Fake News Classification

Rohil Rao (3299480), Poulami Nath (404974), and M A Al-Masud (3156691)

Institute of Computer Science
University of Bonn

MA-INF 4232 Lab Information Retrieval in Practice
Summer Semester 2021

Supervisors:
Prof. Dr. Elena Demidova
Dr. Ran Yu
Alishiba Dsouza

July 9, 2021

Abstract. The spread of fake news on the internet is a matter of serious concern due to its potential to cause social and national damage. If left unchecked, it can have disastrous consequences like nationwide mistrust in public institutions. A lot of existing research is focused on the challenging task of automatic fake news detection. This report analyzes some of the research in the domain and explores various machine learning and deep learning models. We present a comparison of these models using natural language processing tools to analyze news articles and classify them based on their textual characteristics. Classification performance is evaluated using different metrics comparing pre-processing and word embedding techniques. We show how explainable artificial intelligence can be used to understand the predictions of these black-box models which can help users decide if they should trust the model. Lastly, we discuss our findings and show two implementations of our models as web applications.

Keywords: Fake news · Fact-checking · Deep Learning · Explainable AI

1 Introduction

Over the last few decades, we have seen an exponential growth of information. Although this has been beneficial for the society, but also led to the problem of misinformation. Misinformation can either be due to inaccurate reporting of facts or malicious reporting to deceive the reader for spreading propaganda. The spread of fake news can have dire consequences for governments and society in general. Consider, for example, the 2016 US Elections where widespread fake news on social media was considered to be a major factor for polarization [1]. This problem is exacerbated when seen in the context of big data. Factors like the ever-increasing volume, velocity, and variety have made it very challenging to ensure the veracity (or correctness) of digital data. Traditionally the job of fact-checking has been done by professionals. However, with the exponential growth of data, it is important to look for more sustainable solutions. Automated fact-checking (AFC) methods can play a key role in assisting humans. Fig. 1 shows the various sub-processed of automated fact-checking.

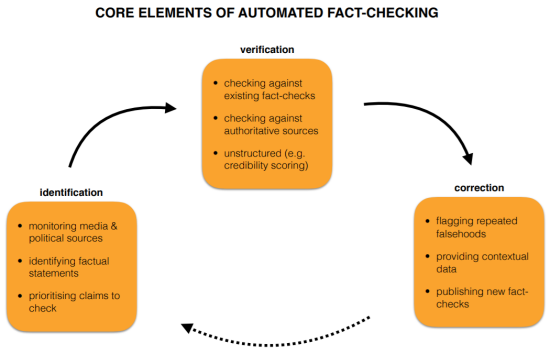

Fig. 1. Overview of automated fact-checking[1]

[1] Lucas Graves: Understanding the Promise and Limits of Automated Fact-Checking, `https://edgeryders.eu/t/get-your-facts-straight-with-ai/10108`

With the continued spread of fake news in various forms across the Internet, it has become even more important to study counteractive methods. The promising results of deep learning and machine learning methods in the field of natural language processing further motivate us to study their applications for fake news classification.

2 Problem Description and Research Questions

Our primary objective in this work is to perform binary classification of news articles as 'Real' or 'Fake' using various machine learning and deep learning methods. Specifically we focus on analysing and using the content and available contextual information in news articles for classification. The project is limited only to news articles available in English.

Problem definition: Given the description for a particular document (news article) and a training set with a fixed set of classes we wish to learn a supervised learning algorithm that maps the documents (news articles) to their respective classes. This definition is adapted from the *Introduction to Information Retrieval* by Manning et al. [2]

For the basis of our project work we have we have defined the following research questions:

1. How well can automated classification methods perform on the task of fake news detection?
2. How does the performance of machine learning algorithms compare with deep learning methods on the given task?
3. What feature engineering or analytical methods can be used to obtain insights about news articles?
4. How can we explain the predictions of the proposed models?

We explain our findings in the sections below. The outline of this report is as follows: In section 3 we present the literature review and related work. In Section 4 we describe the selected datasets. Section 5 describes the methodologies used for the purpose of automated classification. Section 6 and 7 describe our experiments and results respectively. In section 8 we describe the implementation of our methods as a web application. Finally in section 9 we present our conclusion and future work for this project. Section 10 talks about the individual contributions made to the project.

