General FakeFlow: Multi-Domains Fake News Detection by Modeling the Flow of Affective Information

Stanford CS224N {Custom} Project

Anthony TAING
Department of Computer Science
Stanford University
anth248@stanford.edu

Abstract

Fake news detection merely based on news content is tremendously challenging due to the usage of different text length, due to the variety of sources and different styles. However we notice that they always play with some affective factors to manipulate the readers. In this paper, we aim to reproduce the Fakeflow model which learns the flow of emotions by combining the topic and affective information. We evaluate the model's performance with several experiments on three real-world datasets by using a multi-domain contexts to test its transferability. By using differents data sets, the results show that capturing this flow of emotions is helpful, the model still outperforms some baselines but it becomes weak when dealing with shorter texts and unseen topics. Then the combination of the two branches Topic information and Affective information seems to be relevant and useful for this task. Last, we have proposed a new version of Fakeflow model which improves the performance in this cross-domain environment with these datasets.

Key Information: This is a custom project with mentor Manan Rai and no external collaborators; it is not shared with another course.

1 Introduction

The emergence of social and news media provide convenient conduit for users to create, access, and share diverse information. Due to the increasing usage of the web, more people seek out and receive timely news information online, where news broadly includes articles, claims, statements, speeches, posts, among other types of information related to public figures and organizations. Meanwhile, there is also an explosive growth of fake news, which contains miss-information. Fake news has altered society in many areas [1], [2],[3] and its detection is even more challenging with the speed and the differents sources of information available. Fake news is intentionally written to mislead readers, and some methods from several perspective have been used: the false knowledge it carries [4], its writing style [5], its propagation patterns [6], and the credibility of its source [7]. This problem has attracted increasing attention in recent years with more demand for fake news detection and intervention but prior works on fake news detection entirely rely on the datasets from social media or blogs, or data come from a particular domain like politics.

However despite the limited available datasets, which mainly contain short texts, they are small in size, or they have a limited number of category that limits scopes of context and styles of writing [8], the task could be even more challenging and generic if we study this fake news detection problem in multiple domain scenarios and when covering a wide range of topics.

Consequently, fake news authors are putting efforts to make their news articles look more realistic, they add misleading terms or events that can have a negative or positive impact on the readers' emotions. They also expose the readers to be emotionally manipulated while reading longer texts that

have several imprecise or fabricated plots. But previous works demonstrated that fake news has a different distribution of affective information across the text compared to real news, e.g. more fear emotion in the first part of the article or more overall offensive terms [9].

In this work, we aim to extend the research of the Fakeflow model, which models the flow of affective information in texts, in particular we want to know how well it generalizes on cross-domains contexts by using differents datasets with a variety of topics, we would be able to know how capturing the flow of emotions is effective.

2 Related Work

Previous work on fake news detection are using external resources (e.g. Web, knowledge sources) to verify the factuality of the news [10], most of them also rely on a particular domain like political, or the focus is mainly on proposing new feature sets [11] including readability (number of unique words, SMOG readability measure, etc.), stylistic (frequency of partof-speech tags, number of stop words, etc.) and psycholinguistic features (i.e., several categories from the LIWC dictionary [12]. Next, few researchers have experimented new systems across multiples domains [13], [14] to test their robustness on unseen data.

Then to detect fake news, other researchers explored auxiliary information to improve detection [15], they have used all information available by studying the social context during news dissemination process on social media. The focus was on each author, publisher, and user who might have written, published, or spread the news stories. Next, previous works used a combination of encoders, FakeNewsTracker [16] is a deep neural network-based model that consists of two branches: one encodes news article texts and the other encodes social media engagements (e.g., tweets and their replies).

However, news articles may have some eye-catching terms that aim to manipulate the readers' emotions. It was already demonstrated that false information has different emotional patterns and emotions play a key role in deceiving the reader [17]. Then the emotions were already used in other task like sentiment analysis for classification to distinguish between fake and non-fake reviews, these were very helpful and efficient on this task. Recently, the authors of the original Fakeflow model [9] have proposed a method that takes into account the affective changes in texts to detect fake news.They also combine two branches: one uses the embeddings and convolution and the other uses a Bidirectional-Gated Recurrent Units. Our implemented work is inspired by this study, but we use other datasets and a cross-domains configuration to evaluate the efficiency of the model.

