AI for FakeNews Detection in Social Media

Stefano Ivancich
Department of Information Engineering (DEI)
University of Padova, Italy
Email: stefano.ivancich@studenti.unipd.it
Student ID: 1227846

Marco Zanetti
Department of Information Engineering (DEI)
University of Padova, Italy
Email: marco.zanetti.12@studenti.unipd.it
Student ID: 1220423

***Abstract*—Fake news are not a new thing, there are documents that prove their usage in politics at least since Cicerone 64 BC. According to the most recent behavioral psychology research fake news are today's winning communication strategy for politicians. In this paper we briefly present a few psychological and historical concepts that will be the foundation for any AI technique to solve the problem. Then we review the 2 most common techniques used today (2021). And finally we propose our detection schema which implements novel Machine Learning approaches in Natural Language Processing (NLP) to identify if a text is either a Fake news or not.**

***Index Terms*—AI, Fakenews, Social Networks, NLP**

I. Introduction

Firstly we introduce what fake news are and a few historic and psychological concepts, because those concepts will be, as you will read later in the paper, the key for making a successful AI detector.

A. What are fake news

Fake news is defined as: false or misleading information presented as news It often has the aim of damaging the reputation of a person or entity, or making money through advertising revenue [1]. The Oxford Dictionary, in 2016 announced as a international word of the year: "Post Truth", that is the series of events in which objective facts are less influential on public opinion than the emotions that are aroused by the way these are told. For the Collins, the most authoritative English dictionary, the representative word of 2017 was Fake News. Authors in [2], analyzed how fake news disseminate thought social media, in particular they discovered that they spread very rapidly, with a peak in the first 2 hours.

B. It's not a recent phenomenon

These techniques of conquest of consensus through the diffusion of false news have not been invented by the politicians of the last years, but there have always been. For example, in 64 BC Marcus Tullius Cicero was a candidate for the office of Consul of the Roman Republic. His brother Quintus, to help him win the election, wrote the Commentariolum Petitionis [3], a manual for the election campaign. Rather than political advice, it deals with communication strategies, such as that of always answering "yes" to the requests of the people even if you knew in advance that you would not be able to satisfy them. Among these advices we also read: "*Take care that a suspicion arises towards your enemies, appropriate to their behavior or of guilt, or of luxury, or of squandering*". Not a truth then, but a likely accusation, one that would ruin the reputation of the opponent. In other words, Cicero's brother was saying that defamatory Fake News in an election campaign is not just permissible, indeed it is useful practice. Even then, it didn't matter to dismantle the rival's arguments, but to cast suspicion on them through fake news.

C. Psychology behind fake news

According to Psychology decision making theory [4] most of the decisions that people make are not made with logic, but are made with emotions. Then the logic is used to justify the irrational choice that the individual has made. Users on social media tend to form groups containing like-minded people where they then polarize their opinions, resulting in an echo chamber effect. The echo chamber effect facilitates the process by which people consume and believe fake news due to the following psychological factors [5]: (1) social credibility, which means people are more likely to perceive a source as credible if others perceive the source is credible, especially when there is not enough information available to access the truthfulness of the source; and (2) frequency heuristic, which means that consumers may naturally favor information they hear frequently, even if it is fake news.

All these facts:

- Fake News on Social Networks spread rapidly, with a peak in the first 2 hours.
- There is a strong economical and political interest in their use . (E.g., Cambridge Analytica, Trump elections, Russian Ads, ...)
- Manual techniques such as debunking and fact checking sites are slow, not applicable in large scale and easily stoppable.
- Social Network bubbles make people isolate themselves in their beliefs and prejudices that can be easily manipulated
- The average person doesn't have the time, want, resources or intellectual abilities to verify the sources and study deeply a topic, they just take each statement they read as true.
- Post Truth and Fake News are the winning communication strategy of today's politics.

highlight the urgent need for an automatic system for detecting Fake News and hoaxes on Social Networks platforms.

In this work, we focus on detecting Fake news given social media posts trough the use of Machine Learning systems; in particular, we explore the classification of 'tweets' (short text of 240 characters posted on the social network Twitter) as either True or Fake news.

In this paper we will:

- present the most common practices to tackle the fake news problem in social media today (2021);
- Apply different Machine Learning approaches to see how current advances in AI allows machine to perform in the Fake News detection task.

