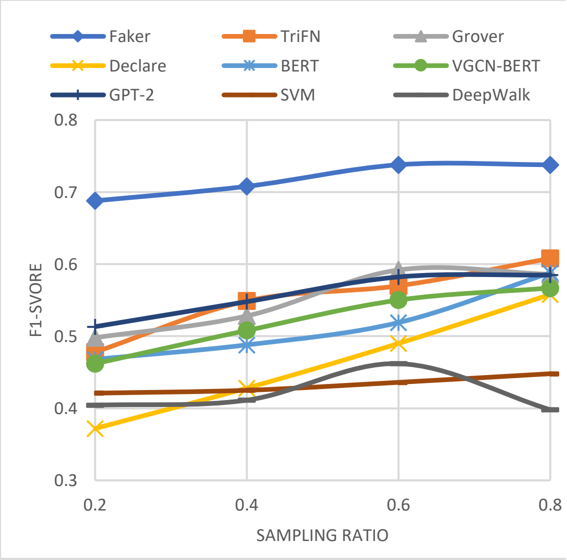

Figure 4 (b): Binary Classification F1-score

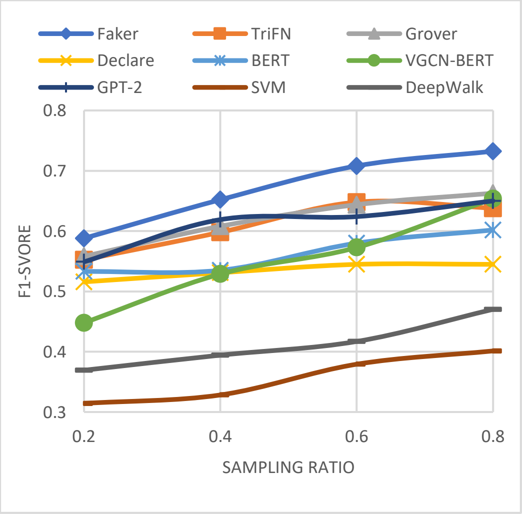

Figure 4 (c): Binary Classification AUC

Appendix B. Multiclass Classification Sampling Ratios

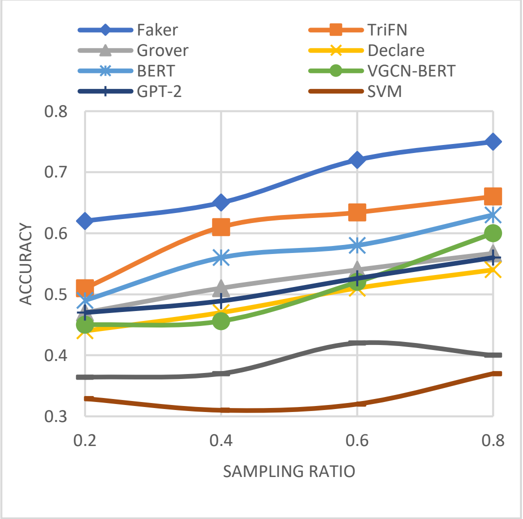

Figure 5(a): Multiclass Classification ACC

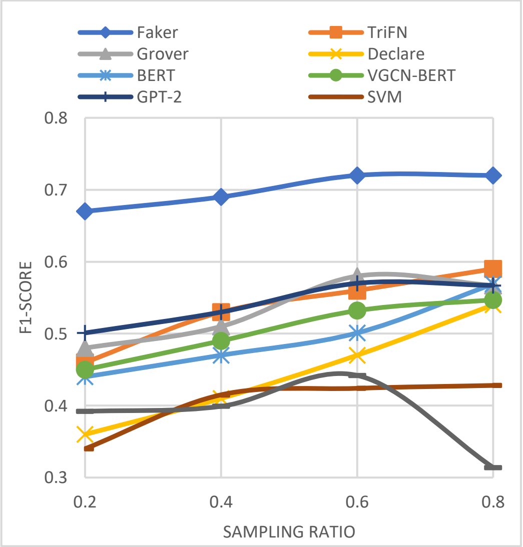

Figure 5(b): Binary Classification F1-score

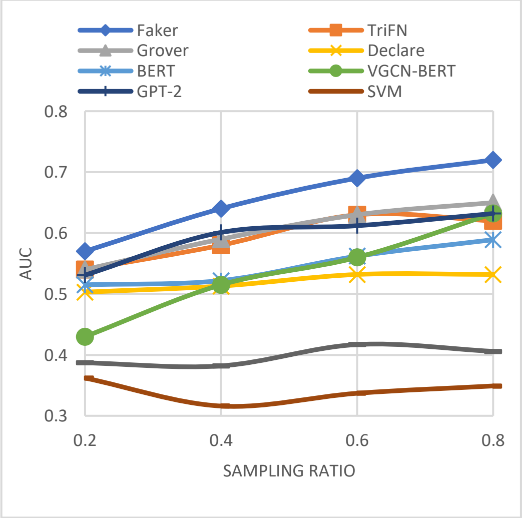

Figure 5(c): Multi-label Classification AUC