3 Literature Review

3.1 Machine Learning Methods

For the purpose of fake news classification, Bharadwaj et al. [3], compared the performance of various machine learning models and word embedding methods on a subset of the Kaggle fake news dataset[2]. Ensembling machine learning methods were compared for four different datasets by Ahmad et al. [4]. Shrestha et al., [5] show various methods of extracting stylistic, psychological and complexity features from text articles. They perform a reproducibilty study of Horne and Adali [6] to understand the effect of textual characteristics of the title and the body of news articles on the performance of machine learning methods.

[2] Getting real about fake news, `https://www.kaggle.com/mrisdal/fake-news`

3.2 Deep Learning Based Techniques

After the dawn of deep neural network, it has become very much popular in text classification. Girgis et al., citegirgis2018deep primarily focused on fake news classification based on vanilla recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU) and convolutional neural network (CNN) but the result is not promising due to low test accuracy score. Bahad et al., [7] worked on bidirectional LSTM (Bi-LSTM) but worked on very small dataset. So their result does not reflect how it will behave for larger dataset. Singhania et al., [8] worked on hierarchical attention network (HAN) showing promising result. Thota et al., [9] worked on dense neural network with only TF-IDF and Word2vec [10] embedding. Abdullah et al., [11] showed that combining CNN and LSTM could produce excellent result but they only worked with one dataset. Use of different datasets can show the confidence of a model to predict.

3.3 Explainability

Explaining the predictions of automated models is important to understand if the model can be trusted. There have been many proposed methods in the domain of explainable fact-checking (of which fake-news detection is also a part) as summarized by Kotonya and Toni [12]. Many works in this domain, that use attention mechanisms, have used neural attention weights to explain predictions. Shu et al., [13] use a co-attention network for explainable fake news detection using user comments. We present the results of model agnostic explainability methods LIME [14] in our work.

4 Datasets

We have chosen 4 different datsets for comparing the performance evaluation among different models. Dataset (DS 1)[3] and Dataset 2 (DS 2)[4] have been taken from Kaggle. We have chosen Dataset 3 (DS 3)[5] provided by Information security and object technology (ISOT) research lab of University of Victoria. Lastly, we have combined all the above three datasets to form Dataset 4 (DS 4). All these datsasets contain real news from reputed website like Reuters whereas the fake news have been collected from unreliable websites flagged by Politifact. Table 1 summarizes the number of samples of each dataset including the number of fake and real news samples. In the end, we have considered the Dataset 4 as our final dataset.

Table 1. Number of dataset samples

Dataset	Real Sample	Fake Sample	Total Sample
DS 1	1,872	2,137	4,009
DS 2	10,387	10,413	20,800
DS 3	21,417	23,481	44,898
DS 4	33,676	36,031	69,707

[3] Kaggle jruvika fake news detection dataset. `https://www.kaggle.com/jruvika/fake-news-detection`
[4] Kaggle fake news dataset by UTK Machine Learning Club.`https://www.kaggle.com/c/fake-news/data`
[5] ISOT fake news detection dataset.`https://www.uvic.ca/engineering/ece/isot/datasets/fake-news/index.php`

The following table 2 shows the distribution of article lengths for each dataset. This gives us a general idea about the ability of our machine learning models to handle data as well as necessary information for selecting maximum length while feeding input to our deep learning models.

Table 2. Article length statistics of chosen datasets

Dataset	Minimum Length	Maximum Length	Average Length
DS 1	3	2,902	307.64
DS 2	5	20,680	409.50
DS 3	6	4,737	218.19
DS 4	3	20,680	280.32

5 Methodology

5.1 Data pre-processing

The steps taken for text pre-processing are: 1) removal of all the null values from the dataset, 2) removing duplicate records, 3) removing outlier articles, 4) removing other language articles, 5) getting rid of HTML tags, 6) removing extra whitespaces, 7) fixing contractions (for example, "don't" is converted to "do not"), 8) special characters removal, 9) transforming all texts into lowercased texts, 10) numbers removal, 11) stopwords removal, 12) tokenization and 13) lemmatization.