The report is structured as follows: in Section II we present the current state of the art in Fake news detection, in Section III we show the data used in this project and in Section IV how we processed it so it becomes usable for a Machine Learning system. In Section V we describe the various architectures we considered, in Section VI we report their results and finally in Section VII we make some extra considerations on future developments and possible improvements.

II. Related Work

We start by describing the 2 most common approaches to fake news detection used in social media today (2021), those will be the foundation of our work.

A. Detection by likes on the social media post

The authors in [6] uses a dataset consisting of 15,500 Facebook posts of a list of selected Facebook pages and 909,236 users. Those Facebook pages are divided into two categories: science news sources and conspiracy news sources. Given each pair of pages, they studied how many users have liked at least one post from one page and one post from the other page. The result is that hoax pages have more users in common with other hoax pages than with non hoax pages. So they tried to classify the posts as hoaxes or non-hoaxes on the basis of the users who "liked" them. They experimented a novel adaptation of boolean crowdsourcing algorithms [7] reaching an accuracy of 99%.

B. Detection by coherence between article title and its content

In recent years, newspapers are financially supported mainly by online advertisement, so they tend to write "catchy" titles. A common technique used by politicians in social media to deceive users, is to write a post with some false information and link the article with a "catchy" title to pretend endorsement. But as we said before, the average user doesn't even click on the article to delve the argument, or doesn't read it fully. So the idea proposed in [8] is that an article linked in a social media post can be automatically opened and processed before letting the post be visible to all users. They used a dataset of 49,972 pairs of titles and body text, each with a label: unrelated, agree, disagree, or discuss. In total there are 1683 articles and 49972 titles. They used bag-of-words [9] followed by a three-layer perceptron (BoW MLP) reaching a test accuracy of 89%.

III. Dataset

Finding data suitable to develop Artificial Intelligence application is always a hard task. While there's an abundance of news (both true and fake) on the web, collecting and labelling them proves to be a time consuming task. In our project, we decided to use the Covid19 Fake News dataset published by Patwa et al [10], which contains around 10.000 tweets about the pandemic. We found this particular application a fitting choice for our project given the recent spread of misinformation around the Covid-19 pandemic. Since data are labeled either 'true' or 'fake', we say that we are in a binary classification setting: there are two possible labels our input data can be associated with, and our model's objective is to correctly predict which one it will be given novel data. The dataset is balanced, that is, the number of Fake news and True news it's even. This is a desirable property for classification task as it makes harder for our models to solely rely on statistical properties of the data for prediction and avoid unbalanced class predictions. The dataset has been divided as follows:

- A training set, used to train our models, composed of 6420 tweets;
- A validation set, used to tune hyperparameters and validate our models, composed of 2140 tweets;
- A test set, used to assess the performance of our models on unseen data, composed of 2140 tweets.

IV. Processing Pipeline

A. Preprocessing

Before being used by a Machine learning model, textual data need to go trough a pipeline that remove noise and uninformative words and characters. We adopted the procedure in fig. 1. In particular, the text goes trough the following steps:

1) **Lowercase text:** We convert the whole text to lowercase (e.g 'A' becomes 'a'). This prevent our models to differentiate between lowercase and uppercase letters;
2) **Remove URLs:** We remove all hyperlinks in the text;
3) **Remove stopwords:** We remove all stopwords (e.g 'and', 'or', 'the');
4) **Remove emojis:** We remove all emojis;
5) **Lemmatization:** We lemmatize the text (e.g 'am', 'are', 'is' become 'be');
6) **Remove punctuation:** We remove punctuations (e.g '.', '#', '!').

Those steps transform the text in a simplified and more predictable form, and greatly improve performances of our models.

B. Tokenization and word embedding

While the text have been simplified trough the previous procedure, it's still presented in an unreadable format for our machine. The last step before our model can understand our data is called tokenization. Trough text tokenization we transform each word (or sets of words called n-grams) into a token. Then we match each token with an embedded vector

trough a procedure called word embedding. This learned vector can express meaning trough distance in the vector space (i.e 'king' will be closer to 'queen' or 'man' than to 'plane'). Embeddings work best when trained on a large amount of data, which can be time and resources costly: for this reason the use of pretrained embeddings is common in NLP task. In our project, we use the embeddings provided by FastText [11] for our neural models, and the HuggingFace's embeddings [12] for our attention model.