3 Approach

3.1 Main Approach

We are implementing the entire Fakeflow [9] architecture from scratch inspired by the codes provided by the authors [1], we add some modifications because we had many errors when trying to run their code, due to deprecated packages versions of Tensorflow-Keras and due to missing files, so our structure would be clearer. We pretrain this model and search best hyperparameters.

However to make it original, we experiment this model by doing a test transferability, we want to know if the model is able to generalize on differents datasets covering a wide range of topics, thus we are using a combination of differents data for training and test sets, this would be described in the following part Experiments 4.

As shown in the Figure 1, the model is composed of two main submodules: the Topic information branch and the Affective information branch.

3.1.1 The Topic Information

First, we are using a pre-trained word2vec Google-News-300 embeddings to embed words to vectors through an embedding matrix. Then we uses a CNN (Convolutionnal neural network)[2] [18], which is

[1] `https://github.com/bilalghanem/fake_flow`

[2] `https://cs231n.github.io/convolutional-networks/`

mostly used in computer vision, it applies a convolution processes and max pooling to learn a new features representation of the input and highlight important words according to parameters like the limited size of the receptive field used, the number of filters. Here, the max pooling layer is a fixed filtering operation that calculates and propagates the maximum value of a given region, so useful for filtering important values. Next, we get a smaller representation by adding a fully connected layer on the output that connects every neuron in one layer to every neuron in another layer. It is the same as a traditional multi-layer perceptron neural network (MLP):

$$v_{topic} = f(W_a cnnv + b_a)$$

where W_a and b_a are the corresponding weight matrix and bias terms, and f is an activation function such as ReLU, tanh, etc. The following step consists to concatenate the representation vectors of each branch v_{topic} and v_{affect}, which aimed at capturing the affective information extracted from texts.

$$v_{concat} = v_{topic} \oplus v_{affect}$$

We merge all these representations because we want to capture their interaction then pass it into another fully connected layer.

$$v_{fc} = f(W_c v_{concat} + b_c)$$

Finally, we add a self-attention mechanism on v_{fc} to take into account the context in the news article, this allows us to put some weights on words and rank them differently by their importance according to the similarity with all neighboring words.

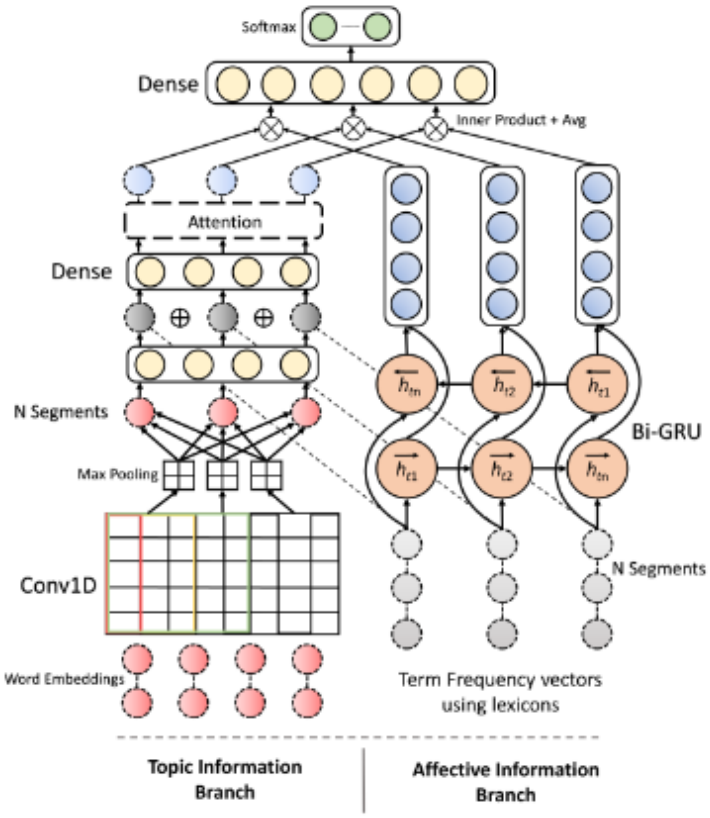

Figure 1: The architecture of the Fakeflow model

3.1.2 The Affective flow of Information

On the other hand, in the Affective flow of information branch (right in the figure 1), we have a term frequency representation of words by using known lexicons to extract the following features (see A.1): we have the **emotions features** to detect their change among articles' parts from NRC emotions

