A Survey on COVID-19 related Fake News Detection using Machine Learning Models

Rayees Ahmad Dar[1], Dr. Rana Hashmy[2]

[1,2]*University of Kashmir, Hazratbal, Srinagar 190006, J&K, India*

Abstract

Uncensored data explosion on social media platforms has on the one hand impelled fast and easy dissemination of news and facts, but at the same time poses serious threats because of its highly unreliable nature. Misinformation and disinformation are mainly prevalent at the time some important event is happening that people are curious about e.g., elections or something untoward happens like the COVID-19 pandemic. Because of the unprecedented nature of these events, people are susceptible to these bogus and potentially hazardous claims and articles. Therefore, we need an early detection mechanism to stop the spread of intentionally and unintentionally written fake news or claims.

Past research has suggested various models based on machine learning, deep learning and pretrained language models to detect false news over the years. This research piece will try to assess the effectiveness of various relevant methods on the task of detecting fake news and false claims related to COVID-19 pandemic in this research. We will be using the combined corpus of two largest datasets available. We explore various pertained language models in addition to deep learning and conventional machine learning approaches and compare their performance. We find that RoBERTa in particular and Bert-based models in general outperform all other models. We believe this piece of research will help the research community a lot in exploring the said domain further.

Keywords [1]

fake news detection, social media fake news, misinformation, COVID-19, machine learning, language models

1. Introduction

Fake news can broadly be defined as “A news article or message published and propagated through media, carrying false information regardless of the means and motives behind it” [1-8]. Fake news gets to its worst at the time of some pandemic as people tend to believe false information in these chaotic situations, as there is scarcity of knowledge and research about it. This gets even worse when it propagates on some social media platforms due to its unauthenticated nature. This could inflict damage on both individual and societal levels. Thus, early detection and stopping of these posts becomes crucial on social media platforms. At the time of COVID-19 breakout, certain infringe elements simultaneously exaggerated the uncertainty and social disruption by spreading false information mostly on social media platforms. This is mostly related to the disease itself in addition to vaccines, medication, mask usage, etc.2 Hence, it becomes equally important to mitigate this infodemic in addition to fighting the pandemic itself. Different machine learning methods have been employed for this purpose.

[1] MoMLeT+DS 2023: 5th International Workshop on Modern Machine Learning Technologies and Data Science, June 3, 2023, Lviv, Ukraine

EMAIL: Rayees.csscholar@kashmiruniversity.net (R. Ahmad); ranahashmy@gmail.com (Dr. R. Hashmy);
ORCID: 0000-0002-4424-6593 (R. Ahmad);

We have tried to analyze the performance of various relevant classical machine learning, deep learning and most importantly the pre-trained language models on the combined COVID-19 dataset, which we accumulated from various already available datasets.

As the labeled data in this case is sparse, Bert based models and language models perform better comparatively. In this comparative study, we will try to analyze how good these approaches perform on the said datasets. Because of the distinct nature of corona (COVID-19) related fake news, we feel the behavior of these models needs to be observed separately in this domain of fake news

2. Related Work

Due to the omnipresence of Internet and its ease of access, social media has become an integral part of our lives. However, its unauthenticated nature poses a serious threat simultaneously. A large number of machines leaning based approaches have been proposed for the automatic detection of false news and claims.

Conventional machine learning based approaches when utilized for the purpose of fake news detection have yielded good results. Reis et al. [9] approached this problem as a binary classification task; various syntactic and semantic features are extracted through feature engineering and later passed to conventional ML classifiers like K-Nearest Neighbor (KNN), Naïve Bayes, Random Forest (RF), XGBOOST (XGB), Support Vector Machine (SVM) for training and classification. Out of these XGB and RF yielded promising results.

[10] Evaluated deep learning methods for fake news detection task. They trained various DL models on COVID-19 fake news detection dataset [11] from Contraint@AAAI 2021. They analyzed various deep learning models like LSTM, CNN, HAN, bi-LSTM+attention, DistilBERT and BERT-base. They treated this problem to be a binary classification task. They mainly focused on news content (Text). They have tried to give the pre-trained BERT and DistilBERT some context by pre-training them on the tweet corpus related to covid-19 that has proved to increase performance in comparison to the models which are trained on the dataset only. COVID-Twitter-BERT when ensembled with BERT-cased model approach outperforms other approaches. Furthermore, HAN outperformed other non-transformer-based models.

