Fake News Detection Using NLP

Phase 1 – Problem definition and Design thinking

Problem definition :

In this subsection, we present the details of mathematical

formulation of fake news detection on social media. Specifically, we will introduce the definition of key components

of fake news and then present the formal definition of fake

news detection. The basic notations are defined below,

• Let a refer to a News Article. It consists of two major components: Publisher and Content. Publisher ~pa

includes a set of profile features to describe the original author, such as name, domain, age, among other

attributes. Content ~ca consists of a set of attributes

that represent the news article and includes headline,

text, image, etc.

• We also define Social News Engagements as a set of

tuples E = {eit} to represent the process of how news

spread over time among n users U = {u1, u2, ..., un}

and their corresponding posts P = {p1, p2, ..., pn} on

social media regarding news article a. Each engagement eit = {ui, pi, t} represents that a user ui spreads

news article a using pi at time t. Note that we set

t = Null if the article a does not have any engagement yet and thus ui represents the publisher.

Definition 2 (Fake News Detection) Given the social

news engagements E among n users for news article a, the

task of fake news detection is to predict whether the news

article a is a fake news piece or not, i.e., F : E → {0, 1}

such that,

F(a) = (1,

if a is a piece of fake news,

0

, otherwise.

(1)

where F is the prediction function we want to learn.

Note that we define fake news detection as a binary classifi-

cation problem for the following reason: fake news is essentially a distortion bias on information manipulated by the publisher. heory [26], distortion bias is usually modeled as a binary

classification problem.

Next, we propose a general data mining framework for fake

news detection which includes two phases: (i) feature extraction and (ii) model construction. The feature extraction

phase aims to represent news content and related auxiliary

information in a formal mathematical structure, and model

construction phase further builds machine learning models

to better differentiate fake news and real news based on the

feature representations.

Design Thinking:

1.Data Sources:

The dataset used in this work is a fusion of several fake and real news articles about COVID-19 which are collected across several platforms such as Facebook, Twitter, The New York Times, Harvard Health Publishing, WHO, etc. The dataset has 1,164 instances out of which 586 instances are true and the remaining 578 are fake news (26).

2.Data Pre processing:

The dataset considered for this work is clean. However, some unnecessary symbols which have an impact on the final classification of the news are to be removed from the dataset. To remove the unnecessary symbols, such as punctuation marks, URLs are removed from the dataset as part of preprocessing.

Tokenization is the process of splitting text into a set of tokens. The fake news dataset is tokenized to convert the long sentences into small words/tokens.

3.Feature Extraction:

The major contribution of this work is the extraction of important features from the COVID-19 fake news dataset. Feature extraction plays a very important role in text processing as it reduces the dimension of feature space by considering only the important features (27–29). To extract the features, the named-entity recognition (NER) approach is used in our work. The NER is a popular approach for feature extraction that can classify unstructured text based on location, person names, quantities, etc. (30).

In this study, 39 features are created from the COVID-19-related fake news dataset. The extracted features are represented in Table 1.

Table 1

Feature name	Data type	Feature name	Data type
News source	Non-Numeric	Num of?	Numeric
Num of Stopwords	Numeric	Num of /	Numeric
Num of @	Numeric	Num of #	Numeric
Num of numeric values	Numeric	Num of uppercase characters	Numeric
Num of lowercase characters	Numeric	Num of all uppercase characters	Numeric
Text language	Numeric	Word count	Numeric
Character count	Numeric	Sentence count	Numeric
Average word length	Numeric	Average sentence length	Numeric
Positive Sentiment Score	Numeric	Negative Sentiment Score	Numeric
Neutral Sentiment Score	Numeric	Compound Sentiment Score	Numeric
Person	Numeric	NORP	Numeric
FAC	Numeric	Organization	Numeric
GPE	Numeric	Location	Numeric
Product	Numeric	Event	Numeric
Work of Art	Numeric	Law	Numeric
Language	Numeric	Date	Numeric
Time	Numeric	Percent	Numeric
Money	Numeric	Quantity	Numeric
Cardinal	Numeric	Ordinal	Numeric
Text Polarity	Numeric		

4.Model Selection:

Different classification models can be applied in this case,

but to choose the most adequate one and to tune its parameters

we run several experiments on different models. We started

experimenting with classification models that have proven to be effective and give good results in related sentence classification tasks. Some of the models did not give good results and were discarded, one of them was Logistics Regression, while Support Vector Machines, naïve Bayes and Passive Aggressive gave promising results and we continued to experiment on them. To check the accuracy, we compare our results with other datasets through performance metrics.