lexicon [3], we retrieve the **sentiment** from the text (positive and negative), we take cue words from the Moral Foundations Dictionary (divided in 10 categories) known as **morality**, we get the **imageability** which contains words rated by their degree of abstractness and we use **hyperbolic** to detect if words are high positive or negative. Last, we represent all theses features into the vector v_{affect} and apply a Bidirectional Gated Recurrent Units (bi-GRU) network to get more knowledge about the context of all features from both directions.

GRU is a recurrent neural network [19], [20], a useful mechanism for fixing the vanishing gradient problem. This problem occurs when the gradient becomes vanishingly small, which prevents the weight from changing its value. This mechanism uses two gates (A.2): the update gate controls information that flows into memory, and the reset gate controls the information that flows out of memory. These gates are two vectors that decide which information will get passed on to the output. And it is said bidirectional because the output layer can get information from past (backwards) and future (forward) states (from right to left and left to right) simultaneously .

Finally, for our final predictions, we combined the outputs of these two branches (the Topic Information branch and Affective Information branch) by applying a dot product before averaging the output matrix to get the final representation $v_{compact}$. We finish with another fully connected layer and a softmax to generate the label:

$$final = f(W_d v_{compact} + b_d)$$

3.2 LSTM model

Long Short Term Memory [21] is an other recurrent neural network architecture, which can keep track of arbitrary long-term dependencies in the input sequences. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The forget gate decides what information should be thrown away or kept. Information from the previous hidden state and information from the current input is passed through the sigmoid function.

3.3 Bi-GRU model

Bidirectional Gated Recurrent Unit[19],(A.2) is a recurrent neural network architecture, similar to LSTM but it uses only the update gate which controls information that flows into memory, and the reset gate which controls the information that flows out of memory. We choose this because it is a component of the Fakeflow model, and we want to compare the efficiency between using a single model (neural architecture) and a combination of methods like in FakeFlow.

3.4 Custom FakeFlow model

We use the same architecture of the FakeFlow model but we apply a different preprocessing step, in particular we are going to replace all contracted words by their original words like "isn't: is not","what's": "what is", because we expect that the affective flow does not use efficiently the contracted term. It might omit an important information between words like the subject and the verb in a sentence, while this contracted term could be used to emphasize a positive sentiment, an example could be the term "not" in the example "i don't hate you", which has a positive degree. Finally, we would change the Max pooling by a combination of Max pooling (for first convolutional layers) and an Average pooling (for the last one), then add a dropout after each fully connected layer.

4 Experiments

4.1 Data

First we train our model with the MultiSourceFake dataset [9], which contains 5,994 real and 5,403 fake news articles. The average document length (number of words) is 422 words. We use the same configuration as in the paper, we split the articles' text into N segments and set the maximum length of segments to 800 words, applying zero padding to the ones shorter than 800 words. We use the

[3] `https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm`

ReCOVery dataset [4] [8], which contains 2029 news (665 fake and 1364 true news) and Celebrity dataset [5] [14], with 460 news (both 240 for fake and true news). Then we split the datasets ReCOVery and Celebrity following the proportion: 80% for training and 20% for testing sets. Moreover, we experiment using differents combination of data from all datasets, as described in the Table 1 below. Last, we also add two additional experiments, we would verify if combined datasets from differents contexts for training are useful in other domain for testing.

Training	Testing
80% MultiSourceFake	20%MultiSourceFake
80% MultiSourceFake	20% ReCOVery
80% MultiSourceFake	20% Celebrity
80% MultiSourceFake +80% Celebrity	20%Celebrity
80% MultiSourceFake +80% ReCOVery	20%ReCOVery
80% MultiSourceFake +80% Celebrity	20%ReCOVery
80% MultiSourceFake +80% ReCOVery	20%Celebrity

Table 1: Table of proportion of data in train and test sets.