In [12], authors trained an ensemble of Bi-LSTM and BI-GRU-dense models on the LIAR [13] dataset and classified the news items as fake or real. The outputs from these two models averaged out to get a single value as output. After experimentations, the results of the proposed model proved to perform better when compared to other studies, which used the LIAR [13] dataset for fake-news detection.

Transformer [14] architecture is the base for majority of the state-of-the-art approaches for fake news detection currently. As these models employ self-attention technique wherein every word in a sentence is weighed on the basis of its significance and are pretrained on a large collection of data, they have proven to be superior to previous non-transformer-based models. One among the transformer based pretrained language models from Google is BERT [15] which has 345 million trainable parameters ($\text{BERT}_{\text{LARGE}}$) and is state-of-the-art architecture for various downstream jobs like text classification. [16] Proposed fakeBERT, a BERT-based deep learning approach combining CNN with the BERT, which helps in reducing ambiguity. The authors of [17] proposed an ensemble of (BERT, ALBERT, and XLNET) and fine-tuned and later tested this model on the Constraint AI 2021 Fake News Detection dataset [11]. A variation of BERT: CT-BERT (COVID-Twitter-BERT) [18] was created by pre-training the BERT model on a large collection of tweets related to COVID-19 and has shown promising outcomes for fake-news detection on Corona related news. Various language models (XLNet, ERNIE 2.0, XLMRoBERT , DeBERTa, RoBERTa, and ELECTRA) were ensembled by [19] for COVID related fake-news detection. Apart from the features based on news content, they incorporated social context-based features e.g., authors, source, username, and URL.

3. Dataset Description

The Following two datasets were used in this comparative study:

3.1. COVID-19-FNIR DATASET:

COVID19-FNIR DATASET [20] (COVID-19 Fake News Infodemic Research Dataset) Consists of true and fake news as separate files with a total of 7588 items which are class balanced (49.99% as real and 50.01% as fake). The fake news items have been collected from Polynter and the true items have been collected from authentic Twitter handles of news publishers. The dataset consists of various columns such as Text, Date, Region, Country, Explanation, Origin, Label, etc. but we will be using Text and the Label columns only in this study.

3.2. COVID19 Fake News Dataset:

COVID-19-fake-news detection [11] Dataset was published as a collection of various articles and posts related to COVID19 from social media with fake and real labels. Real news items in the dataset have been gathered from various verified news sources and the false items from some fact checking platforms like NewsChecker, PolitiFact etc., which were verified to be false. This dataset originally comprises of 10700 social media posts with a vocabulary size of 37505. 52.34% of news items in this dataset are real and the remaining 47.66% are fake and hence is balanced class wise.

Finally, we have created a combined corpus from these two datasets for this comparative study. We renamed various columns to make them uniform, and also replaced label values 'fake' with 0 and ‘real’ with 1 in the first dataset (Section 3.1) to match these values with the label field of the other dataset (Section 3.2). Finally, the combined corpus has 18288 news items in total, of which 51% comprises of real samples and the remaining 49% are fake samples. Finally, we split this combined dataset into train: test: validation with the ratio of 8:1:1 as shown in figure 1.

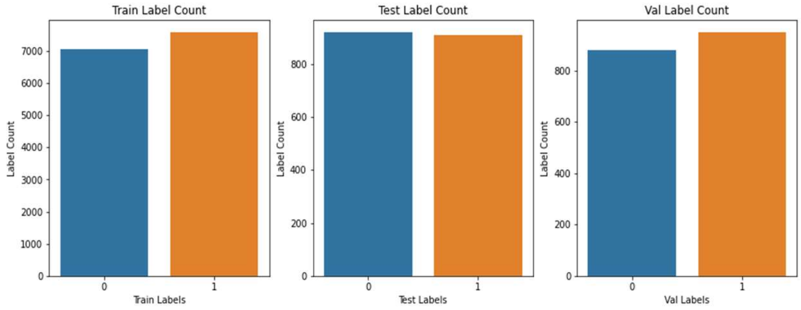

Figure 1: Combined corpus split of real and false news distribution

4. Methods

In this research, we approached COVID-19 fake-news detection task as a classification problem with two classes (fake and real) wherein news pieces are classified as real or fake.

