Adversarial Networks for Early Detection of Fake News

Eric Le, Xinyan Xie, Jianan Liu, Qiwei Ge

I. INTRODUCTION

Social media platforms such as Twitter, Facebook, Weibo, and Tiktok have become one of the major sources for people to acquire information rather than the traditional medias like TV or newspaper. With huge convenience they bring to people, however, the dissemination mode of information through social medias also accelerate the spread of fake news. According [1], it is estimated that about 65% news on social medias are fake news. Furthermore, fake news can easily spread faster, deeper, and wider on social media compared to traditional media [2].

It is obviously impossible to detect fake news manually on the social media because of the endless workload and its complexity. AI can be a good solution to deal with this problem. For example, NLP can be used for the text scenario and CNN for the image processing. Although there are tons of previous work in this area, these existing models do not perform well especially on newly emerging events, which is more close to real situation on today's social platforms. Thus in this project, we will focus on how to use Generated Adversarial Networks(GAN) for early detection of newly emerged fake news.

II. MOTIVATION

In the era of traditional media in the past, the right to publish news was firmly in the hands of journalists; they were all professional journalists, with their own industry norms and professional ethics. As professional gatekeepers, they were basically able to filter out fake news, the real and objective news, to the public. Very few cases exist in the news, and fake news is difficult to disseminate. However, with the continuous development of society, new media represented by Facebook, Twitter, and various mobile intelligent self-media have entered people's lives, changing the way contemporary people receive and read information. Each of us can be both a recipient and a producer of information. In the new media era, the disseminators of fake news may be released by news units due to negligence, interests, and other factors. They may also be self-media people, network correspondents, and even ordinary people from all walks of life.

The more severe impact of fake news may be a situation that directly affects the political situation in the country, such as elections. During the 2016 US presidential election on Twitter, bots were responsible for the early promotion of misinformation, that they targeted influential users through replies and mentions and that the sharing of fact-checking articles nearly disappears in the core of the network, while social bots proliferate. [3] In the new media communication environment, we only need a mobile phone and a network to publish information anytime and anywhere. The information spreads faster and more widely, invisibly expanding the negative impact of fake news. Producing fake news is no longer a difficult thing to do.

The spread of fake news on social media can cause huge political and economic losses to society. For example, during the 2016 US presidential election, there was fake news on Facebook with the title *"Pope Francis Shocks World, Endorses Donald Trump for President"*, receiving 960,000 engagements. On Twitter during the same time, there were more than 19 million bot accounts publishing or retweeting tweets to support either presidential candidate to interfere with the US election [4]. Therefore, it is extremely important to prevent the spread of newly emerging fake news at the earliest time before it becomes widespread. There are two goals for fake news detection. The first goal is to improve the detection accuracy to filter newly emerged fake news as much as possible, and the second is to improve the detection time to stop the fake news' spreading as soon as possible.

III. PROBLEM

The main problem we are looking to solve is the issue of newly emerging events in fake news. As we all know, fake news can come from every aspects of our life, such as politics, healthcare, religion, education and so. Currently deep learning models have a built in assumption when they are trained. This is that they know the entire life cycle of social media post. Which means that the model has all the information when the fake news was labeled faked news. When a new event occurs, however, due to a lack of corresponding prior knowledge, it's hard to identify whether the newly emerged news is fake or not. This in turns means that existing event-specific models will also fail to detect the newly emerged news, because they only capture event-specific features which are not shared among different events. Besides that, there is another problem occurring even in the same field. Since the speed for news updating is extremely fast, the features in one specific field will change rapidly. Therefore, it is important to find a model which can detect the common features of the fake news among different fields of events and accurately classify the news between fake and real.

Adversarial networks [5] will be considered to solve this problem. This method can not only predict whether the posts are fake or real but also decreases the dissimilarity among all events. Yaqing et al. [6] have successfully applied this method to the news from Twitter and Weibo and they achieved an accuracy of 0.715 and 0.827 respectively. Based on their model, we are going to improve it from two aspects, input information and model structure. As for input information, instead of only using the textual post and the attached image, social-Context based information are also considered, such as the number of re-posting and likes, the rating level, or the comment information related to the post. For model structure, more feature extractors are going to be examined. For example, VSM can be used to extract text features and ResNet can be used to extract features from images.

IV. Dataset

For the project, there are two main tasks. The first one is the binary label classification, which only outputs either true or false given a post. The second one is multi-class classification, which are more than two classes rather than just true or false. The two datasets used in this project are MediaEval 2016 [6] and Liar [7] respectively. In this section, we will introduce the what the datasets look like, how to preprecess them, and what the output is, which will be fed directly into the computational models in the next section.