5.2 Word Representation

Word embeddings are the texts converted into numbers and there may be different numerical representations of the same text. Many machine learning algorithms and almost all deep learning architectures are incapable of processing strings or plain text in their raw form. They require numbers as inputs to perform any sort of job, be it classification, regression etc. And with the huge amount of data that is present in the text format, it is imperative to extract knowledge out of it and build applications. Here, we have used three types of word representations, namely - TF-IDF, Word2vec and GloVe.

Term Frequency–Inverse Document Frequency (TF-IDF) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

Word2vec produces a vector space,with each unique word in the corpus such that words that share common contexts in the corpus are located close to one another in the space. It can be done using 2 approaches: skip-gram or continuous bag-of-words. In our project, we have used Word2vec of dimension 300.

The global vectors for word representation or GloVe algorithm is an extension to the Word2vec method for efficiently learning word vectors. In our project we have used GloVe of dimension 100.

5.3 Feature Engineering:

To answer the third research question *i.e.,* the effect of feature engineering methods on performance we extracted certain features based on language patterns of the news titles and articles. The feature extraction methods were inspired from Shrestha et al. [5], however the difference is that we do not use the LIWC (Linguistic Inquiry and Word Count) tool [15]. The types of features extracted are stated below.

Stylistic features which include the length of the text, the number of words and the number of sentences. We then performed parts-of-speech (POS) tagging to assign words to their lexical categories and obtained their respective frequencies in the text. This was done using the pre-trained averaged perceptron POS tagger from NLTK [16].

We then extract **Psychological features**. For this we used the open-source python library[6] for Empath [17] to analyze text to obtain around 200 features related to some pre-validated topics or emotions based on neural embeddings. The authors of Empath have reported a high correlation with LIWC [15]. We further obtain sentiment features from TextBlob (described in section 5.5 of this report) and VADER [7]. We have also done n-gram analysis to find the top 20 frequent 1-word and 2-words using unigram and bigram. Also, we have included WordCloud to understand if the news revolves around any particular person or place.[8].

Lastly we extract **Complexity features** that describe the readability of the text. We use the readability measures Flesh Kincaid Grade Level (FK), Gunning Fog Index (GI), Simple Measure of Gobbledygook Index (SMOG) as specified in [5]. Other features include lexical diversity measures like Type-token ratio (TTR) and average word and average sentence length.

5.4 Models

We have considered the below machine learning models based on their utility and usage for similar tasks in the domain. In **logistic regression** (LR), we used logistic function with l2 penalty to model the probability of an outcome happening. In **linear support vector machine** (LSVM), we used squared hinge loss function along with l2 penalty. In K**-nearest neighbor** (KNN), the number of neighbors used is 20. In **random forest** (RF), the number of trees used is 500 and we have used gini index for splitting. In **decision trees** (RF), we have also used gini index for splitting. We used **AdaBoost** (Adab) with 200 estimators to predict the output. We have also used **voting** which is an ensemble method where multiple classifiers are used to predict the output. In our case, we have used LR, KNN and RF with majority voting rule. In literature, we have found that some researchers have found using ensemble methods in text classification performance improving [4]. That was the motivation behind our implementation. Word2Vec pretrained word embeddings and Glove pretrained word embeddings has almost given similar results, so we have attached one. Splitting data in the ratio 60:40 for training to test datasets, we have managed to avoid overfitting. The hyperparameters used in these models have been described in section 6.2.

In NLP, recurrent neural network (RNN) architecture is used extensively in processing sequences of data like text. We have used 1-layered **long short-term memory** (LSTM) architecture of RNN. We have also added dropout to prevent the model from overfitting. Then we passed the output to a dense layer with one unit producing the output whether the given text is fake or real. We have

[6] `https://github.com/Ejhfast/empath-client`

[7] Valence Aware Dictionary and sEntiment Reasoner `https://github.com/cjhutto/vaderSentiment`

[8] Fake news classifier `https://www.kaggle.com/benroshan/fake-news-classifier-lstm`

chosen LSTM over vanilla RNN as vanilla RNN has a problem called vanishing gradient which is solved by LSTM.