4.1. Data preprocessing

We do some initial preprocessing of the raw text before actual processing. We eliminate unnecessary URLs, HTML tags extra spaces, stop words and special characters from the text, which is fed into the models after tokenization.

4.2. Studied features

Word level, n-gram level TF-IDF features, pre-trained fastText[21] which is an extension of word2vec model and represents words as n-gram of characters, Glove[22] embeddings which is a unsupervised learning algorithm which learns the word embedding based on the observation that word-word co-occurrence probability ratios have the potential to encode some meaning, and word embedding features from language models like BERT [15] pre-trained on English Wikipedia with 2500M words and BooksCorpus with 800M words. 128-dimensional words embedding of BERT are used in this study. We have used Bert word embeddings because these capture the contextual meaning and produce high-quality feature inputs, which are dynamically informed by the words around them. These pre-trained embeddings were used with deep learning models like CNN and off course for respective language models like roBERTa embedding for roBERTa model. We use the embeddings from their corresponding tokenizers. We further experimented with combining conventional machine learning models with word embeddings from finetuned BERT and the fastText embeddings. The TF-IDF features outperformed other embeddings on the majority of analyzed traditional ML models.

4.3. Studied models

We analyzed various models centered on classical machine learning, deep learning, and pretrained language model approaches:

4.3.1. Conventional Machine Learning methods

Traditional NLP approaches like Logistic Regression, Random Forest (RF), K-nearest neighbors (KNN), Support Vector Machines (SVM), Multinomial Naïve Bayes, XG-Boost and Decision Trees (DT) have been studied in this study.

We analyzed the results from these approaches using TF-IDF, fastText word vectors as well as the Bert word embeddings. We find that the TF-IDF features proved better than the fastText as well as the Bert embeddings when used on traditional Machine learning approaches. We used SelectKBest of sklearn to select k (k=1200) best features to be used for the training purpose.

Finally, out of these analyzed models, we found that the SVM combined with TF-IDF features performed best on test data of the combined corpus showing an accuracy of 84.29 %.

4.3.2. Deep learning models

CNN: We have used a one-dimensional convolutional model with two layers containing 128 filters of filter size 5. Embedding layer is the first layer. The model is initialized with pre-trained Glove embedding of dimension 300 and also experimented with BERT embedding and compared their performance. The outputs of con1D layers are passed through the ReLu activation function. This function outputs 0 for negative values and outputs positive values as it is.

A max-pooling layer of pool size 2 is stacked after each convolutional layer to reduce the size of model. The outputs from these maxpooled layers is concatenated into a single layer before being fed into a dropout layer (dropout =0.4). During compilation, learning rate of the adam optimizer is set to 0.0001. Finally, as we are dealing with a Binary classification problem, we pass these final outputs to a dense layer (1 unit) and sigmoid as the activation function.

LSTM: We initialized the embedding layer with pre-trained Glove embedding of size 300. The LSTM layer's output dimension was set to 300 and finally, we add a dropout layer (dropout = 0.6) before feeding the output into a sigmoid activated dense layer for classification. We also experimented with initializing the embedding layer with Bert embedding of length 128. The model is compiled using Adam optimizer. Training of the model was done for 10 epochs and 64 was set as the batch-size.

CNN+LSTM: We analyzed the performance of a hybrid model consisting of CNN layer and a LSTM layer on top of it. We define a CNN model as described above and before passing the outputs to the final dense layer, we pass it through an LSTM layer of output dimension 300.

4.3.3. Pretrained language models

Here, we describe the experimental setup of two advanced language models used in our study.