A. MediaEval 2016

The MediaEval 2016 dataset has abundant twitter information including texts, pictures, social context, etc. There are four files that will used in our paper, which can be seen below.

1) set_images.txt: This file include 4 columns, which are *image_id, image_url, annotation, event*. The image_id is the unique identifier of each image, and the image_url is the link to download the image. The annotation is the label to say if the image is true or false. The event records the event info, which will be used in our model. There are total 399 images in this file, however, only 207 images are available because of the dead link. Therefore, we will use 207 images in our project.

2) posts.txt: This file is the main file which records the tweets information, and the columns are *post_id, post_text, user_id, image_id(s), username, timestamp, label*, respectively. Each tweet will be identified by its unique post_id, and the post_text will includes the text content. However, it also include some useless information such as some url links. We filter out those useless information through REGEX. Each tweet might includes some pictures from set_images.txt and image_id(s) will store that info.

3) post_features.txt: This file contains extensive features of a post. The columns are *post_id, num_words, text_length, contains_questmark, num_questmark, contains_exclammark, num_exclammark, contains_happyemo, contains_sademo, contains_firstorderpron, contains_secondorderpron, contains_thirdorderpron, num_uppercasechars, num_possentiwords, num_negsentiwords, num_hashtags, num_URLs*. Among these features, num_mentions and num_retweets are not used since when the fake tweet starts to spread, it may not get much attention like num_mentions or num_retweets. The rest features are used in this project.

4) user_features.txt: This file contains social context information about the person who post the tweets. The columns are *post_id, num_friends, num_followers, folfriend_ratio, times_listed, has_url, is_verified, num_posts*. These features can be used to make the prediction, which is one of the contributions of the project, taking advantage of the social context infomation to detect fake news.

The posts.txt file include 15629 tweets in total, and after applying left join with post_features.txt and user_features.txt based on its unique identifier, i.e., post_id, removing the null values, adding event columns for each record, there are 13420 records left. Each record consists of 31 features. After the data pre-processing, the data can be feed into the model. Parts of the data can be seen in Figure **??** .

	post_id	user_id	image_id(s)	event	post_text_remove_url	label
0	324597532548276224	886672620	boston_fake_03,boston_fake_35	boston	Don't need feds to solve the #bostonbombing wh...	fake
1	325145334739267584	21992286	boston_fake_23	boston	PIC: Comparison of #Boston suspect Sunil Tripa...	fake
2	325152091423248385	16428755	boston_fake_34	boston	I'm not completely convinced that it's this Su...	fake
3	324554646976868352	303138574	boston_fake_03,boston_fake_35	boston	Brutal lo que se puede conseguir en colaboraci...	fake
4	324315545572896768	180460772	boston_fake_15	boston	4chan and the bombing, just throwing it out th...	fake
5	324581777614180352	46224814	boston_fake_08	boston	4chan thinks they found pictures of the bomber...	fake
6	324665423956176896	90735851	boston_fake_35	boston	Ola ke ase, investigando las bombas de Boston ...	fake
8	325099014355820544	21769179	boston_fake_13	boston	@DLoesch have you seen this? Bomber #2 looks ...	fake
9	325705653026975744	51410043	boston_fake_03,boston_fake_35	boston	da 4chan think tank BOSTON #photos #suspect...	fake
10	324637464927039488	224438064	boston_fake_03,boston_fake_35	boston	@Anonymoussops @bernardosampa @urbanohumano de...	fake

Fig. 1. Overview of the MediaEval 2016 dataset

B. Liar

LIAR is a dataset for fake news detection with 12,800 human labeled short statements from politifact.com's API, and each statement is evaluated by a politifact.com editor for its truthfulness [8]. Compared to MediaEval 2016, Liar is a multi-class dataset. Instead of simply classifying the news into two categories, i.e., true or false, it considers six fine-grained labels for the truthfulness ratings: pants-fire, false, barelytrue, half-true, mostly-true, and true. However, only four features are used for Liar, which are post_id, post_text, event, and label. The value of rest 27 features are set to 0s in consistent with the input data format. Parts of the Liar can be seen in Figure 2.

id (string)	label (class label)	statement (string)	subject (string)
		beliefs).	
7891.json	0 (false)	Says Having organizations…	campaign-finance,congress,taxes
8169.json	1 (half-true)	Says nearly half of Oregons…	poverty
929.json	1 (half-true)	On attacks by Republicans tha…	economy,stimulus
9416.json	0 (false)	Says when armed civilians stop…	guns
6861.json	3 (true)	Says Tennessee is providing…	education,state-budget
1122.json	0 (false)	The health care reform plan…	health-care

Fig. 2. Overview of the Liar dataset.