Gated Recurrent Unit (GRU) is also an architecture of RNN which is newer than LSTM and also solves the problem of vanishing gradient. We have used 1-layered GRU accompanied with dropout feeding into a one dimensional dense network producing output. We used GRU to compare the performance between LSTM and GRU in terms of detecting fake news instances as well as computation time.

Bidirectional LSTM (Bi-LSTM) is like LSTM but here the input is being fed from beginning to end of the architecture and from end to the beginning as well hence the name bidirectional LSTM. We wanted to observe whether Bi-LSTM is an improvement to LSTM or not. We have also implemented like LSTM except instead of LSTM we have used Bi-LSTM.

Bidirectional GRU (Bi-GRU), like Bi-LSTM, is GRU where the input is fed from beginning to end and from end to beginx'x'ning. We have implemented it like GRU with the only exception of replacing GRU with Bi-GRU. We used it to see the performance comparison between GRU and Bi-GRU.

C-LSTM: Convolutional neural network (CNN) is heavily used to analyze images. But recently it has been used in text classification too. C-LSTM is a combination of CNN and LSTM. CNN is used in higher level phrase extraction and then it is being fed into LSTM. In practice, we have used 1-layered convolutional layer and then 1 dimensional max pooling with pool size 2. In the covolutional layer we have used ReLu as activation function. Then we have used a Bi-LSTM to predict the output. We have chosen this model because of the recent use of this hybrid-structure in the literature and see it for ourselves the performance with other models.

Attention mechanism has shown significant performance in deep neural networks in text processing. In **Hierarchical Attention Network** (HAN), the word encoder produces an annotation for a given word in a sentence using Bi-GRU. Then we use a 1 layer multi-layer perceptron (MLP) to get the word attention. Similarly, we use a sentence encoder and sentence attention to produce the document vector which gives a summary of the entire document. In practice, we have used 2-layered Bi-LSTM, 1 HAN layer, 1 dense layer with 100 units and 1 dense layer with 1 input to produce the output. In literature, the other researchers have found use of attention mechanism to be significantly performance improving. We also implemented this to see the performance for ourselves.

The hyperparameters used in these models have been described in section 6.2.

5.5 Sentiment Analysis

Sentiment analysis is the process of determining the emotion of the writer.The sentiment function of TextBlob [9] returns two properties, polarity, and subjectivity. Polarity is float which lies in the range of $[-1, 1]$ where 1 means positive statement and -1 means a negative statement. In our case, we have calculated the polarity of the news when the user enters the url in our web application to check the sentiment and also initially while processing the dataset.

5.6 Performance Metrics

Accuracy is the fraction of predictions the model got right. **Confusion matrix** is a summary table that breaks down the number of correct and incorrect predictions by each class. **Receiver**

[9] TextBlob library `https://textblob.readthedocs.io/en/dev/`

operating characteristic (ROC) curve is a plot that shows the true positive rate against the false positive rate at various threshold settings. The **area under curve** (AUC) indicates the probability that the classifier will rank a randomly chosen positive observation higher than a randomly chosen negative one. **Precision** is the fraction of relevant instances among the retrieved instances. **Recall** is the fraction of the total amount of relevant instances that were actually retrieved. **F1-score** is the harmonic mean of the precision and recall.

5.7 Explainability

Although deep learning models can achieve very good results on the task of binary classification, it is important to understand the reason for their decisions. What factors has the model considered most important to classify a news article as 'Real' or 'Fake'. In this project we have used LIME (Local interpretable model-agnostic explanations). This method suggested by Ribeiro et al. [14] is used to obtain local explanations. It treats the model as a black-box model and generates explanations by building surrogate models. Local explainability methods do not try to explain the entire model but instead try to approximate predictions of the model for individual instances. For each instance to be explained LIME creates a new dataset by making perturbations in the neighborhood of the instance. The idea then is to use an interpretable model like lasso regression or decision trees as a surrogate model to explain predictions of complex models on the newly created dataset.

