Towards partial fulfillment for Undergraduate Degree Level Programme
Bachelor of Technology in Computer Engineering

A Third Stage Project Evaluation Report on:

NEWS DATA ANALYSIS

Prepared by :

Admission No.	Student Name
U17CO085	KALP PANWALA
U17CO104	KESHAV GOYAL
U17CO107	RAJ SHAH
U17CO113	VIREN KATHIRIYA

Class : B.TECH. IV (Computer Engineering) 7th Semester

Year : 2020-2021

Guided by : Dr. DIPTI P. RANA

DEPARTMENT OF COMPUTER ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY,
SURAT - 395 007 (GUJARAT, INDIA)

Student Declaration

This is to certify that the work described in this project report has been actually carried out and implemented by our project team consisting of

Sr.	Admission No.	Student Name
1	U17CO085	Kalp Panwala
2	U17CO104	Keshav Goyal
3	U17CO107	Raj Shah
4	U17CO113	Viren Kathiriya

Neither the source code therein nor the content of the project report has been copied or downloaded from any other source. We understand that our result grades would be revoked if later it is found to be so.

Signature of the Students:

Sr.	Student Name	Signature of the Student
1	Kalp Panwala	Kalp Panwala
2	Keshav Goyal	
3	Raj Shah	Rshah
4	Viren Kathiriya	Kathiriya

Certificate

This is to certify that the project report entitled

<u>News Data Analysis</u> is prepared and presented by

Sr.	Admission No.	Student Name
1	U17CO085	Kalp Panwala
2	U17CO104	Keshav Goyal
3	U17CO107	Raj Shah
4	U17CO113	Viren Kathiriya

Final Year of Computer Engineering and their work is satisfactory.

SIGNATURE :

GUIDE JURY HEAD OF DEPT.

Abstract

The issue of online fake news has been increasing rapidly misleading many people. Because of These people find it difficult to believe the news online. This generates the need to find new tools that can do the verification process. Thus, the goal is to find a classification model that identifies the phony features accurately using fine-grain and coarse-grain feature extraction techniques. This report proposes a framework that explores a method to combine the coarse-grain features and fine-grain features. The fine-grain features were extracted using the empath library and Vader which consists of around 200 features and creates a word vector. This word vector was tested on various ML classifiers like SVM, KNN, Gradient Boost. NER and Text Summarization was ranked and prioritized as compared to LDA Topic Modelling, Doc2Vec models in Coarse Grain Analysis. Finally, both granularity concepts [1] were merged, applied on isolated datasets as well as on grouped news data articles.

Keywords Granularity - Fine Grain - Coarse Grain - Remix - Empath - Data Analysis - Vader - LDA - Text Summarization - Named Entity Recognition - Doc2Vec - CovidNews - Politifact - WorldNews - Efficiency

Table of Contents

List of Figures

List of Tables

List of Acronyms

SVM Support Vector Machine

KNN KNearest Neighbours

LDA Linear Discriminant Analysis

ANN Artificial Neural Network

NLTK Natural Language Toolkit

LDA Latent Dirichlet Allocation

NER Named Entity Recognition

CNN Convolution Neural Network

DM Distributed Memory

DBOG Distributed Bag of Words

Chapter 1

Introduction

1.1 Definition/Applications

News varies from a simplistic review or a comment on social sites to a rumor or fake data. Its analysis and getting fruitful results can assist everyone around. E-commerce shopping and retails have become much more common now. Being profit-driven, sellers tend to spam the review to which legitimate businessmen fall in their prey. Its also true that fake news and its propagation had a non-negligible influence in politics, industries, and specific markets. In this report, we experimented on the possibility to detect fake news on the basis of granularity, i.e. Coarse and fine-grained features, aiming at textual data.

1.2 Motivation

The prevalence of fake news has increased with the rise of social media. Social media for news consumption is a double-edged sword. On the one hand, its low cost, easy access, and rapid spread of information allow users to consume and share the news. On the other hand, it can be used to spread fake news, i.e., news with false information. The rapid spread of fake news has the potential for calamitous impacts on individuals and society. For example, the most popular fake news was more widely spread on Facebook and Twitter than the most popular authentic mainstream news during the U.S. Presidential election [2]. Therefore, fake news detection has attracted increasing attention from researchers to politicians.

Fake news detection on social media has unique characteristics and presents new challenges. First, fake news is intentionally written to mislead readers to believe false information, which makes it difficult to detect based on news content. Thus, we need to include auxiliary information, such as user social engagements on social media, to help differentiate it from the true news. Second, exploiting this auxiliary information is nontrivial in and of itself as users social engagements with fake news produce data that is big, incomplete, unstructured, and noisy. This quick guide is based on a recent survey that presents issues of fake news detection on social media, state-of-the-art research findings, datasets, and further directions.

Fake news itself is not a new problem, and the media ecology has been changing over time from newsprint to radio/television, and recently online news and social media. The impact of fake news on traditional media can be described from the perspective of psychology and social theories. For example, two major psychology factors make consumers naturally vulnerable to the fake news: (i) Naive Realism: consumers tend to believe that their perceptions of reality are the only accurate views. (ii) Confirmation Bias: consumers prefer to receive information that confirms their existing views.

1.3 Contribution

Implemented the granularity aspect wrt. Fine Grained features and Coarse grained features. Data pre-processing and cleaning along with training of various models having different architecture backed up with literature survey was carried out. Detailed in Chp. 3 of the report.

1.4 Objectives

Detecting dishonest behavior of retailers can make an impact and maintain the social trust on such applications. To proceed with, we used various topic models to detect such fake news on the basis of granularity. The main objective would be to divide the attributes into respective defined granularity [3] and apply Machine Learning techniques individually on them. The resultant will be combined and fed into another model and the final result gives us the statistics and analysis, beyond which we can infer further details.

1.5 Organization

This report is divided into 5 chapters.The first chapter provides the introduction to the topic.The details of fake news and techniques used in its detection.The second chapter lists out the previous work done in fake-news detection and techniques which can be employed for detection.The third chapter gives the detailed description of the framework proposed and the methods involved in it. Fourth chapter gives the detailed simulation and results derived and finally fifth chapter contains conclusion and future work.

Chapter 2

Literature Survey

Fake news is intentionally written to misguide readers and leads to unexpected consequences. To detect such news, be it on social media, text contents, video streams, or image snapshots, many techniques have been researched and developed.

2.1 Past Works/Researches

Markines et al. (2009) highlighted six techniques of tagging system [4] like TagSpam, TagBlur, DomFp, NumAds, Plagiarism, ValidLinks. On the basis and reference of proposed features, Adaboost algorithm was implemented. They used several algorithms together and combined them using multi-voting methodologies.