The task of this project is to develop a generalized fake news detection model independent with the datasets. Some previous work are highly dependent on the datasets, they may perform well on the binary classification but when it comes to the multi-class dataset, the prediction accuracy drops down significantly. That is the reason why both binary classification and multi-class datasets are used in our project.

V. COMPUTATIONAL MODELS

Our computational model is based on the paper [6], and this architecture contains three parts: multimodal Feature extractor, fake news detector and event discriminator. The entire architecture is drafted, as shown in Figure 3. The multimodal Feature extractor is depicted inside the red dashed rectangle, the fake news detector is depicted inside the

upper purple dashed rectangle and the event discriminator is depicted inside the lower purple dashed rectangle. The blue rectangles represent the different types of input, such as images and the text content, and the pink rectangles represent the classification outputs for detecting whether it is a fake news and whether it is belong to a specific category among a range of fixed categories. The orange rectangles represent the different CNN architectures for a specific purposes. Here, we use Text-CNN to extract the informative features from textual context, VGG-19 to extract visual features from the images, detector CNN to predict whether the posts are fake or real from the multi-modal feature representation and discriminator CNN to classify the post into one of K events from the multi-modal feature representation.

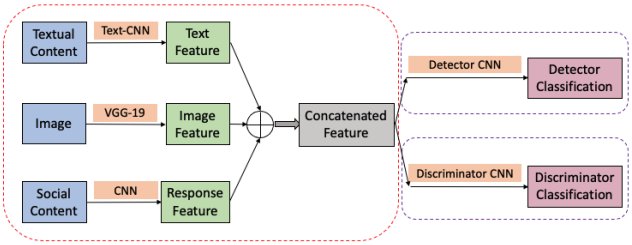

Fig. 3. The drafted architecture.

The process for extracting the textual feature is described as follows: Firstly, the whole sentence is converted to a list with all individual words; we feed the list into the text-CNN architecture and get the word embedding vector for each word; the concatenation (as indicated in Equation 1) of different parts of word embedding vectors are used to represent the different part information from the sentence with a filter size h, as indicated in Equation 2; the whole sentence can be represented by the different word embedding vectors, as indicated in Equation 3; Finally, we apply the max-pooling operation to the whole sentence representation and get the maximum value to represent the corresponding feature for the given filter h. The final textual feature can be represented by Equation 4.

$$T_{1:n} = T_1 \oplus T_2 \oplus ... \oplus T_n, \tag{1}$$

$$t_i = \sigma(W_c \cdot T_{i:i+h-1}), \tag{2}$$

$$t = [t_1, t_2, ..., t_{n-h=1}], \tag{3}$$

$$R_T = \sigma(W_{tf} \cdot R_{T_c}), \tag{4}$$

For the visual feature extractor, a fully connected layer is added on top of the last layer of VGG19 network to modify the visual feature dimension and it can be represented by Equation 5.

$$R_V = \sigma(W_{vf} \cdot R_{V_{vgg}}), \tag{5}$$

We also add the social content information into our model, such as the number of the forwarding, the comment altitude under each post. They are consist of the numbers and the words. We use the regular CNN to extract the response features, which is represented as R_S.

The multi-modal feature is created by the concatenation of the texture feature representation and the visual feature representation, as shown in Equation 6.

$$R_F = R_T \oplus R_V \oplus R_S, \tag{6}$$

There are two kinds of outputs in this model, one is to detect whether the specific post is real or fake and the other one is to classify the category of the post.

The fake news detector is denoted as $G_d(.;\theta_d)$, and θ_d represents the parameters in the detector model. The probability of the post being fake is represented as Equation 7. And the cross entropy is employed to calculate the corresponding loss, as shown in Equation 8. The detection loss is expected to minimized by tuning the parameters $\hat{\theta}_f$ and $\hat{\theta}_d$, as shown in Equation 9.

$$P_\theta(m_i) = G_d(G_f(m_i;\theta_f);\theta_d), \tag{7}$$

$$L_d(\theta_f,\theta_d) = -\mathbb{E}_{(m,y)\sim(M,Y_d)}[y\log(P_\theta(m)) + (1-y)(\log(1-P_\theta(m))], \tag{8}$$

$$(\hat{\theta}_f,\hat{\theta}_d) = \arg\min_{\theta_f,\theta_d} L_d(\theta_f,\theta_d), \tag{9}$$