DistilBERT. DistilBert [23] was built based on the knowledge distillation compression technique, the knowledge is distilled from the BERT base model using almost only half of its parameters while retaining 95% of the BERT's performance on its benchmark GLUE. The token type embeddings and the pooler were removed from the original architecture by its creators to make it lighter. DistilBERT is less resource intensive while retaining the performance closer to the BERT model and is thus suited for production-level usage. We add a sigmoid activated dense layer as a classification head to the distilBERT model.

RoBerta: RoBERTa **(**Robustly optimized BERT pretraining approach**)** **[**24**],** alters key hyperparameters of the BERT model like removing its next-sentence-prediction(NSP) objective and training with higher learning rate on relatively larger mini batches, which proved to significantly improve the performance . In RoBERTa byte pair encoding (BPE) is used as a tokenization algorithm instead of BERT's word piece tokenization. NSP objective is removed for a better training strategy. A dropout of 0.4 is applied to the output from the transformer before being fed into a classification head, which is a sigmoid activated dense layer.

Corresponding word embedding is used with respective pre-trained language models i:e distilBERT embeddings for DistilBert and RoBERTa embedding for RoBerta. These models were trained for 18 epochs and 128 was set as the batch-size. In order to avoid overfitting, we used early stopping (Validation loss as the metric). Finally, the models were trained with Adam optimizer setting learning rate = 1e-4, b1 = .8, b2 = .898 and epsilon set to 1e-7. As a loss function, we used sparse-categorical-cross-entropy. The experiments were performed on Tesla P100-PCIE -16GB GPU provided by Kaggle.

5. Evaluation matrices

We utilized the following evaluation metrics to measure the performance of these models:

Accuracy: The percentage of correctly classified tweets, calculated as:

accuracy= (Tps' + Tns') / (Tps' + Tns' + Fps' + Fns')

Precision: The percentage of true positive predictions out of all positive predictions, calculated as:

precision= Tps' / (Tps' + Fps')" .

Recall: The percentage of true positive predictions out of all actual positive tweets, calculated as:

recall= Tps' / (Tps' + Fns')

F1 score: The harmonic average of precision and recall that provides a singular performance metric for the model, calculated as:

F1-score= 2 * (precision * recall) / (precision + recall

Here, Tps' denotes the number of true positive predictions, which represents the number of fake tweets correctly identified as fake by the model. Similarly, Tns' represents the number of true negative predictions, which corresponds to the number of real tweets correctly identified as real by the model. Fps' refers to the number of false positive predictions, indicating the number of real tweets incorrectly identified as fake by the model, and Fns' denotes the number of false negative predictions, representing the number of fake tweets incorrectly identified as real by the model.

6. Experiments and Results

Much research has been done on automatic detection of fake-news on social media using various machine learning and deep learning models. Some of them focused on comparative analyses of these models on fake news datasets. Keeping in view the unique nature of fake news about COVID19, it seems quite worthwhile to investigate various machine-learning models on COVID19 datasets. We will try to address this concern in this study. We first analyze different conventional machine-learning and deep-learning models with the combined corpus of COVID-19 related datasets and importantly evaluate some pre-trained language models as well.

Further, we also analyze the efficacy of various classical machine-learning and deep-learning based approaches on different embedding vectors. We analyze the performance of traditional machine learning approaches using three different embeddings (fastText, Glove, BERT) and deep learning approaches using Glove and Bert.

Table 1

Performance of traditional machine learning models

Method	Accuracy-Score (in %ge)			Precision (in %ge)			Recall (in %ge)			F1 Score (in %ge)		
	BERT Embedding	**FastText**	**TFIDF**	**BERT Embedding**	**FastText**	**TFIDF**	**BERT Embedding**	**FastText**	**TFIDF**	**BERT Embedding**	**FastText**	**TFIDF**
Logistic regression	**81.84**	82.34	83.78	**83.19**	**79.91**	**84.36**	81.79	**85.68**	**83.96**	82.48	**82.69**	**84.16**
Multinomial Naive Bayes	81.73	78.73	82.17	82.94	73.40	79.38	81.90	89.12	87.80	82.42	80.50	83.38
K-Nearest Neighbor	81.79	76.10	78.76	82.41	73.57	80.28	**82.84**	80.35	77.28	82.62	76.81	78.75
XG-Boost	**81.84**	**82.68**	81.78	82.63	79.79	82.77	82.63	86.79	81.10	82.63	83.14	81.93
Random Forest	81.62	81.30	81.78	82.56	79.57	81.78	82.21	83.46	85.01	**82.38**	81.47	83.36
Decision Tree	81.57	71.35	78.16	83.03	73.14	78.16	81.38	70.03	77.75	82.19	73.14	77.95
Support Vector Machine	81.73	82.39	**84.29**	82.59	79.57	83.83	**82.**42	86.45	**85.70**	82.51	82.87	**84.75**