To detect Deceptive reviews **Ning Cao et al. (2020)** used coarse and fine-grained features [1]. To verify the effectiveness and performance of this framework, typical LDA-BP + TextCNN model, explicit fine-grained feature mining models Unigram and POS, as well as excellent deep learning-based implicit feature mining models such as TextCNN, LSTM, and Bi-LSTM are selected and compared where LDA-BP + TextCNN model gains the best performance on the balanced/unbalanced Yelp datasets. Here LDA-BP + TextCNN was used for extracting coarse and fine-grained features while SVM was used as a classifier.

Qazvinian et al. (2011) proposed three features to identify rumors and also to identify users who believe the rumor and further spread it [5]. The three features were content-based, network-based, and twitter specific memes. For the experiment, the author collected 10,000 annotated tweets from twitter and achieved 0.95 in Mean Average Precision.

Gupta et al. (2013) analyzed the propagation of false information on Twitter regarding the 2013 Boston Marathon Bombings. To do so, they collect 7.9M unique tweets by using keywords about the event. Using real annotators, they annotate 6% of the whole corpus that represents the 20 most retweeted tweets during this crisis situation. Their analysis indicates that 29% of the tweets were false and a large number of those tweets were disseminated by reputable accounts whereas 51% were generic opinions and comments. Furthermore, they note that out of the 32K accounts that were created during the crisis period, 6 thousand of them were suspended by Twitter [6]. indicating that accounts were created for the whole purpose of disseminating false information.

Ahmed et al. (2017) proposed a fake news detection system which uses Term Frequency-Inverse Document Frequency (Tf-Idf) and n-gram analysis as feature extraction techniques [7]. In this paper, two different feature extraction techniques and six different machine classification techniques (SVM, LSVM, KNN, DT, SGD, Logistic Regression) are used for investigating the datasets and obtain the results.

Chen, Konroy and Rubin (2015) classified the news into three types named Type A-Serious Fabrication, Type B-Large Scale Hoaxes, Type C- Humorous Fakes [8]. Serious Fabrications or Tabloids focused on sensational crime investigations , exaggeration. Type B was based on deliberate false and deceive audiences masquerade as news. Type C used to classify news on humor and identify originating sources.

Chhabra (2011) proposed a malicious website detection method which uses URL static feature based detection with accurate results. The author has taken data from a phishing platform named as 'Phishtank which contains malicious URLs. The author has considered features such as IP addresses, a vector construction VSM[9] is chosen as the URL vector model.

Ethan Fast et al. delivered methods to classify large-scale texts into lexical categories based on semantics, emotions, reviews. Empath [10] with around 200 features which are fine grained features for given sentences. This tool was used to explore reviews on hotels, sentiment analysis of tweets, etc. This tool analytics was compared with LIWC (Linguistic Inquiry and Word Count) [11] and found to be more appealing than other tools.

Fuller et al. [12] integrated several linguistic-based cues and previous sets. The first cue set included was the Zhou/Burgoon set comprising 14 linguistic-based cues that were found effective for deception detection and included the percentage of first-person singular and plural pronouns in the text, verb quantity, sensory ratio, temporal and spatial ratio, average word length and imagery. The second set of cues was derived from deception constructs drawn from deception theories that included sentence and word quantity, activation, certainty terms, generalizing terms, imagery, and verb quantity. The third cue set was a comprehensive set of 31 cues created by including the first two cue sets along with additional Linguistic Inquiry and Word Count(LIWC) based cues utilized by previous studies and included lexical diversity, modal verbs, passive verbs, emotiveness, exclusive terms, and redundancy.

P. Lakshmi Prasanna and Dr. Rao emphasized on the Artificial Neural Network [13] for text classification. They proposed this method to overcome poor results by linear statistical techniques like SVM, Regressions and initiate text classification using neural networks [14] . ANN could also deduce unseen connections on unseen and untrained data, thus providing an unsupervised technique with proving results.

Matthew Whitehead and Larry Yaeger [15] put forth automatically classifying human sentiment from natural language written text. Bagging, Boosting [16], improvement over linear SVM were delivered. They classified reviews into negative and positive classes which further were provided as input to model for future classification [17].

Kuai Xu et al.(2020) characterized the websites and reputations of the publishers of real and fake news articles on their registration patterns. Analyzed the similarity and dissimilarity of the real and fake news on the most important terms of the articles via tf-idf and topic modeling using LDA [18]. Explored document similarity between real, fake, or hybrid news articles via Jaccard similarity to distinguish, classify, and predict fake and real news.

T. Almeida, A. Elghafari, E. Nakamura (2019) proposed a new classification strategy that was topic-agnostic [19]. Instead of using the bag of words on a page, they explored Morphological Features, Psychological Features, Readability Features, Web-Markup Features. They performed feature selection using four different methods Shannon Entropy, Tree-Based Rule, L1 Regularization and Mutual Information and combined and normalized them to feed to the supervised learning algorithm. They experimented it with KNN, Random Forest, SVM.

Pouriyeh et al. (2017) reviewed the main approaches to the text-summarization process [20]. Extractive summarization produces summaries by choosing a subset of sentences in the original text. The tasks each summarizer performs are constructing an intermediate representation of input text, scoring the sentences, and selecting a summary based on sentences. Some topic representation approaches based on frequency methods are TF-IDF[21], Word Probability. LDA has been extensively used for document summarization. Graph-based methods and Machine learning-based methods are used to determine the important sentences to be included in the summary. Evaluation of summary is a difficult task as there is no ideal summary for the document.

Document based summarization was carried out by **Jae-Seung Shim et. al 2019** [22]. Lexrankr was applied on the dataset which provided an intuition of summary for the document. Preprocessing of dataset as well as summarized data was tried on which preprocessing techniques were applied. Modelling using SVM, PCA, TF-IDF followed by 5-fold cross validation was applied on two topics separately. Comparison between body-data and summary-based model was focused on to get the results for the same.

Jing Li et. al 2020 proposed techniques for Named Entity Recognition [23] categorizing it into traditional form and deep learning form. Rule based approaches and feature selection using supervised techniques were covered into traditional NER. Character, word, and hybrid level representations were listed under Deep Learning NER. Using encoders, convolutional neural networks on data , applying LSTM model to predict future was one of their prioritization.

Ritter et.al. (2011) emphasized building an NLP pipeline with parts of speech tagging, through chunking to Named Entity Recognition [24]. It was found that classifying named entities in tweets is a difficult task. Experiment with trained NLP models for tweets the accuracy was dropped. So the proposed solution was a supervised approach with Topic Models and LabeledLDA is applied. This approach increases F1-Score by 25%.

Alexandra Savelieva et. al 2020 summarized spoken and written instructions with BERT [25]. The summarized data were classified into Type, Genre, Style Transformations, input files, vocabulary, format. Datasets used were youtube playlists on which summarization was applied using BERT. Preprocessed data is converted to PT format (for BERT algorithm interpretability). Quality analysis and assessment of generated summaries were put forth.