The event discriminator is denoted as $G_e(R_F;\theta_e$, and θ_e represents the parameters in the discriminator model. The cross entropy is also employed as the loss of event discriminator, as shown in Equation 10, and it is required to be minimized during the training process, as indicated by Equation 11.

$$L_e(\theta_f,\theta_e) = -\mathbb{E}_{(m,y)\sim(M,Y_e)}[\sum_{k=1}^{K} 1_{[k=y]}\log(G_e(G_f(m;\theta_f));\theta_e)], \tag{10}$$

$$\hat{\theta}_e = \arg\min_{\theta_e} L_e(\theta_f,\theta_e), \tag{11}$$

There are two purposes in this model: (1).The post is clearly classified as the fake news or the real news, which is to minimize the detection loss $L_d(\theta_f,\theta_d)$; (2). The model is also to fool the event discriminator, which is to maximize the event discriminator $L_e(\theta_f,\theta_e)$. And the combination of these two losses is shown in Equation 12. The final parameter set should be the saddle point of the final objective function (shown in Equation 13 and Equation 14).

$$L_{final}(\theta_f,\theta_d,\theta_e) = L_d(\theta_f,\theta_d) - \lambda L_e(\theta_f,\theta_e), \tag{12}$$

$$(\hat{\theta}_f,\hat{\theta}_e) = \arg\min_{\theta_f,\theta_d} L_{final}(\theta_f,\theta_d,\hat{\theta}_e), \tag{13}$$

$$\hat{\theta}_e = \arg\max_{\theta_e} L_{final}(\hat{\theta}_f,\theta_e), \tag{14}$$

The whole algorithm can be summarized in Algorithm 1, which is also from the paper [6].

Algorithm 1 Event Adversarial Neural Networks.

Input: The multi-modal input $m_i{}_{i=1}^{N}$, the auxiliary event label $e_i{}_{i=1}^{N}$, the detection label $y_i{}_{i=1}^{N}$ and the learning rate η

1: **for** number of training iterations **do**
2: Decay learning rate according to Eq.15
3: Update the parameters of multi-modal feature extractor θ_f according to Eq.14;
4: Update the parameters of the event discriminator θ_e:
5: $\theta_e \leftarrow \theta_e - \eta \frac{\partial L_e}{\partial \theta_e}$
6: Update the parameters of fake news detector θ_d:
7: $\theta_d \leftarrow \theta_d - \eta \frac{\partial L_d}{\partial \theta_d}$
8: **end for**

VI. LITERATURE REVIEW

In this section, we will do the literature review work under three aspects: related work in fake news detection field, in adversarial networks field, and in our specific problem, i.e., GAN for early detection of fake news.

A. Related work in fake news detection

There are many previous work on the fake news detection field. For example, there are similar research topics such as spam filter [9] or rumor filter [10] already been conducted. In fake news detection filed, the major problem is to decide what are the effective features and how to extract them. Those features can be pure text, emojis, or even images, and videos. According the complexity of the feature selection in one or multiple fields, the fake news detection models can be divided into two categories: single-field model and multi-field model.

Single-field model just extract features from a single filed. Textual features are nothing but just semantic features extracted from text content like posts in Twitter etc. There have been many related work in this filed [11]–[14]. The major challenge is that natural language and linguistic patterns are not fully understand because of the high dependency on context information [15]. Therefore, it is difficult to develop a general ML model to solve all fake news problems.

Visual features can also be extracted from pictures or videos. Visual features can also be considered as a vital indicator in fake news detection model. There are some work done in this field [14], [16]. However, the features are still hand-crafted and can hardly represent complex distributions of visual contents.

Social context features play another important role in fake news detection since it can capture the propagation pattern of news. Those feature are like the number of followers, retweets, or hash-tags/topics [17].

Multi-field model just utilize features from more then one field and fuse them, feeding them into the deep neural networks, and make the fake new detection [18]. The problem is still that even though the multi-field model take advantage of features in different fields, it still cannot generalize well when the dataset changes.

B. Related work on Generative adversarial networks(GAN)

GAN is one type of generative neural networks, Which consists of a generator and a discriminator [5]. In the backpropogation process, the discriminator is used to distinguish whether the input comes from real datasets or fake samples generated by the generator, while the generator is used to generate realistic samples based on the sampling distribution from datasets to confuse the discriminator [4]. After many iterative steps, both the generator and discriminators comes to an equilibrium and neither can be improved anymore.

Fake new detection models should have the ability to identify the unseen or newly emerged events, rather than just have a high performance on the existing dataset. The useage of GAN-based models are a promising direction to build generalized fake news detection models with multi-field features [6].