6 Experiments

6.1 Environment

We have used Jupyter Notebook with Python 3 for carrying out our programming tasks. We have used *spacy* for natural language processing, *gensim* for word embedding, *scikit-learn* for traditional machine learning methods, *keras* for deep learning architectures, *matplotlib* and *seaborn* for visualisation. We have used the pre-trained English language model *en_core_web_lg* that comes up with *spacy* for Word2vec embedding and for GloVe, we have used the pre-trained Stanford word vectors. We have used two already implemented classes one for GloVe vectorizer[10] and another for HAN layer[11]

6.2 Hyperparameters

Using grid search we found the optimum hyperparameters for the traditional machine learning models. We could not perform the grid search for deep learning models because of the huge amount of time taken during training. So we checked for optimum parameters in some cases by manually altering the values. We have used the parameters mentioned in relevant papers in the literature review section for deep learning model. We used sigmoid function for output activation function, binary cross entropy for loss function, Adam for optimizing, 32 batch size for all deep learning models. For DS 1, we have trained the model for 5 epochs and for the rest 3 epochs.

[10] GloVe Vectorizer, `https://edumunozsala.github.io/BlogEms/jupyter/nlp/classification/embeddings/python/2020/08/15/Intro_NLP_WordEmbeddings_Classification.html`

[11] HAN Layer, `https://www.kaggle.com/sermakarevich/hierarchical-attention-network`

7 Result

7.1 Result for detecting fake news on DS 4

We have split all the datsets into 60% training dataset and 40% test dataset. The accuracy, macro precision, macro recall and macro f1-score for the combined dataset (DS 4) are displayed in table 4. Similar trends are observed in DS 1, DS 2 and DS 3. The ROC curves for DS 4 for both Word2Vec and GloVe are displayed in Fig. 2 and 3 respectively. The AUC score for traditional ML models for DS 4 with three word representations have been displayed in table 5.

Table 3. Hyperparameters used in different models

Model	**Values Used**
Logistic Regression (LR)	Penalty: l2, solver: stochastic average gradient
Linear Support Vector Machine (LSVM)	Penalty: l2, loss function: squared Hinge
K-nearest Neighbor (KNN)	Number of neighbors: 20
Random Forest (RF)	No. of trees: 500, maximum depth: 7
Ada Boost (AB)	Number of estimators: 200, maximum depth: 3
Decision Tree (DT)	Split function: Gini index, maximum depth: 5
Voting (LR + RF + KNN)	Used the previous values
Long Short-Term Memory (LSTM)	No. of units: 32, Dropout: 0.5
Gated Recurrent Unit (GRU)	
Bidirectional LSTM (Bi-LSTM)	
Bidirectional GRU (Bi-GRU)	
C-LSTM	1D Convolutional Layer - filters: 32, kernel size: 3, padding: same, activation: ReLU 1D Max pooling size: 2 Bi-LSTM unit: 32, Dropout: 0.5
Hierarchical Attention Network (HAN)	HAN dimension: 32 Bi-LSTM units: 32, 32 Dense units: 100, activation: ReLU

7.2 Results for feature engineering based methods on DS 4

Statistical Analysis: We calculate two-sample T-tests (two-sided statistical tests) for the null hypothesis that two independent samples have identical averages. We only report a few of the results for 'Real' vs 'Fake' news articles (from the combined dataset DS 4) that are statistically significant (p-value < 0.05). As found in [5] we note that 'Real' news articles have a higher word and sentence count, as well as higher average word and sentence length. Also, 'Fake' articles have a higher lexical diversity (Type-token ratio) and use more personal pronouns and adverbs. Some results are not conclusive across different measures used or do not match findings in [5]. The complete list of results can be seen in our code.