Mahesh Korlapati et. al used LDA topic modelling to classify documents into topics [26]. LDA considers a document as a mix of various topics and each word belongs to one of the document's topics. They proposed a method to categorize research papers to topics using LDA for text categorization. Using Cora Dataset they observed that LDA model is better than the conventional Naive Bayes and Support Vector Machines in the field of text classification.

Authors	Paper Title	Model	Dataset	Features	Advantage	Future Work
Markines et al. (2009)	Social Spam Detection using Adaboost	SVM, Ada-Boost	Spam posts on social media, tags.	TagSpam, TagBlur, DomFp, NumAds, Plagiarism, ValidLinks	Use of Boosting is better than bagging to classify.	Include multiclass classification rather than binary.
Ning Cao et al. (2020)	A deceptive review detection framework	LDA-BP + TextCNN + SVM	Yelp Datasets	Fine-grained and coarse-grained features	Proposed model out-performed TextCNN, LSTM, and BiLSTM	Improvement Possible by integrating other better deep neural network algorithms and coarse-grained algorithms
Qazinian et al. (2011)	Identified tweets in which rumor is endorsed.	Nave Bayes	Tweets	Content-based, network-based, Twitter-specific memes	Rumor analysis with 0.95 Mean Average Precision	To build a system to identify whether a trending topic is a rumor or not
Gupta et al. (2013)	Analysis of Twitter content during Boston Marathon.	Logistic Regression	Tweets and user information	Topic engagement, Global engagement, Social reputation, Likability, Credibility	To find out fake Twitter accounts created amid high impact news events	Improvement possible by using a decision tree can add culture affected feature.

Authors	Paper Title	Model	Dataset	Features	Advantage	Future Work
Ahmed et al. (2017)	Detection of Online Fake News Using N-Gram Analysis and ML Techniques	SVM, LSVM, KNN,DT, SGD, Logistic Regression	News articles (Reuters), Fake news dataset (kaggle)	TF-IDF	TF-IDF out-performed n-gram analysis	Include more feature extraction techniques
Chen, Konroy and Rubin	Deception Detection and 3 Types of Fakes	NLP, Sentiment Analysis, Big Data	E-Mails, Web Crawling, Fake product reviews, publicly available social media data, Law enforcement data	Type A - Serious Fabrication, Type B - Large Scale Hoaxes, Type C - Humorous Fakes	Broader analysis of news data along with fake classification	Classify the divisions further based on cultures and religions
S. Chhabra et al. (2011)	Fake and Malicious URL Detection	Naive Bayes, Logistic Regression, DT, SVM-RBF, SVM-Linear SVM-Sigmoid	Malicious URL dataset from ‘Phishtank’	Grammar, Lexical, Vectors and Static.	proposed URL static feature based detection method	Can use binary classification through Adaboost algorithm.

Authors	**Paper Title**	**Model**	**Dataset**	**Features**	**Advantage**	**Future Work**
Ethan Fast, BinBinbin Chen, Michael Ber-stein	Empath: Understanding Topic Signals in Large-Scale Text	Empath and LIWC	Hotel Reviews, Movie Reviews, time analysis of mood on Twitter, deception data	Text classification, neural network training, 200 in-built features	Simplified fine-grained classification	More qualitative aspect of classification, include fiction reviews.
P. Lakshmi Prasanna, Dr. Rao	Text classification using artificial neural networks	ANN, Document conversion, stemming	Multi-dimensional datasets From medical background	TFIDF Matrix from text classification.	ANN is better than linear statistical techniques	Implement NN on text.
Matthew Whitehead, Larry Yaeger	Sentiment Mining Using Ensemble Classification Models	Bagging, Boosting [12], single model SVM, K-fold(10) cross validation	Dataset by Snyder and Barzilay, Amazon, layerratingz, tvratingz	Ensemble models give much better accuracy	Out-performed SVM model	Improvement in time complexity
Kuai Xu, Feng Wang, Haiyan Wang, Bo Yang	Detecting Fake News Over Online Social Media via Domain Reputations and Content Understanding	LDA Topic Modelling	Major News Outlets - NewYork Times, NBC News Washington Post, and the USA Today	TF-IDF	Uses Document similarity to distinguish real and fake news	Word embedding of the important words found in Tf-idf

Authors	**Paper Title**	**Model**	**Dataset**	**Features**	**Advantage**	**Future Work**
Castelo et al (2019).	A Topic-Agnostic Approach For Identifying Fake News Pages	SVM, KNN, Random Forest	Political News	Morpho logical, Psycho logical, Readability, Web-Markup Features	Uses Four Different Feature Selection Measures combined and normalize them.	Improvement possible by using ensemble learning algorithms like Gradient Boosting.
Jae-Seung Shim et al	Document Summarization Technique on the Fake News Detection Model	PCA, SVM, Regression, Decision Tree	Full text of a news article.	Lexrankr to get 3 line summary.	Overfitting of various subjects was uncovered using complete document summarization.	Can improve performance using fine grained embedding techniques.
Jing Li et. al	A Survey on Deep Learning for Named Entity Recognition	CNN, LSTM, encoder, Tag Decoder.	Data articles	Apply Deep Learning which has its own significance along with NER for tagging.	Applied Deep Neural Nets to learn along with NER.	Fine-Grained NER and Boundary Detection. Introduce unknown entities.
Ritter et.al	Named Entity Recognition in Tweets:An Experimental Study	NER	Twitter Tweets	Postagging, Shallow Parsers,LDA	The F1 score for tweets data was improved compared to other trained NLP models.	Build a model working for both news data and tweet data

Authors	Paper Title	Model	Dataset	Features	Advantage	Future Work
Savelieva et.al	Abstractive Summarization of Spoken and Written Instructions with BERT	Text summa--rization	Youtube	NLP, BERT, Neural Network.	Quality Analysis of Summarized Data. Check-in with Version Control for future reference.	Test it on various datasets.

Table 2.1: Literature Review

Chapter 3

Proposed Algorithm & Implementation

Coarse Grained features are explicitly defined as overall data in the text which has a tendency to split enough. The smallest possible meaningful content in a topic model can be a word which defines **Fine Grained** features. When combined together define the coarse grained features being a superset. High level block diagram is explained thoroughly in further sections.