4.2 Data preprocessing

We have two datasets structured in csv format (MultiSourceFake and ReCOVery) but for the Celebrity dataset, we only have texts files in two directory called fake and legit. Therefore we had to iterate over each file to extract only the text and stored it to get a better structure of the text like a csv file. This has required more work compared to other datasets. Next we have preprocessed all these structured data by cleaning regex patterns, dropping useless columns and missing values, then we have put the text to lowercase. We have renamed columns in the same way for all datasets. Finally, we have reformatted the labels of text (True or Fake) in binary format and we have splitted each dataset according to the proportion 80/20 for training and test sets once, thus we will experiments using the same precomputed split for each combination for better consistency.

4.3 Evaluation method

We use the accuracy and F1-score (available in A.4) in order to evaluate our models for fake news detection. Accuracy measures how often the classifier correctly predicts. We can define it as the ratio of the number of correct predictions and the total number of predictions. It is a good measure when the target variable classes in the data are nearly balanced.

Next, the F1-score gives the harmonic mean of precision and recall, where Precision for a label is defined as the number of correctly predicted cases (true positives) divided by the number of predicted positives, whereas Recall is defined as the number of true positives(actual positive cases) divided by the total number of actual positives. These metrics are commonly used in binary classification task [7], [13], [14].

4.4 Experimental details

As described earlier in Data 4.1, we experimented by using different training and testing sets,we decided to use the same precomputed split for each combination for better consistency, and we first configured the Fakeflow model with default parameters (see Appendix A.3). We tuned various parameters (dropout, the size of the dense layers, activation functions, CNN filter sizes and their numbers, pooling size, size of the GRU layer, and the optimization function) for the search space using early stopping on the validation set. Since this is a new experiment, in order to make a comparison, we have implemented a Bi-GRU model with 100 units, dropout 0.6 and l1 regularizer (0.001) (this replaces the BERT model that I had planned to use at the beginning because of coding issues) then a LSTM model with same parameters with Tensorflow-Keras. These models are mostly used in many previous works [7], [14] as baselines. The Fakeflow model had an accuracy of 0.96 on

[4] `https://github.com/apurvamulay/ReCOVery`

[5] `https://lit.eecs.umich.edu/downloads.html#Fake%20News`

the MultiSourceFake dataset, but the model has never been experimented with these new datasets: ReCOVery (COVID-19 fake news)[8] and the Celebrity [14].

For the LSTM model, we first vectorized the text into a vector, we truncated and padded the input sequences so that they were all in the same length for modeling. The maximum length of the sequence is 800, similar to the implemented Fakeflow model, so sentences with less than 800 tokens were filled with [PAD] tokens. Then the implemented model is composed of an embedding layer learned through word2vec (which is also used in the Fakeflow model), followed by a LSTM layer with 100 units, we add a dropout(0.4) and a final fully connected layer with a softmax activation function. We also tried a LSTM version without embeddings but results were less significant compared to the previous version one, so we decided to omit this experiment.The implemented Bi-GRU follows the same process.

Furthermore, we used the cross-entropy loss, the Adam optimizer with their default parameters, and use an early stopping that stop the training after no improvement on validation loss within 4 epochs. For the rest of parameters, we started with 20 epochs and use a batch size of 32, we vary these hyperparameters. Most experiments were run on Google Collaboratory notebook environment.

4.5 Results

For simplicity, we refer to the differents combinations of data as experiments by using their letter associated on the left side of the table (e.q A for 80%MSF training and 20%MSF testing...). We reported the following scores obtained in the test sets according to the differents experiments described earlier (and according to the best results obtained in validation set):

	Data sets		FAKEFLOW	
	Training	Testing	Accuracy	F1score
A	80% MultiSourceFake	20%MultiSourceFake	**0.88**	**0.89**
B	80% MultiSourceFake	20% ReCOVery	0.63	0.75
C	80% MultiSourceFake	20% Celebrity	0.77	0.80
D	80% MultiSourceFake +80% Celebrity	20%Celebrity	0.76	0.78
E	80% MultiSourceFake +80% Celebrity	20%ReCOVery	0.60	0.44
F	80% MultiSourceFake +80% ReCOVery	20%ReCOVery	0.83	**0.89**
G	80% MultiSourceFake +80% ReCOVery	20%Celebrity	0.49	0.63

Table 2: Table of performances with Fakeflow model.