 Naïve Bayes: It is a powerful classification model that performs well when we have a small dataset and it requires less storage space. It does not produce good results if words are co related between each other [18]. Fig. 5 contains the Naïve Bayes formula that explains the probability of an attribute that belongs to a class independent from other classes.

 Support Vector Machine: It performs supervised learning on data for regression and classification. The SVM computes the data and converts it into different categories

5.Model Training:

Fake news is increasing every second without proper checks and balances, so there is a need for computational tools that can handle this problem. Machine learning algorithms like "CountVectorizer", "TFIDFVectorizer", naïve Bayes, Support Vector Machine, Passive Aggressive Classifier and NLP for the identification of false news in public data sets are proposed. This is purely a text-based classification problem but our actual goal is the combination of text-based classification with machine-based text transformation and then

choosing which type of text is to be used, e.g. single news or the full body of the news. The overall data cleaning process is

A.NLP Models

Irrelevant and redundant features in a dataset have a negative impact on the accuracy and performance of the classifier. So, in those cases, we perform feature reduction to reduce the text feature size that limited the words like "the", "and", "there", "when" and focus only on those words which appear a given number of times. This is done by using nnumber of use words, lower casing and stop word removal since the sensitivity of the problem, which is increasing every second without check and balance, is understood [22]. essential to use machine learning algorithms like CountVectorizer and TF-IDF to speed up the task and improve performance.

learning models

B.Count Vectorization

CountVectorizer provides a simple way to collect text documents and to help build the vocabulary of known distinctive words and also to encode new documents using that vocabulary [23]. Given a collection of text documents, S to Count Vectorizer and it will generate a sparse matrix of size A where m = total number of documents, n = total number of distinct words used in S. With the Count Vectorizer, we can produce a table for each word and occurrence of each class.

C.Term Frequency- Inverse Document Frequency

To measure a term in documents over a dataset, we used the term frequency-inverted document frequency. A term's importance increases in the document which appears in the

dataset and also the frequency of the words. So, with the help of this method, we can weigh the metric that is used for information retrieval [24]. TF-IDF for the word with respect to document d and corpus D is calculated as follows:

$$TF - IDF(w)d,D = TF(w)d \times IDF(w)D. \quad (1)$$

Let us suppose we have a document with 100 words and we need to calculate TF-IDF for the word "rumor." The word "rumor" occurs in the document 4 times; then we can calculate, TF = 4/100 = 0.04. Now, we need to calculate the IDF; let us assume that we have 200 documents, and "rumor" appears in 100 of them. Then, IDF(rumor) = 1+log (200/100) = 0.5, and TF-IDF(rumor) = 0.05 × 0.5 = 0.025

6.Evaluation:

The results suggested that the approach is highly favorable since this application helps in classifying fake news and identifying key features that can be used for fake news detection. Our proposed technique suggests that to differentiate fake and non-fake news articles, it is worthwhile to look at machine learning methods. The developed system with accuracy up to 93% proves the importance of the combination; next, we need to look into other methods for fake news detection except for simple text classification. The producers of fake news are using different techniques to hide their identity, so they can easily mislead readers. As we are aware that every single news has different characteristics so there is a need for a system that can check the content of the news in depth. Our future work includes building an automated fact-checking system that combines data and knowledge to help non-experts and checks the content of the

news thoroughly after comparing it with known facts. We want to look into the issue of fake news from different angles like known facts, source, topic, associated URLs, geographical location, year of publication, and credibility of the source for a better understanding of the problem. The open issues and challenges are also presented in this paper with potential research tasks that can facilitate further development in fake news research.