Table 2

Performance of deep learning approaches

Method	Accuracy (in %ge)		Precision (in %ge)		Recall (in %ge)		F1-Score (in %ge)	
	GLoVE	**BERT**	**GLOVE**	**BERT**	**GLOVE**	**BERT**	**GLOVE**	**BERT**
CNN	84.53	85	**85.40**	83.31	79.83	85.40	83.02	84.34
LSTM	85	85.5	84	85	84	85.8	84	85.39
CNN+LSTM	**86**	**87.3**	85	**86**	**85**	**86.40**	**85**	**86.50**

Table 3
Performance of pre-trained language models

Method	Accuracy	Precision	Recall	F1 Score
DistilBERT	88.51	88.46	88.61	88.53
RoBERTa	89.66	90.20	89.77	89.98
RoBERTa + BiLSTM	**92.34**	**92.34**	**92.38**	**92.36**

6.1. Results on machine learning models

In this subsection, we present the results obtained for various classical machine-learning models. Table 1 summarizes the results obtained. We experimented with fastText, TF-IDF word vectors and BERT word embedding embeddings. Out of the analyzed models, Random Forest, Multinomial Naïve Bayes, Support vector machine, and Logistic Regression showed best performance when trained on Tf-idf feature vectors while K-Nearest Neighbor and Decision Tree showed best performance when trained on pre-trained Bert embedding and XG-Boost shows best performance on FastText word embedding. Hence, majority of the analyzed classical machine learning models perform best with TF-IDF feature vectors on this specific dataset, especially when using TF-IDF weighted average. The performance of classical machine learning models is depicted in figure 2.

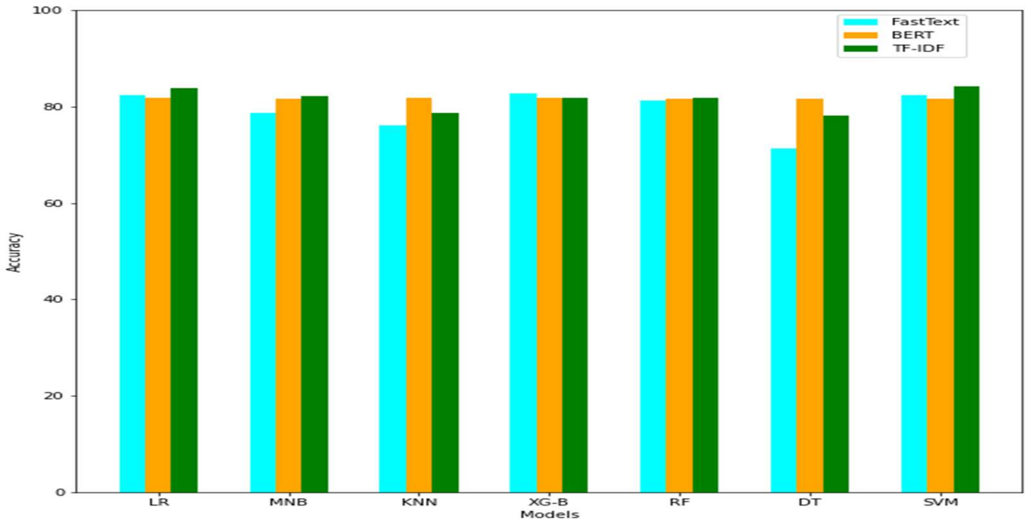

Figure 2: Performance of Machine learning models on various embedding

6.2. Results on deep learning models

We analyzed CNN, LSTM, and an ensemble of these two for the purpose of this study. Two different embeddings, BERT word embedding, and Glove word vectors were used to evaluate their performance. The summarized results are shown in Table 2. It is clear that deep-learning based approaches usually beat classical machine-learning approaches in terms of performance on this particular dataset. Figure 3 shows a plot of analyzed deep learning models on GloVe and BERT embeddings.