V. Learning Framework

Originally, statistical models were used to process textual data. However, with the rise of deep learning in the last decade, neural networks architectures have dominated the domain. Since text is processed in a sequential way, timeseries models got a lot of success: architectures such as Recurrent neural Networks and Attention models are capable of processing both current and past inputs, which enables them to learnt he underlying structure of the text. In our project, we experimented with the following architectures:

- **1 Dimensional Convolutional Neural Network.** 1DCNN, first introduced by Yann Lecun and Yoshua Bengio in 1997 [13], are a particular variation of convolutional Neural network that produces 1D features map per kernel. Each kernel will learn to detect a short sequential patterns. 1DCNN can also be combined together with RNNs;
- **Long Short Term Memory Neworks:** LSTM were first proposed by Schmidhuber and Hochreiter in 1997 [14], as a mean of solving the gradient vanishing problems that occurs for deep recurrent networks. LSTM make use of so-called gates that allows the LSTM cell what to store in the long-term state, what to throw away and what to read from it;
- **BERT:** After the relase of the groundbreaking paper 'attention is all you need' [15] in 2017, which introduce for the first time the tranformer architecture, attention based models have dominated the natural language processing scene. The BERT architecture that we've used in our project was proposed by Devlin et al. [16]. In the paper. they showed that it's possible to pretrain a model and then use it in a different array of applications by simply finetuning the architecture for the current task. This allows to skip the time-consuming and resource-intensive training procedure of a transformer.

Both our LSTM and 1DCNN based models are implemented using the Tensorflow framework [17]. For our BERT model, we use HuggingFace's pre-trained implementation of DistilBERT, a smaller and faster version of BERT (more suitable to be used on our machine). For reproducibility purposed, we share the spec of the machine where we run our tests: Intel Core I7 10710U, 16gb RAM, SSD and GTX 1650 maxQ GPU. In particular, GPU may be a technical bottleneck to train heavier models (such as DistiBERT).

VI. Results

In this section we share the architectures and the parameters used in our models and the results obtained. We experimented with 4 different models: a 1DCNN, a LSTM, a combination of 1DCNN and LSTM, and a fine-tuned DistilBERT.

A. Models architecture

Our 1DCNN network is composed by an initial embedding layer (to encode words in a suitable way for our network), 3 one dimensional convolutional layers (which can extract useful information from the data trough filters) each followed by a Max Pooling layer (to reduce complexity), and finally 3 dense layers. The output layer consist in a single neuron with sigmoid activation function (which allows for binary output). The numbers of the filers are respectively [128, 72, 64] while the kernel size is 5. Dropout is implemented in the dense layers with probability 0.5, and relu activation function is used. We used Adam optimizer with learning rate 0.0001 and binary cross entropy as loss function. Early stopping is implemented to avoid overfitting.
Our LSTM network consist by an initial embedding layer followed by four LSTM layers with size 100, followed by the output layer consist in a single neuron with sigmoid activation function. Each LSTM layers implements both dropout and recurrent dropout with a probability of 0.2 to avoid overfitting. Early stopping is used once more here. We used Adam optimizer with learning rate 0.001 and binary cross entropy as loss function.
The combined 1DCNN + LSTM model consists in the usual embedding layer followed by two 1 dimensional convolutional layers of size (128, 64) and kernel size 5, each followed by a max pooling layer. Then, 3 LSTM layers of size 100 are stacked, and finally the output layer. LSTM layers implements both dropout and recurrent dropout. The optimizer is the same as the aforementioned model.
The last model we implemented is the DistilBERT: this is composed by a pre-trained DistilBERT network with 4 dense layers of size [200, 150, 100, 50] with dropout and a final output layer. We use Adam optimizer once more with learning rate 0.00003 and binary focal loss. Here instead of having an embedding layer we use the pre-trained DistilBERT tokenizer to convert our data prior to the training phase. The training regime for this model is done in two phases: in the first one we freeze all DistilBERT layers and train solely our stacked dense layers. In the second pass, we unfreeze all of DistilBERT's layers and then train the whole model for 10 epochs.