There have been some existing GAN-based models, they have already been applied to several fields such as domain adaption [19] and discriminative image features [20]. The proposed model is build based on top of GAN, which ultilize a MinMax framework between the event discriminator and multi-filed feature extractor. Specifically, the feature extractor can be considered as a generator, which generates the false events to confuse the discriminator. This can decrease the model dependencies on the training dataset, it can be generalized to detect the fake news even with unseen data.

C. Related work for our specific problem

Generality for new events as a domain issue for fake news detection is a relevantly new problem. There has only been a few papers that have dealt with this issue in depth. There are new methods that have been tired that we have seen so far. One of these methods is using transfer learning and the other method of dealing with this issue is reinforcement learning.

One way of dealing with generality to new events is using transfer learning. Transfer learning is a where a model is built using a lot of different domain knowledge, this domain knowledge is then used to train a new a model that ventures into a different domain. The most notable transfer learning models out there are built upon ImageNet. ImageNet is a large database of images of different things, and train on a model. When someone wants to train a new a model to detect something new, they use the previous information from a model trained on ImageNet to extract new information for there dataset. This concept was shown in [21].

Another method that has been tried out is reinforcement learning. Reinforcement learning is basically the idea that you have an agent that is trying to learn something through trail and error. There is a reward function that gives out rewards or penalty to different actions that the agent makes. Reinforcement learning in this case would be based on the idea that in order to learn what is fake news it needs to have more generality. [22] created a reinforcement learning based model called REinforced Adaptive Learning Fake News Detection (REAL-FND). They created a fake news classifier and a domain classifer that are trained on fake news. These are then used as the reward function in the RL training. Their model worked well on cross-domain classification of fake news.

VII. Modifications

In this section, we will introduce our modifications which will potentially improve the performance to generally predicting whether the post is fake or not. We employ BERT (Bidirectional Encoder Representations from Transformers) to extract the textual features from the texts, Resnet-18 to extract the image features from the images, CNN-based method to extract the social-context features from the social contexts, and Attention-based C-BiLSTM to classify the post into different levels of the truthiness.

A. Textual Feature Extractor

1) Tokenizers: To make network easily understand the input of textual information, the words and the sentences need to be translated into a form that the model can understand. Here, we use a tokenizer method from HuggingFace [23]. Bert tokenizer is employed and is pretrained on English language using a masked language modeling (MLM) objective [24], [24], [25]. The bert tokenizer architecture is shown in Figure 4.

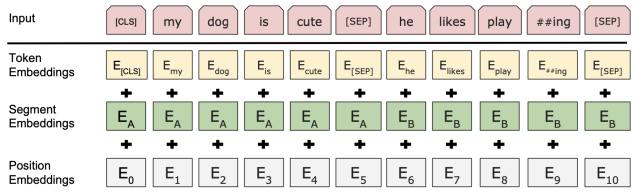

Fig. 4. BERT input representation.

From Figure 4, it can be seen that bert tokenizer consist of three embeddings: token embeddings, segment embeddings, and position embeddings. We can see that a [CLS] is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence. This is used to indicate different sentences. Besides that, segment embeddings are also employed to differentiate different sentences by assigning them to different parts. For example, all words in first sentence are assigned to one token and the words in second sentence are assigned to another token. The positional embedding shows the position information for each word in one input textual sample. The input embeddings are the sum of these three embeddings.

2) Textual Feature Extractor: We use Bidirectional Encoder Representations from Transformers (BERT) [24] to extract the latent space from textual information and it can learn contextual relations between words (or sub-words) in a text. The original Transformer is from the google paper [26], and the overall architecture is shown in Figure 5. It can be seen that there are two separate networks. An encoder is used to read the input array and produce the latent spaces from different layers. A decoder is used to transform the information from these latent spaces and make predictions for specific tasks.

In the BERT model, only the encode part is employed to perform a number of different Natural Language Processing (NLP) tasks. The specific sequential layers are added following the encode network. In our project, we need the same size features from the different aspects of the posts, which is 1×32. Therefore, a fully connected layer with 32 nodes and a Leaky Rectified Linear Unit (ReLU) [27] activation layer follow BERT network.