	Data sets		LSTM	
	Training	Testing	Accuracy	F1score
A	80% MultiSourceFake	20%MultiSourceFake	**0.83**	**0.81**
B	80% MultiSourceFake	20% ReCOVery	0.62	0.75
C	80% MultiSourceFake	20% Celebrity	0.44	0.58
D	80% MultiSourceFake +80% Celebrity	20%Celebrity	0.48	0.54
E	80% MultiSourceFake +80% Celebrity	20%ReCOVery	0.54	0.67
F	80% MultiSourceFake +80% ReCOVery	20%ReCOVery	0.60	0.70
G	80% MultiSourceFake +80% ReCOVery	20%Celebrity	0.49	0.58

Table 3: Table of performances with LSTM model.

Generally, we get higher performance with the Fakeflow model than baseline models with single training set and combination of training sets, even if we get lower results in a cross-domains contexts, this was what we expected at the beginning of this work,

However we can find few exceptions, the F1 score is similar to other baseline in the experiment **B** (=0.75 and LSTM) or worse in the experiment **E** (0.44vs0.67 LSTM). Then with accuracy, we observe the same case, equal in experiment **G** (0.49).The LSTM and Bi-GRU model have sometimes similar performances, but the scores are still worse compared to FakeFlow, so this shows that using a

	Data sets		Bi-GRU	
	Training	Testing	Accuracy	F1score
A	80% MultiSourceFake	20%MultiSourceFake	**0.86**	**0.88**
B	80% MultiSourceFake	20% ReCOVery	0.46	0.45
C	80% MultiSourceFake	20% Celebrity	0.54	0.37
D	80% MultiSourceFake +80% Celebrity	20%Celebrity	0.47	0.50
E	80% MultiSourceFake +80% Celebrity	20%ReCOVery	0.45	0.41
F	80% MultiSourceFake +80% ReCOVery	20%ReCOVery	0.61	0.27
G	80% MultiSourceFake +80% ReCOVery	20%Celebrity	0.44	0.32

Table 4: Table of performances with BI-GRU model.

	Data sets		Custom	
	Training	Testing	Accuracy	F1score
A	80% MultiSourceFake	20%MultiSourceFake	0.71	**0.83**
B	80% MultiSourceFake	20% ReCOVery	0.32	0.49
C	80% MultiSourceFake	20% Celebrity	0.54	0.70
D	80% MultiSourceFake +80% Celebrity	20%Celebrity	0.78	0.80
E	80% MultiSourceFake +80% Celebrity	20%ReCOVery	0.66	0.31
F	80% MultiSourceFake +80% ReCOVery	20%ReCOVery	**0.86**	0.80
G	80% MultiSourceFake +80% ReCOVery	20%Celebrity	0.60	0.61

Table 5: Table of performances with our custom model.

Bi-GRU model, part of the architecture of Fakeflow, is not sufficient to get high performance. Then our custom model seems to performs better than baselines models too.

Finally in term of accuracy, our custom model achieves the best performance only when we combine training data and is efficient on unseen data, but it is weaker when we have a single training data with MultiSourceFake (Figure 2). It seems to overcome all weakness of other datasets.

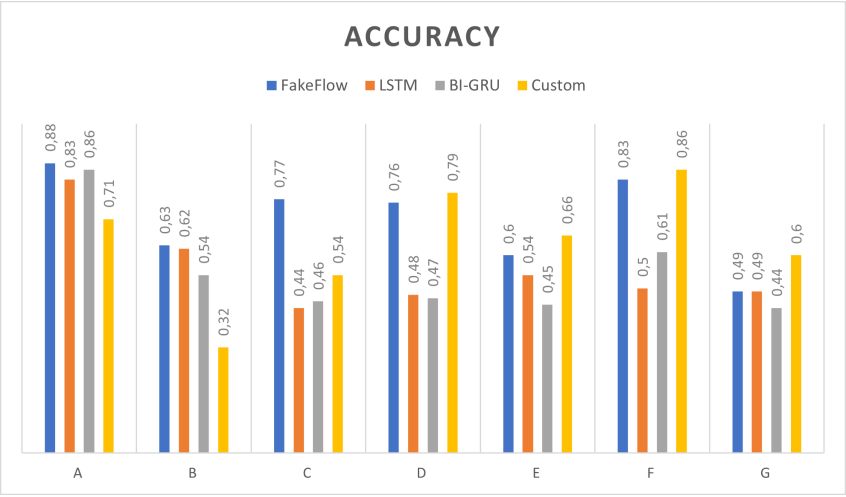

Figure 2: Overview of the accuracy with all experiments.