CNN: We examine CNN with Bert Word embedding as well as Glove Word embedding. We initialize the embedding-layer with pretrained embedding (BERT or Glove). Using Bert embedding, resulted in a slightly higher performance (85% accuracy and 84.34% F1 score) when compared with using glove word embedding (84.53% accuracy and 83.02% F1 score).

LSTM: As the next deep-learning approach, we evaluate LSTM. We recreate the same setup as defined for CNN above; that is, used Bert and GloVe word embeddings. LSTM model outperforms the CNN model generally. The use of Bert word embedding proves to perform best in this case also (Table 2).

CNN+LSTM: We finally explore a hybrid model based on a CNN followed by an LSTM layer. This hybrid model performs best overall and specifically when initialized with Bert word embedding (see Table 2).

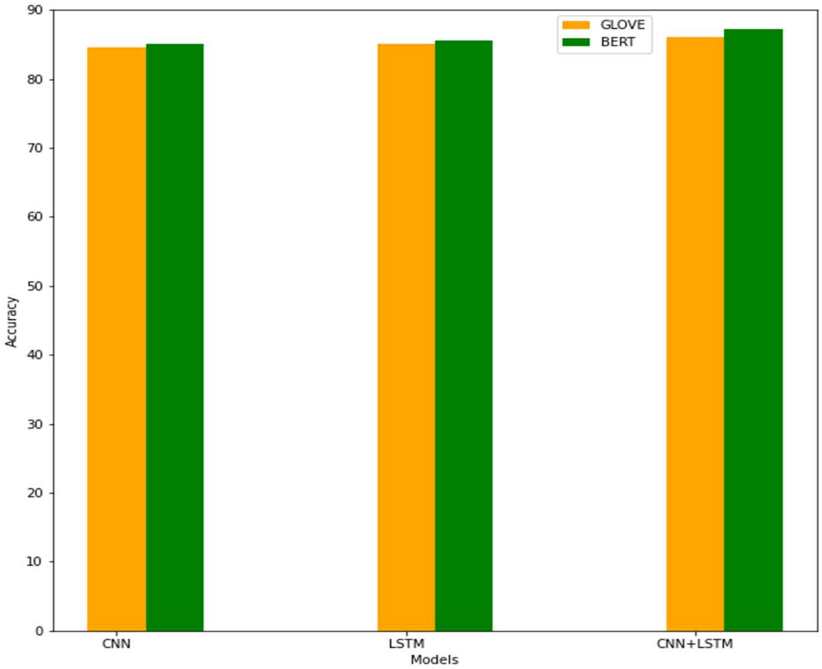

Figure 3: Performance of Deep learning models on GloVe and BERT embeddings.

6.3. Results on pre-trained models

As can be observed from Table 3, the advanced language models clearly outperform all the machine-learning and deep-learning based approaches. The language models do not need large datasets because they use pretrained embedding weights, and hence they show better performance during the start of fine-tuning itself as compared to the deep learning models that require large datasets for satisfactory performance.

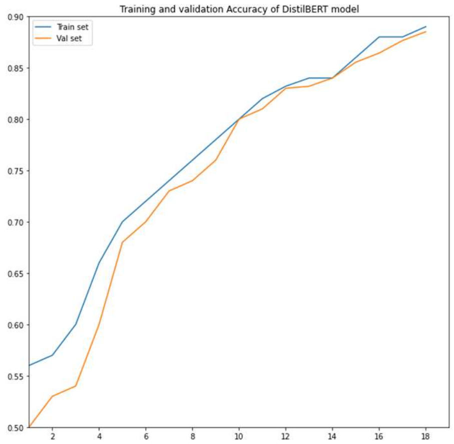

Figure 4: DistilBERT training and validation accuracy.