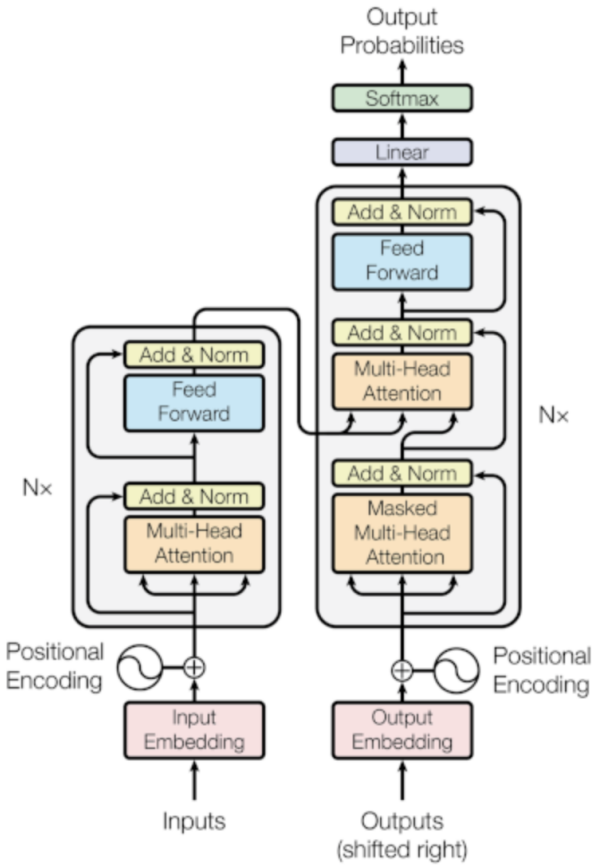

Fig. 5. Transformer network.

B. Image Feature Extractor

1) Image Resize: The image sizes from posts are different. Before feeding them into a CNN model, we need to resize all images with a specific size. We use the cubic interpolation over 4×4 pixel neighborhood and create an image dataset with size of 224×224 as the input for Resnet18 [28].

2) Resnet-18: Resnet [28] architrcture has a deep residual learning framework, where all layers fit a residual mapping. And it is proven that it can effectively solve the degradation problem. The whole arhitecture is shown in Figure 6.

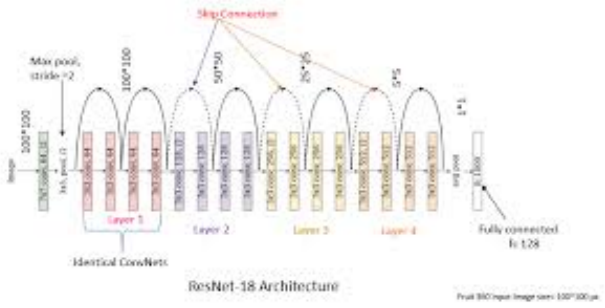

Fig. 6. Resnet Architecture

From Figure 6, we can see that there are several convolution blocks and the output of this network is a fully connected layer

with 1000 nodes. In our project, we use the resnet architecture with 18 layers. The detailed information is presented in Figure 7. It indicates that each convolution blocks replicates twice and there are a total number of 8 convolution blocks. To get the same size of the feature map as the textual context, a similar fully connected layer with 32 nodes and a Leaky Rectified Linear Unit (ReLU) [27] activation layer are following Resnet network.

layer name	output size	18-layer
conv1	112×112	
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$
	1×1	
FLOPs		1.8×10^9

Fig. 7. Resnet-18 parameters

C. Social-context Feature Extractor

The significant modification in our project is the extraction of the social-context feature. Based on the features provided by the dataset, the columns *num_words, text_length, num_questmark, num_exclammark, contains_happyemo, contains_sademo, contains_firstorderpron, contains_secondorderpron, contains_thirdorderpron, num_uppercasechars, num_possentiwords, num_negsentiwords, num_hashtags, num_URL num_friends, num_followers, folfriend_ratio, times_listed, has_url, is_verified, num_posts* are used. All these feature are considered as the structured data. They can be generally classified into two feature types, one categorical feature type and the other one numerical feature type.

1) Categorical feature type: Categorical feature type determine whether the feature is true or fake, such whether the post contains happy emojis or not. This kind of feature type has a binary answer. Therefore, each answer will be 0 if it is false and 1 if it is true. The categorical features contain columns of *contains_happyemo, contains_sademo, contains_firstorderpron, is_verified, contains_secondorderpron, contains_thirdorderpron, has_url*.

2) Numerical feature type: Numerical feature type indicates the specific number of the answer, such the number of the question marks is 6. And it also can be divided into 3 kinds, integer with small numbers, integer with large numbers and floating points.

For the answer with integer with small numbers, we firstly get the maximum value of this column and then change each sample to one-hot coding form. It contains columns of *num_words, num_questmark, num_exclammark, num_uppercasechars, num_possentiwords, num_negsentiwords num_hashtags, num_URLs*.