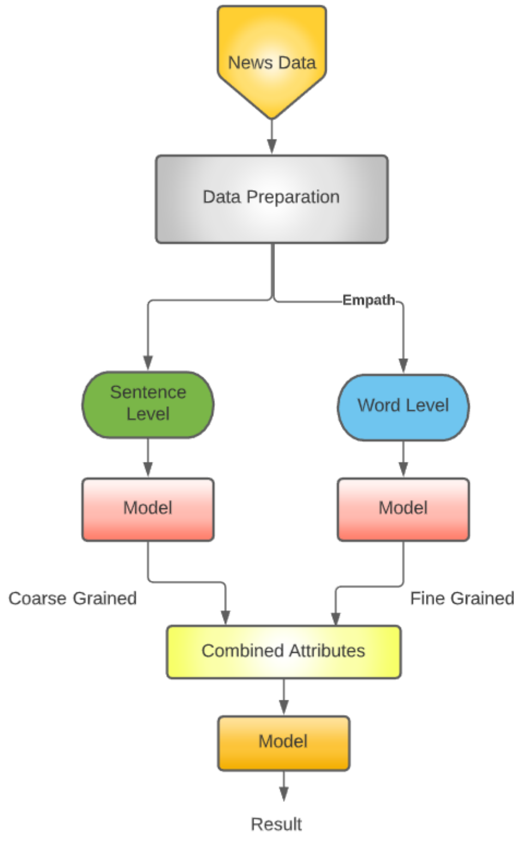


Fig 1. Proposed Framework

3.1 High Level Block Diagram

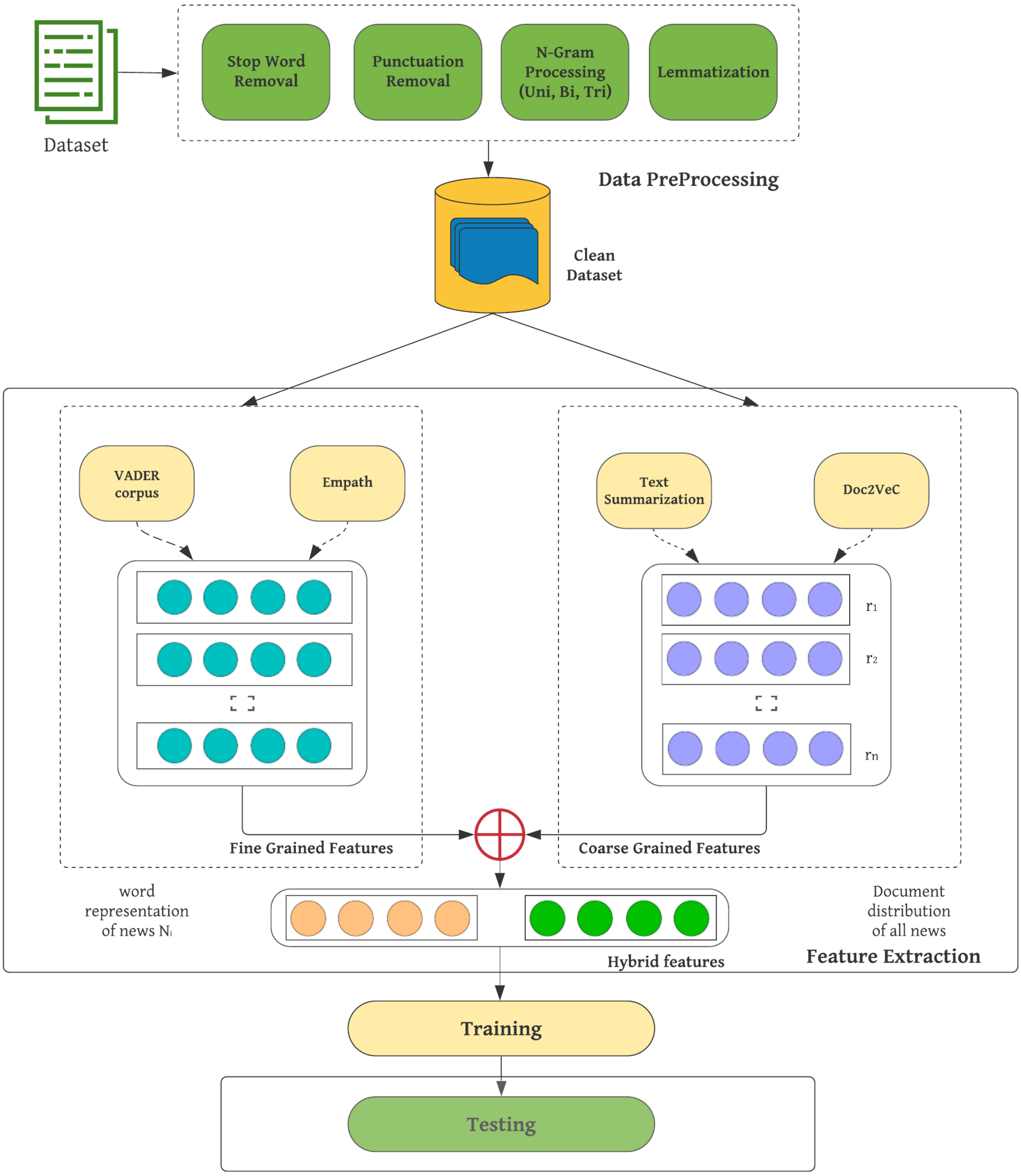


Fig 2. Architecture of Block Diagram

The dataset is initially processed as described in above sections. Classification based on granularity is imposed and features are extracted on the basis of proposed algorithms. Fine Grained features are retracted using **Empath** library of Python. Further modifications of dataset is carried out. Training the dataset on models detailed above is carried out. Coarse Grain analysis using **LDA Topic Modelling, Named Entity Recognition, Text Summarization** was undertaken. Results of both granularity concepts are combined and stacked together. Thus, word level and sentence level attributes play a vital role to analyse the news data. The resultant vector of attributes are trained using Supervised ML technique to further emphasize on best analysis so as to avoid deceptions.

3.2 Development of Algorithm

3.2.1 Dataset Description

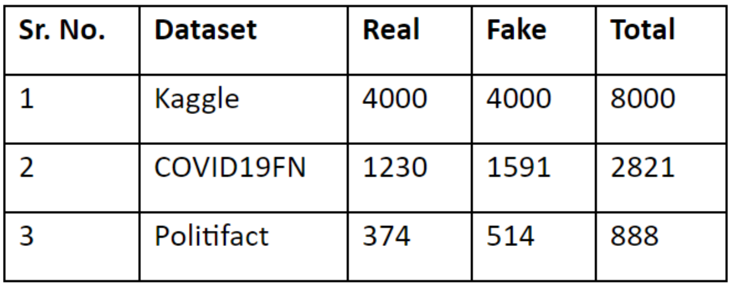
The experimentation is carried out on three standard publicly available datasets:
(i) KaggleNewsDataset (ii) COVID19FN (iii) PolitiFact. (iv) Mixed

1. This dataset consists of about 8000 articles consisting of fake as well as real news. The fake and real news data is given in two separate datasets with each dataset consisting around 4000 articles each. The features of the dataset are title, news subject, date, text.

2. COVID19FN consists 1591 Fake news articles and 1230 actual news articles on the novel coronavirus pandemic. The fake news articles are from fact-check engine Poynter while the real ones have been obtained from well-known sources such as FactCheck, Observador, Snopes and the like. For each article, the dataset contains several features like titles, text, date published, country and source URL. This work focuses on providing spatial and temporal features for classification.