Comparing performance on title, body and both combined: Using the feature engineering methods specified in section 5.3 we obtain around 240 features for both the title and the

Table 4. Accuracy, macro precision, macro recall and macro f1-score for DS 4

Model	TF-IDF				Word2vec				GloVe			
	Acc	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**
LR	0.91	0.91	0.91	0.91	0.85	0.85	0.85	0.85	0.80	0.80	0.80	0.80
LSVM	**0.92**	**0.92**	**0.92**	**0.92**	0.86	0.86	0.86	0.86	0.80	0.80	0.80	0.80
KNN	0.61	0.70	0.60	0.54	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
RF	0.87	0.87	0.87	0.87	0.83	0.83	0.83	0.83	0.82	0.82	0.82	0.82
Adab	0.87	0.87	0.87	0.87	0.80	0.80	0.80	0.80	0.78	0.78	0.78	0.78
DT	0.86	0.86	0.86	0.86	0.72	0.72	0.72	0.72	0.73	0.73	0.73	0.73
Voting	0.82	0.85	0.82	0.81	0.86	0.86	0.86	0.86	0.84	0.84	0.84	0.84
LSTM					0.92	0.92	0.92	0.92	0.87	0.87	0.87	0.87
GRU					0.93	0.93	0.93	0.93	0.89	0.90	0.90	0.89
Bi-LSTM					0.91	0.91	0.91	0.91	0.84	0.84	0.84	0.84
Bi-GRU					0.93	0.93	0.93	0.93	0.90	0.90	0.90	0.90
C-LSTM					0.92	0.90	0.96	0.93	0.87	0.88	0.87	0.87
HAN					**0.96**	**0.97**	**0.96**	**0.96**	**0.93**	**0.93**	**0.93**	**0.93**

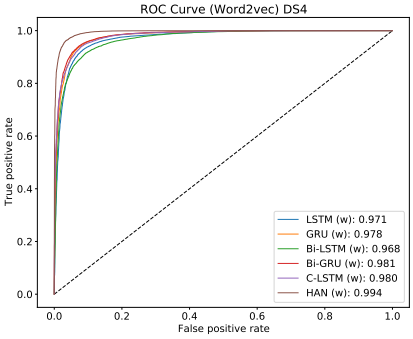

Fig. 2. ROC curve for DS 4 (Word2Vec)

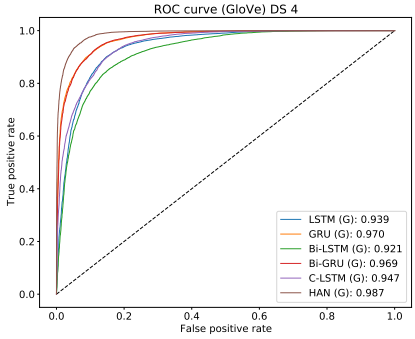

Fig. 3. ROC curve for DS 4 (GloVe)

Table 5. AUC score of traditional ML models for DS 4

Model	TF-IDF AUC	Word2Vec AUC	GloVe AUC
LR	0.96	0.95	0.93
LSVM	0.96	0.95	0.93
KNN	0.83	0.90	0.91
RF	0.83	0.88	0.88
Adab	0.93	0.94	0.93
DT	0.91	0.80	0.84

body of news articles in the combined dataset 4 (DS4). In table 6 we present a comparison of some machine learning models like Adaboost (Adab), Decision Trees (DT), Random Forests (RF) and Logistic Regression (LR) as done in [5]. The results are presented separately for the set of stylistic, psychological, and complexity features (described in section 5.3) as well as all combined. Also, the results are reported separately considering only the title or the body as well as both combined.

Table 6. Area under ROC and Macro average precision for feature engineering method used on DS 4