5 Analysis

We can see that we get lower scores when we combine different training sets and use either Celebrity or ReCOVery data on each side training/testing. Therefore we notice that we get worse performance

when we use data from the Celebrity dataset, this is most likely due to the short texts length in this dataset. There is a maximum of 480 words in a news article, which is less than other datasets. The length of text is closed to 1,750 words in ReCOVery and 12000 words in MultiSourceFake (Figure 3). Then this performance can also be explained by the fact that we used a maximum sentence of 800 words and used the default parameter of N=10 segments like in the original paper. This means that we keep only 800 words and we splits the text into 10 parts (this achieved best scores in previous work).

Consequently, we have less cue words, emotions in each segments in Celebrity dataset, which is composed of shorts texts from websites and magazines. All segments were composed of many null/pad tokens in order to reach the maximum text size of 800. Whereas MultiSourceFake and ReCOVery have more similarities (from online websites, or tweets, longer articles).

Furthermore, with our custom model, we have used all contracted terms and used Max pooling and Average pooling with dropout after a fully connected layer. Theses modifications have shown that the model is able to learn better the flow of affective information with these datasets that contain many contracted words, despite the different lengths of texts and their different topics. It learns better the affective sentiment with the structure of the contexts of words, because a missing term could change the affective meaning of a text (like the example in 3.4), but this was not learned with the original model and with our baselines models.

Otherwise, we might improve the performance on these datasets by changing the number of segments splits and the maximum sentence length in order to consider smaller texts lengths. Finally, it seems that the model has a better representation of the affective flow information in longer news articles than in shorter texts.

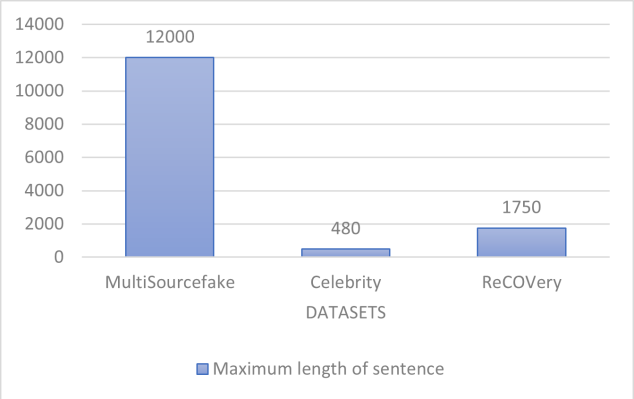

Figure 3: Comparison of the distribution of length of news between datasets

6 Conclusion

In this work, we develop, implement, and analyze the FakeFlow model which captures the flow of affective information in news. The project has demonstrated that the Fakeflow model is still efficient in a cross-domains contexts by using recent and diverse datasets including differents topics and it outperforms our baselines models. However, we show that it becomes weak when testing on shorter texts and covering unseen topics. We also demonstrated the useful combination of the two different branches (Convolution processs with self attention and BI-GRU) to capture useful information in news compared of using a single part of the model.Then we have proposed a modified architecture that is able to learn better when we combine training data and tested on unseen data, we have replaced all contracted terms with their original words and have used Max pooling and Average pooling to learn better features representations and use additional dropout. Nevertheless this work is still limited, we don't have finished to study the custom model with a single training set or implemented other baselines that are used in others works. It would be therefore interesting to compare to other common baselines and test the efficiency of the model by using multiples languages, since it only uses English language.

References

[1] H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election.

[2] E. Dreyfuss and I. Lapowsky. Facebook is changing news feed (again) to stop fake news. *Wired*, 2019.

[3] Mike Isaac Nick Wingfiled and Benner Kate. Google and facebook take aim at fake news sites. posted Nov. 14, 2016, Online.

[4] Kai Nakamura, Sharon Levy, and William Yang Wang. Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. In *the Association for Computational Linguistics(ACL)*, May 2020.

[5] Fan Yang, Arjun Mukherjee, and Eduard Dragut. Satirical news detection and analysis using attention mechanism and linguistic features. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017.