3. Politifact contains Fake news articles from US presidential elections. This is from where the fake news originated. It consists of 514 fake and 374 true instances. The dataset includes attributes such as authors, canonical source link, images, publish date, text and label.

4. Combining the above dataset as described, we applied all the defined algorithms. The dataset consist of around 13700 news articles.



Sr. No.	Dataset	Real	Fake	Total
1	Kaggle	4000	4000	8000
2	COVID19FN	1230	1591	2821
3	Politifact	374	514	888

Table 3.1: Dataset Description

3.2.2 Dataset Pre-Processing

Data preprocessing [27] is one of the most required steps in data analysis in order to achieve maximum accuracy and throughput . It includes techniques to remove incomplete data, making data consistent and ready to use for experiments. Mostly, library called pandas [28] is used for such preprocessing.

Preprocessing text simply means to convert text into a form that is predictable and analyzable for given task. Various steps involved in Data Preprocessing are Data Cleaning, Data Transformation, Data Reduction.

Data Cleaning involves handling of missing data, noisy data. Strategies to handle missing data involve removing the tuples, filling the missing values. In noisy data Lowercasing, Stemming, Lemmatization, Stop words removal, outlier analysis can be done to clean irregular and inconsistent data.

Lowercasing is one of the simplest and most effective form of text preprocessing. **Stemming** is the process of reducing inflection in words (e.g. changing, changed, changer) to their root form (e.g. change).The "root" in this case may not be a real root word, but just a canonical form of the original word. **NLTK** provides implementation of stemming. Stemming is desirable as it may reduce redundancy as most of the time the word stem and their derived words mean the same.

Lemmatization is very similar to stemming, where the main aim is to remove inflections and map a word to its root form. The only difference is that, lemmatization tries to do it properly. It doesn't just remove things off, it links words with similar meaning to one word.

Stop words are a set of commonly used words in a language. Examples of stop words in English are articles, etc. The intuition behind removing stop words is that, we can focus more on the important words instead.

Noise removal is about unnecessary punctuations and white spaces that can interfere with your text analysis.

Data Transformation is used to transform the data in appropriate form which makes it usable for data mining process.There are different ways to transform your text also for each task different methods are followed.

Text Normalization is the process of transforming a text into a canonical (standard) form. For example, the "3" can be transformed to "three", its canonical form. It is a highly overlooked preprocessing step.

Data Reduction is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis becomes harder. In order to get rid of this, we uses techniques like Dimensionality Reduction, Feature Selection, etc. It aims to increase the throughput, efficiency and reduce data storage and analysis costs.

Existing rows that contain irregular and incomplete data were identified and removed, so as to manipulate the dataset [29]. The rows of the dataset were shuffled. The dataset was further divided into 70% training data and 30% validation data.

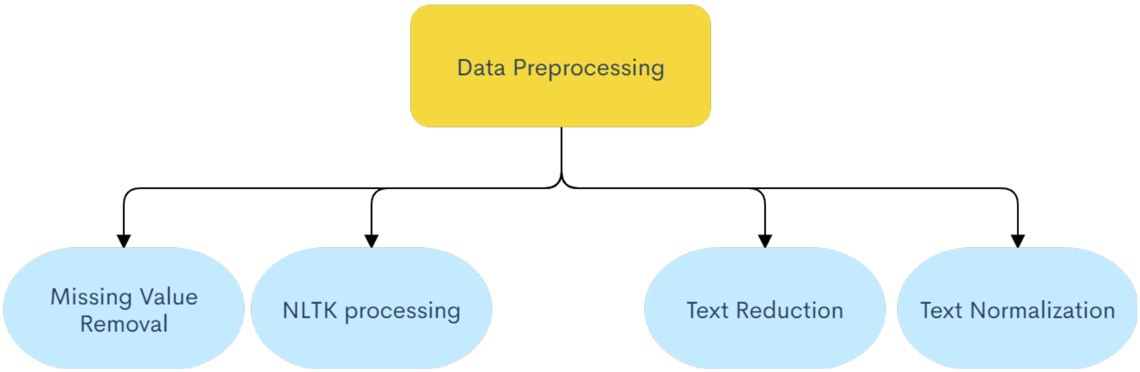


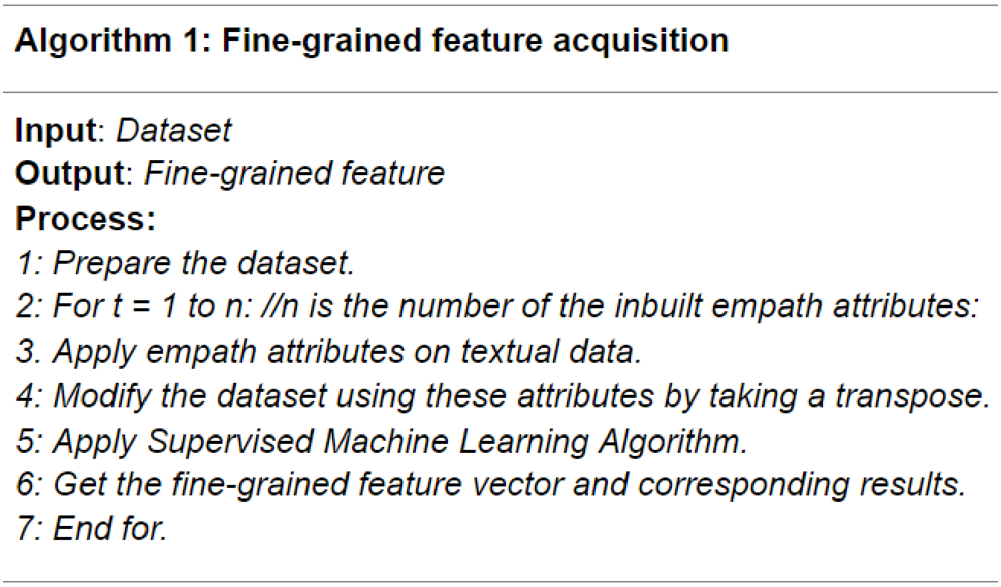
Fig 3. Data Processing Techniques

3.2.3 Fine Grained Analysis

Empath is a tool for analyzing text across lexical categories, and also generating new lexical categories to use for an analysis. It contains 194 inbuilt attributes. Semantic meaning of text can easily be represented in the form empath feature vector.

Vader is a key attribute based on semantics. Textual data can be classified as positive, negative or neutral classes depending on the keywords present in the article. Vader Score, thus can be a feature which can be used for **"News Data Analysis"**.

These attributes represent the columns of the modified dataset with rows as the Empath and Vader values of the existing data. A new attribute named Category is appended to the modified dataset which can further be useful as an input to supervised machine learning techniques.