Model	Stylistic		Psychology		Complexity		All Combined	
	AUROC	AvgP	AUROC	AvgP	AUROC	AvgP	AUROC	AvgP
Title (Adab)	**0.80**	**0.73**	0.81	0.72	**0.75**	**0.68**	0.85	0.70
Title (DT)	0.66	0.66	0.72	0.71	0.68	0.65	0.74	0.74
Title (RF)	0.80	0.73	**0.86**	**0.78**	0.73	0.67	**0.9**	**0.82**
Title (LR)	0.75	0.7	0.78	0.71	0.678	0.659	0.84	0.77
Body (Adab)	0.83	0.75	0.84	0.76	0.73	0.68	0.88	0.8
Body (DT)	0.69	0.69	0.69	0.69	0.62	0.62	0.73	0.73
Body (RF)	**0.86**	**0.78**	**0.88**	**0.8**	**0.76**	**0.69**	**0.91**	**0.83**
Body (LR)	0.81	0.73	0.85	0.77	0.67	0.63	0.89	0.81
Both (Adab)	0.88	0.81	0.87	0.79	0.81	0.75	0.91	0.84
Both (DT)	0.76	0.76	0.73	0.73	0.70	0.70	0.79	0.79
Both (RF)	**0.91**	**0.84**	**0.9**	**0.82**	**0.86**	**0.79**	**0.94**	**0.87**
Both (LR)	0.85	0.78	0.87	0.79	0.72	0.68	0.92	0.85

7.3 Explainability

The fig.4 shows an example of how LIME [14] can explain the words considered most important for prediction. We see that the model considers the word 'reuters' important for prediction (most probably since the dataset contains many true articles collected from Reuters[12]).

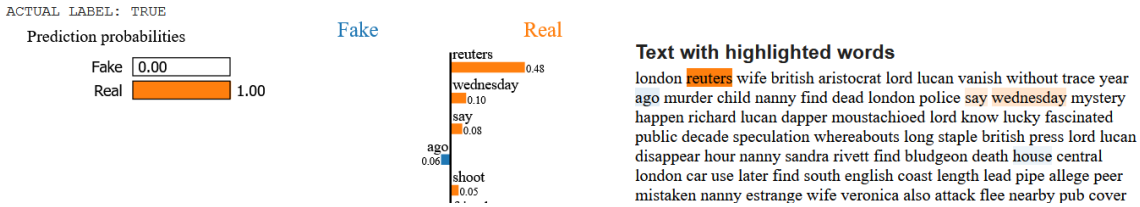

Fig. 4. Example of LIME explainability for prediction of LSTM on test instance

7.4 Discussion

From the results in section 7.1, we answer research questions 1 and 2. We have found out that Linear Support Vector Machine (LSVM) performs better when TF-IDF is used and Hierarchical Attention Network (HAN) performs better while Word2Vec and GloVe are used. Overall, HAN performs better with Word2vec. However, it takes a considerable amount of time while training. Hence

[12] `https://www.reuters.com/`

if we are to trade-off between time and performance, HAN with Word2vec takes more amount of time than LSVM with better performance than LSVM.

In section 7.2, we focus on research questions 3 and 4. We note that there are some statistically significant differences between real and fake news articles in our dataset. We also note that feature engineering methods on titles and bodies of articles give good results. The best results are seen for all features combined. Lastly, we see that our model predictions can be explained using LIME. We only show results for a single instance, however, we noted that the models sometimes assign importance to irrelevant words for classification. This can most likely be attributed to the limitations of the dataset.

8 User Interface Design

Based on our results we propose two possible web application for fake news classification with explainability. The goal was to create a dashboard using the methods discussed in the previous sections of the report. The first web application **App-1** allows provides the classification results along with an explanation. The intention is to help the users decide if they should trust the model prediction. App-1 is hosted via a free heroku server and is available online.[13]

The second web application **App-2** is shown below:This is a more simple application where the user will enter his/her desired URL and based on the that the system will provide a feedback whether the news is fake or not along with polarity and other information.

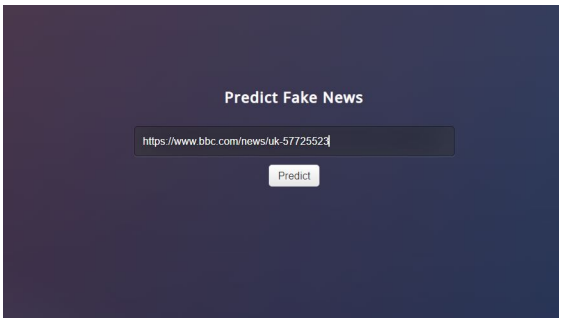

Fig. 5. Example of the prototype web application

Fig. 6. Example of the prototype web application

[13] Heroku deployment address, `https://irlab21-fakenews-explainer.herokuapp.com`

Both the User Interfaces were built using the Flask web framework. Article scraping was done using the Newspaper3k[14] python package. The scraped articles are pre-processed, however in App-1, to reduce response time we only convert text to lowercase and fix contractions. Stop-word removal is not performed. Since stop-words are not used at train time they will be considered as out-of-vocabulary words for the model and hence they will be ignored. Lastly the model results are explained using the Lime package.