B. Model Performances

The performances of each model are showed in table 1. Firstly, we note that all of our model drastically improve from the baseline (random guessing if a tweet is true or fake). This underline that our models are actually learning how to differentiate between true and fake news, and not just relying on pure statistical property of the dataset. We can see that there exists a clear trade-off between accuracy and speed. Lighter model, such as 1DCNN, are extremely

Fig. 1: Text pre-processing pipeline

Model	Accuracy (%)	# of Params	Speed (s)
Baseline	50.0	/	/
1DCNN	87.8	4,804,087	**0.87**
LSTM	91.9	4,995,001	34.76
1DCNN + LSTM	92.3	5,061,681	8.48
DistilBERT	**94.7**	66,363,649	21.67

TABLE 1: model performance table.

faster due to their unique property of reducing the data complexity trough the use of kernels, but fail to fully grasp the complexity behind textual data. On the other hand, models more suitable for NLP tasks such as LSTM and BERT can obtain better performances but are slower both to train and execute. It's interesting how by combining 1DCNN and LSTM we obtain good performance while maintaining the speed of the model quite high. We can also see how current state-of-the-art model such as DistilBERT outperform easily previous models, scoring as high as 94.7% of accuracy. This however is balanced by the extreme complexity of the model: training the model become infeasible without expensive hardware and therefore must be loaded pre-trained and then fine-tuned. While the accuracy is extremely good, this can be a big downside for smaller applications where lighter models would be preferable.

VII. Concluding Remarks

In this paper we introduced the problem of fake news in social network, presented a few important historic and psychological concepts behind them and then we reviewed the 2 most common practices to tackle the problem today, namely, Detection by likes on the social media post and Detection by coherence between article title and its content. Finally we proposed our detection schema which uses textual information and recent Machine Learning approaches in the Natural Language Processing domain. By applying different models we've showed how more recent approaches can outperform older method, at the cost of increased complexity. We've showed that it's possible with relatively good performance to correctly classify Fake News on tweets trough the use of automated process, given a labelled dataset. We expect therefore to see such applications to become diffuse on social media like Facebook and Twitter in the following years and have a major influence to reduce the spread of misinformation.

As a final note we want to add that since the computation can be done in high capable servers, the best solution would be use a combination of the discussed techniques, since each of them are specialized in a particular aspect.

VIII. Acknowledgment

This work is presented for the final grade of the "Artificial Intelligence" course, held by prof. Maria Silvia Pini at the University of Padova in 2021.

References

[1] Wikipedia contributors, "Fake news," 2021. [Online; accessed 16-June-2021].

[2] M. D. Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, "The spreading of misinformation online," *Proceedings of the National Academy of Sciences*, vol. 113, pp. 554–559, Jan. 2016.

[3] Quintus Tullius Cicero, "Commentariolum petitionis," 2021. [Online; accessed 16-June-2021].

[4] J. S. Lerner, Y. Li, P. Valdesolo, and K. S. Kassam, "Emotion and decision making," *Annual Review of Psychology*, vol. 66, no. 1, pp. 799–823, 2015. PMID: 25251484.

[5] M. M. Christopher Paul.

[6] E. Tacchini, G. Ballarin, M. L. D. Vedova, S. Moret, and L. de Alfaro, "Some like it hoax: Automated fake news detection in social networks," 2017.

[7] L. de Alfaro, V. Polychronopoulos, and M. Shavlovsky, "Reliable aggregation of boolean crowdsourced tasks," in *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2015, November 8-11, 2015, San Diego, California, USA* (E. Gerber and P. Ipeirotis, eds.), pp. 42–51, AAAI Press, 2015.

[8] R. Davis and C. Proctor, "Fake news, real consequences: Recruiting neural networks for the fight against fake news."

[9] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 43–52, 12 2010.

[10] P. Patwa, M. Bhardwaj, V. Guptha, G. Kumari, S. Sharma, S. PYKL, A. Das, A. Ekbal, M. S. Akhtar, and T. Chakraborty, "Overview of CONSTRAINT 2021 shared tasks: Detecting english COVID-19 fake news and hindi hostile posts," in *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, pp. 42–53, Springer International Publishing, 2021.

[11] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," 12 2017.

[12] 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. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. 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), pp. 38–45, Association for Computational Linguistics, Oct. 2020.

[13] Y. Bengio and Y. Lecun, "Convolutional networks for images, speech, and time-series," 11 1997.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[17] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.