For the answers with integer with large numbers, we get the maximum value of this column, divide all values uniformly into different classes, classify the values into classes, and then change them into one-hot coding form. It contains columns of *text_length, num_followers, times_listed, num_friends, num_posts*.

For the answers with floating points, we just concatenate the value to the feature array. It contains columns of *folfriend_ratio*.

Finally, we concatenate these three answers together and form a feature with size of 1×1069. Then we use the linear transformation to change it the size of 1×32, which is the same size for the textual features and the image features.

D. Fake News Classifier

The second main task of our model is to improve the multi-class classification ability of the model. DAGA-NN has that you can improve the model by changing different components of the EANN model such as the textual extractor, image extractor, and fake news classifier. Their modifications were shown to improve the model . However, when they tested their model on the Liar database, their model's accuracy significantly decreased from 94% to 29%. This is mostly likely due to their graph based fake news model inability to deal with multi-class classification. Therefore, in our proposed implementation, we will be trying a different model that has shown its ability to hand multi-class fakes news well.

The model that we are using is called an attention-based convolutional bidirectional long short-term memory (AC-BiLSTM) [29]. In the original AC-BiLSTM, a text would go through some pre-processing to remove things such as stop words and then be converted into a word embedding. This word embedding would then go through a CNN, a max pooling layer, a BiLSTM model, and attention layer, a dense layer, flattened layer and then an output layer as can be seen in Fig. 8. However for our process, the pre-processing and word embedding will be taken out since that occurs during the textual feature extraction process. Rather the result of the multi-modal extractor will be the input for the AC-BiLSTM. The rational for using this model is given its ability to achieve a mutli-label classification of 35.1% compated to DAGA-NN's 29%.

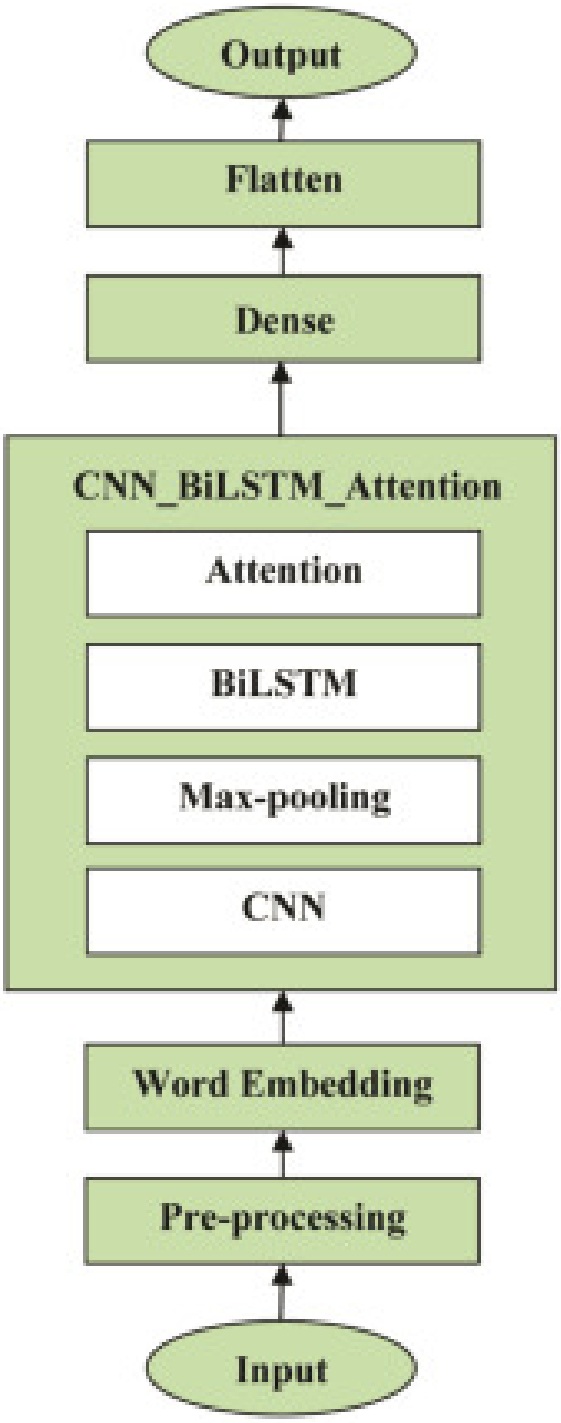

Fig. 8. AC-BiLSTM.