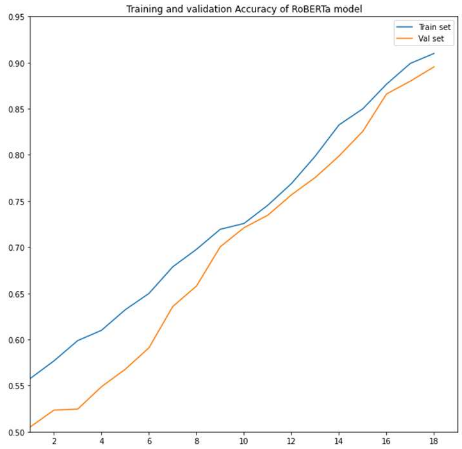

Figure 5: RoBERTa training and validation accuracy.

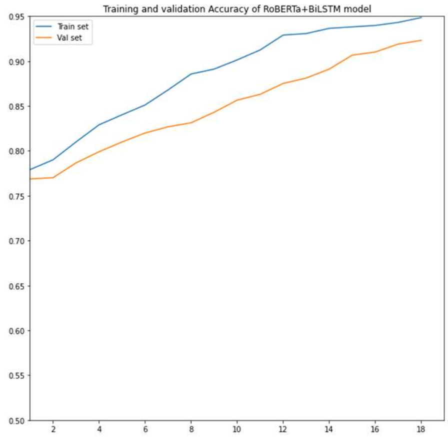

Figure 6: RoBERTa+ BiLSTM ensemble Training and validation accuracy.

DistilBert (66M parameters) model, roBERTa (125M Parameters) achieve an accuracy of 88.5% and 89.7% respectively, as shown in figures 4 and 5. This gives a clear indication of the fact that these models' performance is directly proportional to their parameter size.

Lastly, we analyzed the performance of the roBERTa model when stacked with a bi-LSTM layer on top of it. The bi-LSTM-attention extracts the sentence features automatically. This hybrid model proved to perform best on the combined corpus with an accuracy of 92.3%. As depicted in figure 6.

The confusion matrix showing the individual number of true and fake predictions made by these models is depicted in Figure 5.

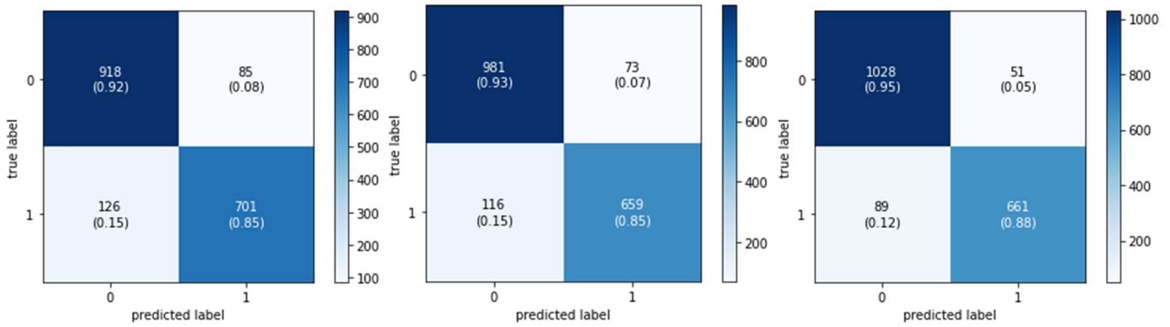

Figure 7: Confusion Matrix of DistilBERT, RoBERTa, and RoBERTa + LSTM models from left to right respectively.

7. Conclusion

In this comparative study, we did analysis of classical machine-learning, deep learning, and pretrained language models on fake-news related to COVID 19 on social media platforms. It is evident from the study that the transformer-based approaches perform best overall. The pre-trained models perform significantly better even on comparatively smaller data samples, as compared to deep learning models which suffer from over-fitting on smaller datasets. Support Vector Machines combined with TF-IDF feature vectors attained performance close to deep learning-based approaches. The CNN-LSTM model showed performance close to pretrained language models. The CNN layer learns the spatial and invariant features of the news items.