[6] Fabiana Zollo Fabio Petroni Antonio Scala Guido Caldarelli H Eugene Stanley Michela Del Vicario, Alessandro Bessi and Walter Quattrociocchi. The spreading of misinformation online. 2016.

[7] Daryna Dementieva and Alexander Panchenko. Cross-lingual evidence improves monolingual fake news detection. In *the Association for Computational Linguistics(ACL)*, August 2021.

[8] Xinyi Zhou, Apurva Mulay, Emilio Ferrara, and Reza Zafarani. Recovery: A multimodal repository for COVID-19 news credibility research. *CoRR*, abs/2006.05557, 2020.

[9] Bilal Ghanem, Simone Paolo Ponzetto, Paolo Rosso, and Francisco Rangel. FakeFlow: Fake news detection by modeling the flow of affective information. In *the Association for Computational Linguistics(ACL)*, April 2021.

[10] Hamid Karimi, Proteek Roy, Sari Saba-Sadiya, and Jiliang Tang. Multi-source multi-class fake news detection. In *the Association for Computational Linguistics(ACL)*, August 2018.

[11] Benjamin D. Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *CoRR*, 2017.

[12] Yla R. Tausczik and James W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 2010.

[13] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. In *the Association for Computational Linguistics(ACL)*, August 2018.

[14] Tanik Saikh, Arkadipta De, Asif Ekbal, and Pushpak Bhattacharyya. A deep learning approach for automatic detection of fake news. In *Proceedings of the 16th International Conference on Natural Language Processing*, International Institute of Information Technology, Hyderabad, India, 2019. NLP Association of India.

[15] Kai Shu, Suhang Wang, and Huan Liu. Exploiting tri-relationship for fake news detection. *CoRR*, 2017.

[16] Kai Shu, Deepak Mahudeswaran, and Huan Liu. FakeNewsTracker: a tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, 2019.

[17] Bilal Ghanem, Paolo Rosso, and Francisco M. Rangel Pardo. An emotional analysis of false information in social media and news articles. *CoRR*, 2019.

[18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent advances in convolutional neural networks. *CoRR*, 2015.

[19] Rahul Dey and Fathi M. Salem. Gate-variants of gated recurrent unit (GRU) neural networks. *CoRR*, 2017.

[20] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, 2014.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

[22] Saif Mohammad and Peter Turney. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, 2010.

[23] Haidt J. Nosek B. A. Graham, J. Liberals and conservatives rely on different sets of moral foundations. 2009.

[24] 1939 Coltheart, M. (Max) and 1939 Wilson, Michael John. MRC psycholinguistic database machine usable dictionary : expanded shorter oxford english dictionary entries / max coltheart and michael wilson. Oxford Text Archive.

[25] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. *CoRR*, 2016.

A Appendix (optional)

A.1 Features

- **Emotions** we use the NRC emotions lexicon [22] that contains 14K words labeled using the eight Plutchik's emotions (8 Features:anger, anticipation, disgust, fear, joy, sadness, surprise, trust)
- **Sentiment**: we extract the sentiment from the text, positive and negative, again using the NRC lexicon [22] (2 Features)
- **Morality**: we consider cue words from the Moral Foundations Dictionary2 [23] where words are assigned to one (or more) of the following categories: care, harm, fairness, unfairness (cheating), loyalty, betrayal, authority, subversion, sanctity and degradation (10 Features)
- **Imageability**: a list of words rated by their degree of abstractness and imageability. These words have been extracted from the MRC psycholinguistic database [24] and then using a supervised learning algorithm, the words have been annotated by the degrees of abstractness and imageability. The list contains 4,295 and 1,156 words rated by their degree of abstractness and imageability, respectively (2 Features)
- **Hyperbolic**: We use a list of 350 hyperbolic words [25], i.e., words with high positive or negative sentiment (e.g., terrifying, breathtakingly, soul-stirring, etc.). The authors extracted these eye-catching words from clickbaits news headlines (1 Feature).