Algorithm 1: Fine-grained feature acquisition

Input: *Dataset*
Output: *Fine-grained feature*
Process:

```
1: Prepare the dataset.
2: For t = 1 to n: //n is the number of the inbuilt empath attributes:
3. Apply empath attributes on textual data.
4: Modify the dataset using these attributes by taking a transpose.
5: Apply Supervised Machine Learning Algorithm.
6: Get the fine-grained feature vector and corresponding results.
7: End for.
```

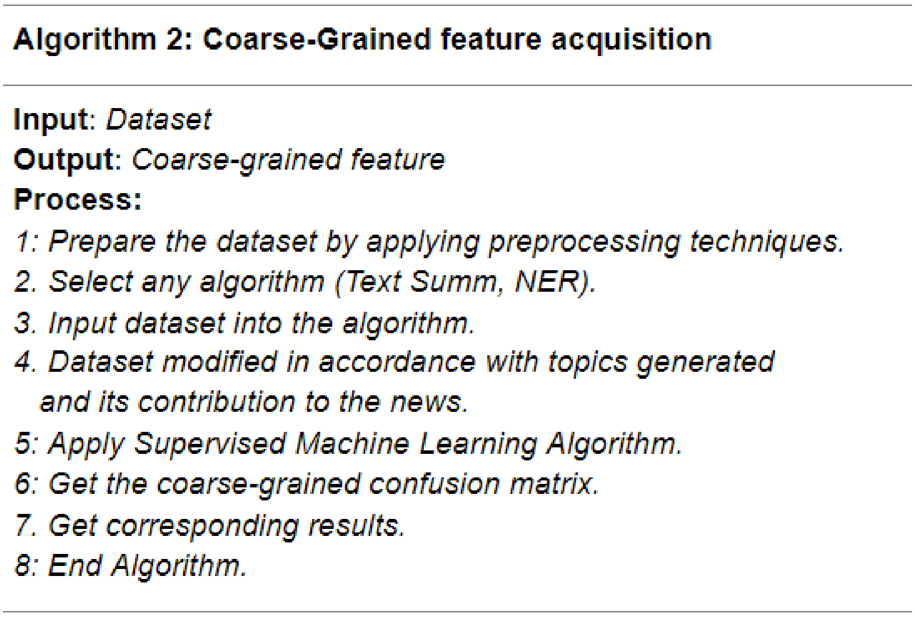
3.2.4 Coarse Grained Analysis

NER - Named Entity Recognition is a NLP technique which is used to identify entities like person, organizations, country, etc. It majorly uses spacy library of python for its implementation. Analysis of a text and contribution of entities in the news classes like fake and true gives a clear idea of tagging on text. It uses large volumes of twitter texts, unstructured data, emails, feeds, etc. to predict named entities in a given corpus or sentence. It is a Coarse Grain Feature unlike Empath since it takes complete corpus at once as input. Performing POS tagging, it produces tokens for the input sentences and finally a doc which is also referred to as processing pipeline.Each pipeline uses default model consisting of tagger, a parser, and an entity recognizer. Each pipeline component returns processed doc which is passed on next component for model training. Linear SVM outsttod the perfomance as compared with other training models with an accuracy of 94.9 %.

Text Summarization is a technique of reducing long pieces of text to create a summary of the text having the main points outlined. Text Summarization techniques are classified into Extractive methods and Abstractive methods. Extractive text summarization involves selecting phrases and sentences from source documents to make the new summary. Abstractive text summarization involves generating new phrases and sentences that have the same meaning as the source document. The Gradient Boosting algorithm gives an accuracy of 92.2 %.

LDA Topic Modeling is a part of statistical modeling for identifying hidden topics from the collection of documents. Latent Dirichlet Allocation (LDA) is the most popular topic modeling technique. LDA presumes documents are produced from a mixture of topics. LDA builds topic per document model and words per topic model. Provided number of topics, it will rearrange keywords distribution within topics and topic distribution within documents to get a good formation of topic-keyword distribution. Dictionary and the corpus are the two main inputs to the LDA topic model. Pythons Gensim package is an excellent implementation for LDA. LDA is primarily used for the analysis of topics that are inferred from the given corpus. The simulations and results are further detailed in the upcoming chapter.

Doc2Vec model was applied on two parameters, ie. DM and DBOG [30]. In brief, DM acts as a storage to remember essence of paragraphs and typical topics. DBOG uses paragraph vectors. Once these models are ready, they both are compiled together to increase the efficiency. This is also known as **Model Pairing**. The paired model is used for classification of news articles.

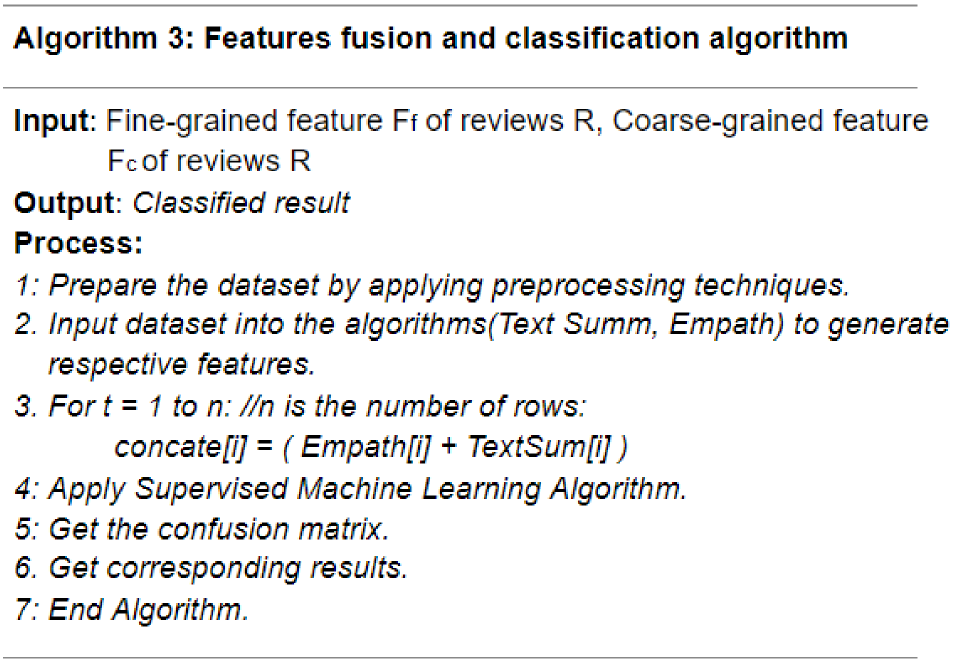


Algorithm 2: Coarse-Grained feature acquisition

```
Input: Dataset
Output: Coarse-grained feature
Process:
1: Prepare the dataset by applying preprocessing techniques.
2. Select any algorithm (Text Summ, NER).
3. Input dataset into the algorithm.
4. Dataset modified in accordance with topics generated
   and its contribution to the news.
5: Apply Supervised Machine Learning Algorithm.
6: Get the coarse-grained confusion matrix.
7. Get corresponding results.
8: End Algorithm.
```

3.2.5 Fine-Coarse Fusion

The preprocessed dataset when modelled over previously defined Fine and Coarse Grain algorithms yield features. As far this report is concerned, Empath Analysis (Fine) and Text Summarization (Coarse) attributes are combined together for each news article. A fused model is created on which ML algorithms briefed below are applied to get the resultant classifications.