Findings from this study can facilitate future research in this direction. In this study, we attended fake-news about COVID19 problem to examine how good different models perform on this very subtask of fake news detection. We will target designing a generalized fake-news detection model in our future work.

8. References

[1] Mustafaraj E, Metaxas PT. The fake news spreading plague: was it preventable? In: In *Proc. of the 9th ACM Conference on Web Science* (WebSci); 2017. p. 235–239.

[2] Balmas M. When fake news becomes real: Combined exposure to multiple news sources and political attitudes of inefficacy, alienation, and cynicism. *Communication Research*. 2014;41(3):430–454.

[3] Brewer PR, Young DG, Morreale M. The impact of real news about "fake news": Intertextual processes and political satire. *International Journal of Public Opinion Research*. 2013;25(3):323–343.

[4] Jin Z, Cao J, Zhang Y, Luo J. News verification by exploiting conflicting social viewpoints in microblogs. In: In *Proc. of the 13th AAAI Conference on Artificial Intelligence* (AAAI); 2016. p. 2972–2978.

[5] Rubin VL, Conroy N, Chen Y, Cornwell S. Fake news or truth? using satirical cues to detect potentially misleading news. In: In *Proc. of the Second Workshop on Computational Approaches to Deception Detection*; 2016. p. 7–17.

[6] Kshetri N, Voas J. The economics of "fake news". *IT Professional*. 2017;19(6):8–12.

[7] Gelfert A. Fake news: A definition. Informal Logic. 2018;38(1):84–117.

[8] Sharma K, Qian F, Jiang H, Ruchansky N, Zhang M, Liu Y. Combating fake news: A survey on Identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology* (TIST). 2019;10(3):1–42.

[9] Reis, Julio CS, Andr´e Correia, Fabr´ıcio Murai, Adriano Veloso, and Fabr´ıcio Benevenuto. “Supervised learning for fake news detection.” IEEE Intelligent Systems 34, no. 2 (2019):76-81.

[10] Aslam N.,Khan I.,Alotaibi F. ,Aldaej L. and Aldubaikil A. *Fake Detect: A Deep Learning Ensemble Model for Fake News Detection.*Complexity ,2021 ,(2021) ,5557784 , 1-8,

[11] Patwa, Parth, Shivam Sharma, Srinivas PYKL, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. “Fighting an Infodemic: COVID-19 Fake News Dataset.” arXiv preprint arXiv:2011.03327 (2020).

[12] A. Wani, I. Joshi, S. Khandve, V. Wagh, and R. Joshi, “Evaluating deep learning approaches for COVID-19 fake news detection,” 2021, http://arxiv.org/abs/2101.04012.

[13] Wang, W. Y. (2017). "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *arXiv*. https://doi.org/10.48550/arXiv.1705.00648

[14] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N.Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need.” Advances in neural information processing systems 30 (2017): 5998-6008.

[15] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pretraining of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018)

[16] Kaliyar, R.K., Goswami, A. & Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimed Tools Appl* **80**, 11765–11788 (2021). https://doi.org/10.1007/s11042-020-10183-2

[17] Sunil Gundapu and Radhika Mamidi. Transformer based automatic COVID-19 fake news detection system. CoRR, abs/2101.00180, 2021.

[18] Muller, M., Salathe, M., Kummervold, P. E.: (2020). COVID-Twitter-BERT: A natural language processing model to analyse COVID-19 content on Twitter. arXiv preprint arXiv:2005.07503.

[19] Dipta, S., Basak, A., Dutta, S. (2021). A heuristic driven ensemble framework for COVID-19 fake news detection. In Combating Online Hostile Posts

[20] Julio A. Saenz, Sindhu Reddy Kalathur Gopal, Diksha Shukla, June 12, 2021, "Covid-19 Fake News Infodemic Research Dataset (CoVID19-FNIR Dataset)", IEEE Dataport, doi: https://dx.doi.org/10.21227/b5bt-5244.

[21] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information", *Transactions of the Association for Computational Linguistics*, vol. 5, no. 1, pp. 135-146, 2017.

[22] Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1532–1543.

[23] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: arXiv preprint arXiv:1910.01108 (2019).

[24] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv: Computation and language