VIII. Implementation and Results

Given that the EANN code is openly available, we decided to base our code off of it and modify it according to our needs. The code is written in Python and for the most part everything came from the Numpy and Pytorch package. Due to the large nature of the pretrained ResNet-18 model, the program was run on HCC. A computing center at the University of Nebraska-Lincoln.

A. Modifications

As previously stated before from the algorithm section, there are multiple parts of the original code-base that needed to be changed. The text input won't be going into a Text-CNN anymore, so that part of the code was changed in a BERT model.

The second modification is the image feature extractor. The previous image feature extractor was a VGG-19 model. This model was replaced with a ResNet-18 model. The original plan was to use a ResNet-50 model, but given it's large size, we decreased it down to ResNet-18 model so that it would be easier to run on HCC.

The next part that needed to be done was getting the social-context data read in correctly. The function that read in the image and text data was changed so that it read in the file that contained all of the social-context information and made sure that each went with it's proper sample. The model function then was also changed so that it included a basic sequential function that read in the social-context information and changed into a feature representation for the multi-modal feature extractor.

The last part that needed to be changed was the fake news classifier. The sequential model was removed and a combined CNN and BiLSTM was created so that it would take input. However, it was not used during testing because of issues that arose during training.

B. Issues

During the implementation phase, there were some issues that arose. One issue that arose was Pytorch. Given that this code was written back in 2018, the code is outdated for the current version of Pytorch. It is possible to create a virtual environment and run the code in the version of Pytorch that it was written in, but from what we can see it didn't specify what version of Pytorch it was written in. Therefore, it was more easy for us to rewrite the code in order to get it running.

C. Preliminary Results

After some modification, the code was able to run. The preliminary results from the training is the accuracy of 60% and and validation accuracy of 60%. However, there was major issue in that the loss was not changing, so even though it was going through each epoch the loss for each epoch would remain in the area of 4%. The Liar dataset was also test to see if the same problem would occur with the Liar dataset and it was the same thing. So it is most likely that either there is something wrong how the data is being inputted in the model or that that there is some wrong with the backward propagation function that was developed.

REFERENCES

[1] T. Blatchford, "Americans believe two-thirds of news on social media is misinformation," Available at https://www.poynter.org/ethics-trust/2018/americans-believe-two-thirds-of-news-on-social-media-is-misinformation/ (2018/06/20).

[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] A. Bovet and H. Makse, "Influence of fake news in twitter during the 2016 us presidential election," 2019.

[4] B. Guo, Y. Ding, L. Yao, Y. Liang, and Z. Yu, "The future of false information detection on social media: New perspectives and trends," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–36, 2020.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets in: Advances in neural information processing systems (nips)," 2014.

[6] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "Eann: Event adversarial neural networks for multi-modal fake news detection," in *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, 2018, pp. 849–857.

[7] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, 2017.

[8] "Liar." [Online]. Available: https://huggingface.co/datasets/liar

[9] Z. Jin, J. Cao, Y.-G. Jiang, and Y. Zhang, "News credibility evaluation on microblog with a hierarchical propagation model," in *2014 IEEE international conference on data mining*. IEEE, 2014, pp. 230–239.

[10] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[11] 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.

[12] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "Tweetcred: Real-time credibility assessment of content on twitter," in *International conference on social informatics*. Springer, 2014, pp. 228–243.

[13] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *2013 IEEE 13th international conference on data mining*. IEEE, 2013, pp. 1103–1108.

[14] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[15] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 797–806.

[16] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *IEEE transactions on multimedia*, vol. 19, no. 3, pp. 598–608, 2016.

[17] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *2015 IEEE 31st international conference on data engineering*. IEEE, 2015, pp. 651–662.

[18] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 795–816.

[19] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[20] Z. C. Lipton and S. Tripathi, "Precise recovery of latent vectors from generative adversarial networks," *arXiv preprint arXiv:1702.04782*, 2017.

[21] V. S. Tida, D. S. Hsu, and D. X. Hei, "Unified fake news detection using transfer learning of bidirectional encoder representation from transformers model," 2022. [Online]. Available: https://arxiv.org/abs/2202.01907

[22] A. Mosslanezhad, M. Karami, K. Shu, M. Mancenido, and H. Liu, "Domain adaptive fake news detection via reinforcement learning," 2022.

[23] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pre-training compact models," *arXiv preprint arXiv:1908.08962v2*, 2019.

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

[27] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1. Citeseer, 2013, p. 3.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] T. E. Trueman, A. K. J., N. P., and V. J., "Attention-based c-bilstm for fake news detection," *Applied Soft Computing*, vol. 110, p. 107600, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S156849462