9 Conclusion

In this paper we have presented various methods for automatic binary classification of fake news. We prepared a combination of datasets to compare the performance of machine learning and deep learning techniques for various pre-processing and feature engineering methods. We also use explainability methods to understand our model predictions. Lastly we show two possible web applications for deploying our work.

Future Work: It is possible to extend this work by using complicated datasets like [18] containing images and spatio-temporal information. We would also like to analyze other explainability methods like SHAP values [19].

10 Individual Contribution

Rohil Rao (Team Leader): Worked on machine learning models using TF-IDF and trained Word2Vec for LSTM. Worked on feature engineering based methods. Explored explainability methods for proposed models. Prepare a web based fake news application using Flask.

Poulami Nath: Implemented the mentioned traditional Machine Learning models using TF-IDF, Word2Vec and GloVe embeddings for DS 1 and DS 4. Participated in data cleaning and preprocessing and prepared a web based fake news application using Flask.

M A Al-Masud: Implemented all traditional machine learning and deep learning based models with TF-IDF, Word2Vec and GloVe embeddings for all 4 datasets.

The tasks which are common in individual contributions have been carried out by all and the best results have been reported.

Acknowledgement. This project report and the associated work are courtesy of Prof. Dr. Elena Demidova, Dr. Ran Yu and Alishiba Dsouza. We thank them for their guidance in the completion of this work.

[14] Newspaper3k: Article scraping curation, `https://newspaper.readthedocs.io/en/latest/`

References

1. Alexandre Bovet and Hernán A Makse. Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):1–14, 2019.
2. Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.
3. Avinash Bharadwaj, Brinda Ashar, Parshva Barbhaya, Ruchi Bhatia, and Zaheed Shaikh. Source Based Fake News Classification using Machine Learning. *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, 9, 2020.
4. Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad. Fake news detection using machine learning ensemble methods. *Complexity*, 2020, 2020.
5. Ana Shrestha and Francesca Spezzano. Textual characteristics of news title and body to detect fake news: A reproducibility study. 2021.
6. Benjamin 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. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.
7. Pritika Bahad, Preeti Saxena, and Raj Kamal. Fake news detection using bi-directional lstm-recurrent neural network. *Procedia Computer Science*, 165:74–82, 2019.
8. Sneha Singhania, Nigel Fernandez, and Shrisha Rao. 3han: A deep neural network for fake news detection. In *International conference on neural information processing*, pages 572–581. Springer, 2017.
9. Aswini Thota, Priyanka Tilak, Simrat Ahluwalia, and Nibrat Lohia. Fake news detection: a deep learning approach. *SMU Data Science Review*, 1(3):10, 2018.
10. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
11. A Abdullah, M Awan, M Shehzad, and M Ashraf. Fake news classification bimodal using convolutional neural network and long short-term memory. *Int. J. Emerg. Technol. Learn*, 11:209–212, 2020.
12. Neema Kotonya and Francesca Toni. Explainable automated fact-checking: A survey. *CoRR*, abs/2011.03870, 2020.
13. Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. Defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '19, page 395–405, New York, NY, USA, 2019. Association for Computing Machinery.
14. Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.
15. James W Pennebaker, Roger J Booth, and Martha E Francis. Linguistic inquiry and word count: Liwc [computer software]. *Austin, TX: liwc. net*, 135, 2007.
16. Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.
17. Ethan Fast, Binbin Chen, and Michael S. Bernstein. *Empath: Understanding Topic Signals in Large-Scale Text*, page 4647–4657. Association for Computing Machinery, New York, NY, USA, 2016.
18. Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*, 2018.
19. Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.