Algorithm 3: Features fusion and classification algorithm

```
Input: Fine-grained feature $F_f$ of reviews R, Coarse-grained feature
       $F_c$ of reviews R
Output: Classified result
Process:
1: Prepare the dataset by applying preprocessing techniques.
2. Input dataset into the algorithms(Text Summ, Empath) to generate
   respective features.
3. For t = 1 to n: //n is the number of rows:
       concate[i] = ( Empath[i] + TextSum[i] )
4: Apply Supervised Machine Learning Algorithm.
5: Get the confusion matrix.
6. Get corresponding results.
7: End Algorithm.
```

3.2.6 Machine Learning Models

Various supervised machine learning models were implemented on the dataset to bridge the gap between fake and real news.

Different models such as Logistic Regression, Linear Support Vector Machine, Gradient Boosting, K-Nearest Neighbour, Naive Bayes and tree algorithms(Random Forest, Decision Tree) [31] were implemented and analyzed. Confusion matrix was built for each of the model and best suitable algorithm was found to be random forest. The accuracy [32] touched to an extreme of 90% on the testing dataset. Thus, the fine grained features should be shuffled with coarse grained analysis resulting in better prediction on news data.

3.3 Explainability Of Classes on Basis of Prediction

Explanability technique is used to extract the many insights like which features in the data are most important, how much does each feature effect the prediction from sophisticated machine learning models. To find that, concept of permutation importance is useful. It is calculated after a model has been fitted. So the model or the predictions won't change for a given value.

Here, a single column of the validation data is randomly shuffled, leaving the target and all other columns in place, and the accuracy of predictions in then checked. If the resulting data does not correspond to anything observed in the real world then randomly shuffling a single column causes less accurate predictions. If a single column that the model relied on heavily for predictions is shuffled then accuracy suffers quite a lot.

The process is as follows:

1. Get a trained model.
2. Shuffle the values in a single column, make predictions using the resulting dataset. Use these predictions and the true target values to calculate how much the loss function suffered from shuffling. That performance deterioration measures the importance of the variable you just shuffled.
3. Return the data to the original order (undoing the shuffle from step 2).
4. Now repeat step 2 with the next column in the dataset, for every column.

3.4 Summary

Finally, clear insights regarding the flow architecture, dataset preprocessing and granularity concepts is versed. Algorithms for Fine Grain, Coarse Grain and fusion of the two and its step wise implementation on various models has been discussed. The simulations and corresponding results are displayed in the next chapter in detail.

Chapter 4

Simulation and Results

4.1 Upshots

On the basis of previous discussions, implementation and its analysis was carried out on the basis of news granularity.

The analysis of type of news and its subject is displayed below. Further, they are classified into its corresponding fake and true classes. The news varied from Politics, world news to government, Middle-East, US election news.

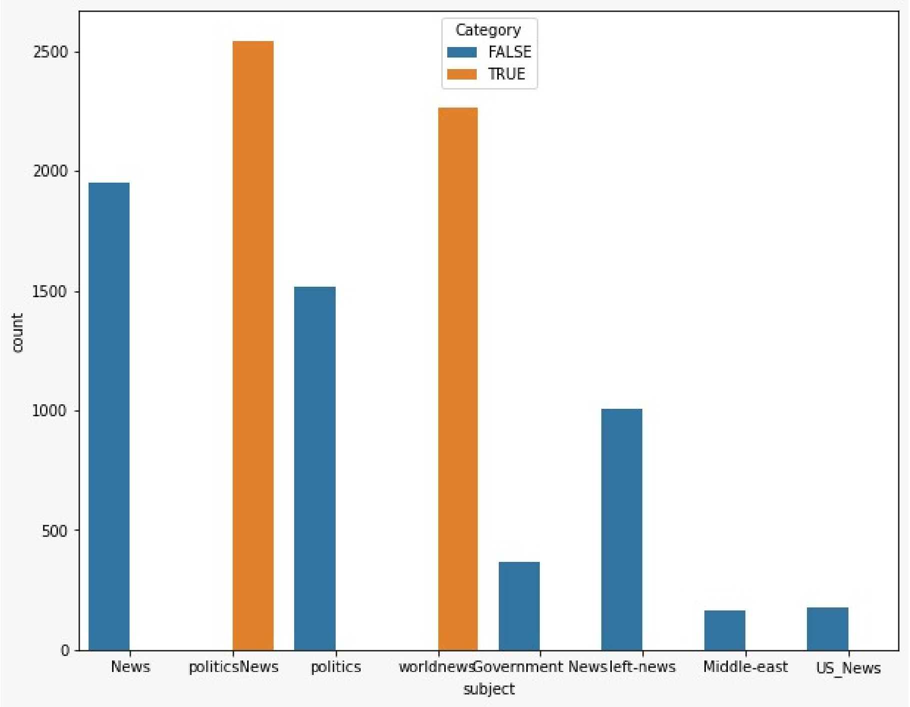


Fig 4. News Analysis

4.2 Result Analysis

4.2.1 Fine-Grained

Word Clouds for Fake and True news is displayed below. Word Cloud gives an insight of the fine words that actually represents its involvement in the classes. It is a Data Visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Word clouds are widely used for analyzing data. Larger the height of the word, more it ensembles that class.