A.2 Gated Recurrent Unit

A bidirectional Recurrent neural network can be represented with the Figure 4 from [6], then replacing every A and A' in the diagram with a gated recurrent unit with Figure 5 from [7] yields the bidirectional GRU. We use the following:

- $z_t = \sigma_g(W_s x_t + U_z h_{t-1} + b_z)$
- $r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_z)$
- $\hat{h_t} = \phi_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$
- $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot b_h)$

with x_t: input vector, h_t: output vector, $\hat{h_t}$: candidate activation vector, z_t: update gate vector, r_t: reset gate vector, W, U and b: parameter matrices and vector. σ_g: the original is a sigmoid function. ϕ_h: the original is a hyperbolic tangent. $\odot$ the Hadamard (elementwise) product.

[6] `http://colah.github.io/posts/2015-09-NN-Types-FP`

[7] `https://d2l.ai/chapter_recurrent-modern/gru.html`

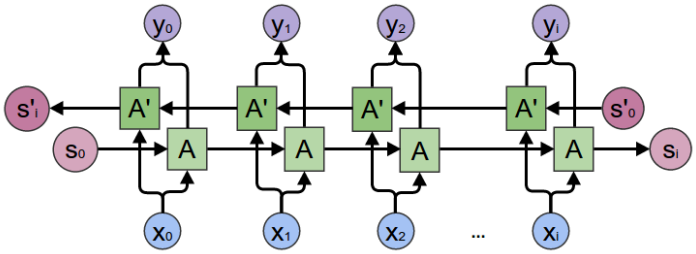

Figure 4: The architecture of a bidirectional RNN

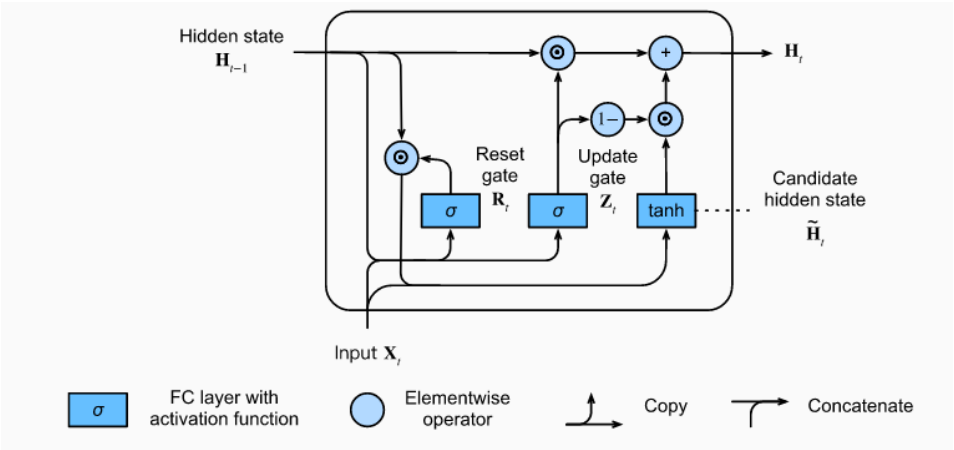

Figure 5: The architecture of a Gated Recurrent units.

A.3 Hyperparameters

For FakeFlow hyperparameters, we tune the following parameters with their correspondent search space:

- Dropout: random selection in the range [0.1,0.6],
- Dense layers: [8, 16, 32, 64, 128],
- Activation functions: [selu, relu, tanh, elu],
- CNN filters' sizes: [(2, 3, 4), (3, 4, 5), (4, 5,6), (3, 5), (2, 4), (4,), (5,), (3, 5, 7), (3, 6)],
- Numbers of CNN filters: [4, 8, 16, 32, 64,128],
- Pooling size: [2, 3],
- GRU units: [8, 16, 32, 64, 128],
- Optimization function: [adam, adadelta, rmsprop, sgd],

For the early stopping, we set the 'patience' parameter to 4 and we set the epochs number to 50. For the parameters selection, we use hyperopt library that randomly select different N combination of parameters (trials). We use a small value of N in all of our experiments to avoid overdrawn finetuning; we set N to 35.

A.4 Evaluation method

$$Precision = \frac{\textbf{True Positive}}{\textbf{True Positive+False Positive}}; Recall = \frac{\textbf{True Positive}}{\textbf{True Positive+False Negative}}$$

$$Accuracy = \frac{\textbf{Number of correctly classified}}{\textbf{Total Number of texts}}; F1score = 2 * \frac{\textbf{Precision*Recall}}{\textbf{Precision+Recall}}$$