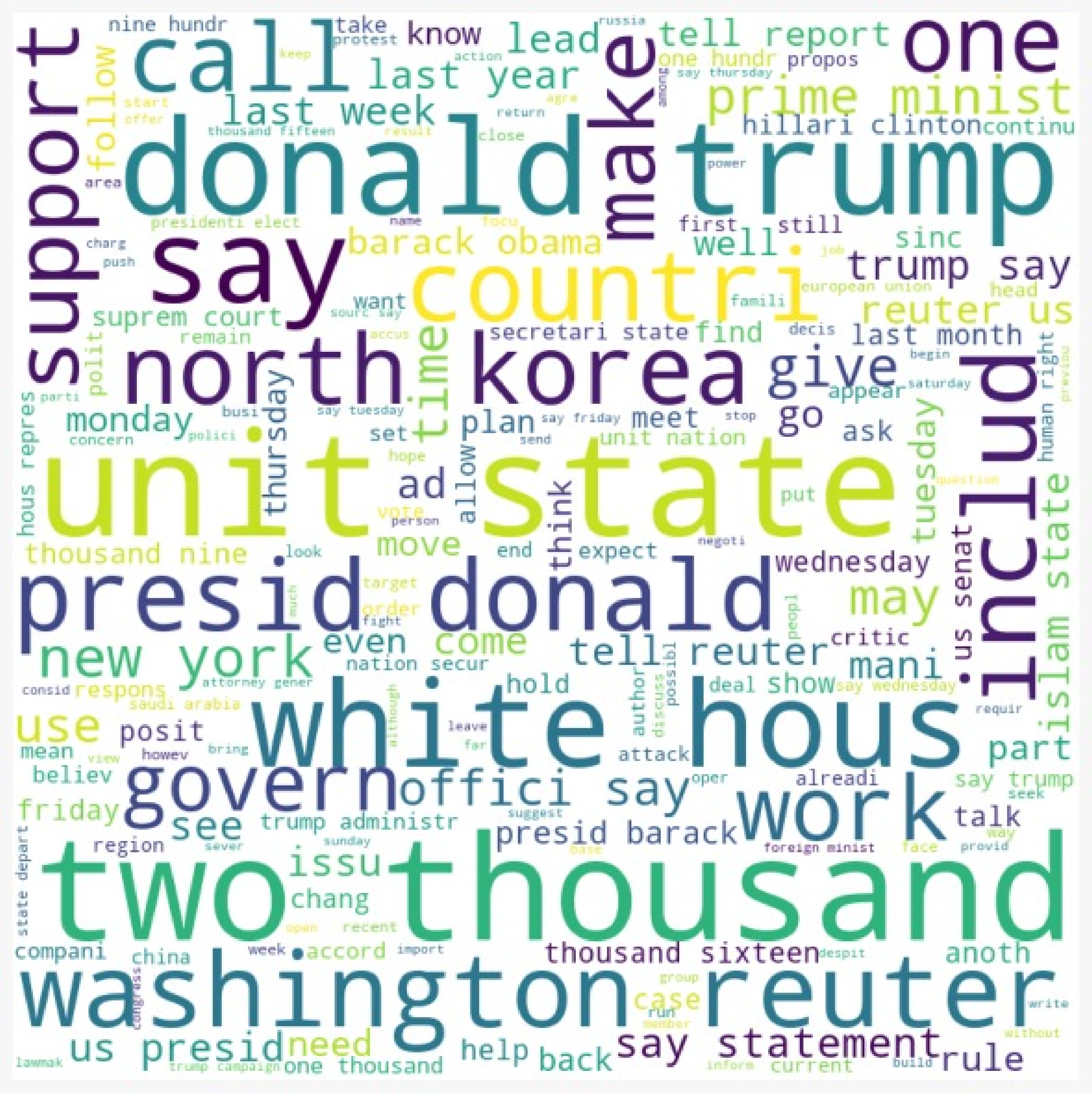


Fig 5. True News Word Cloud

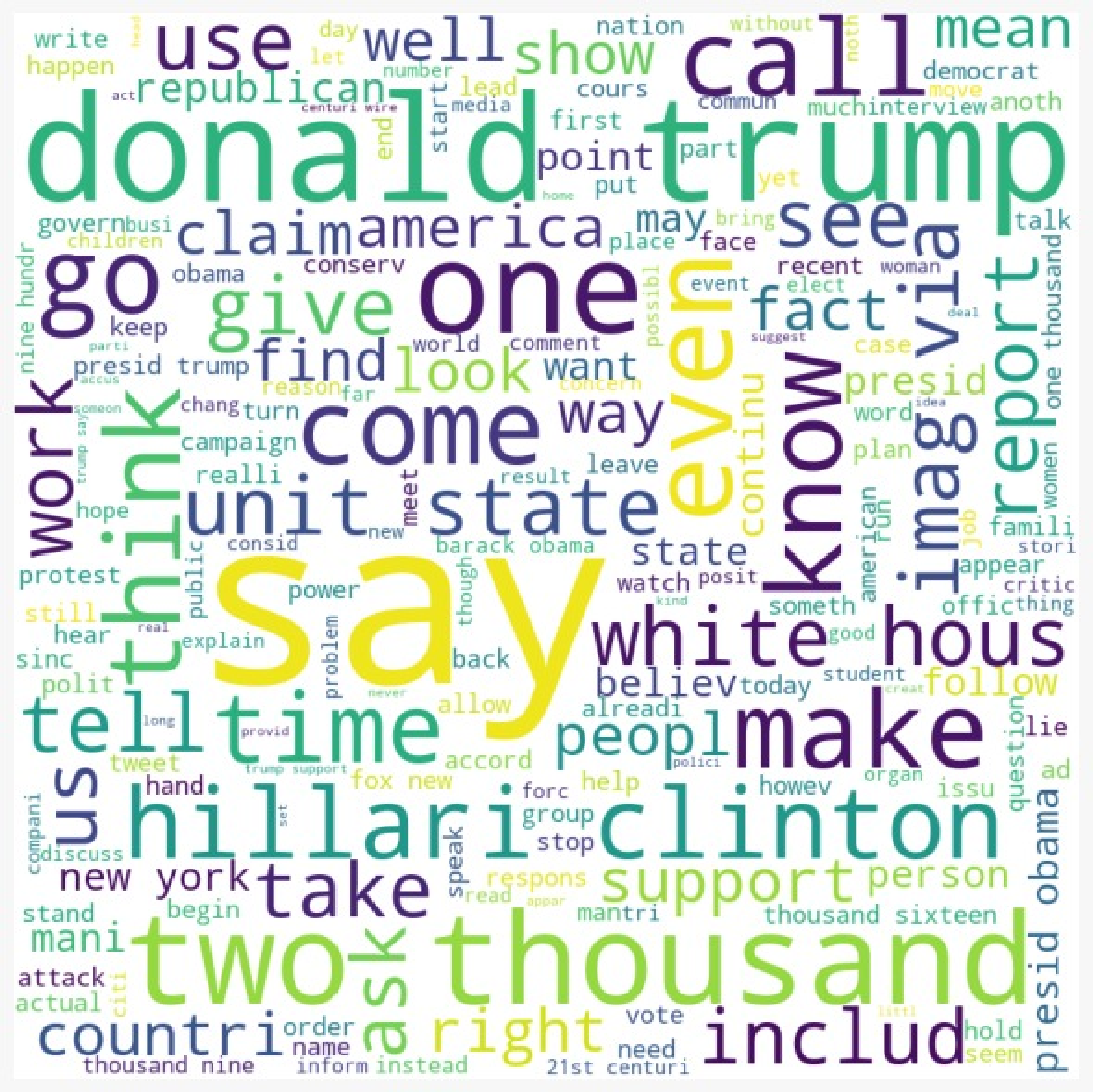


Fig 6. False News Word Cloud

4.2.2 Coarse-Grained

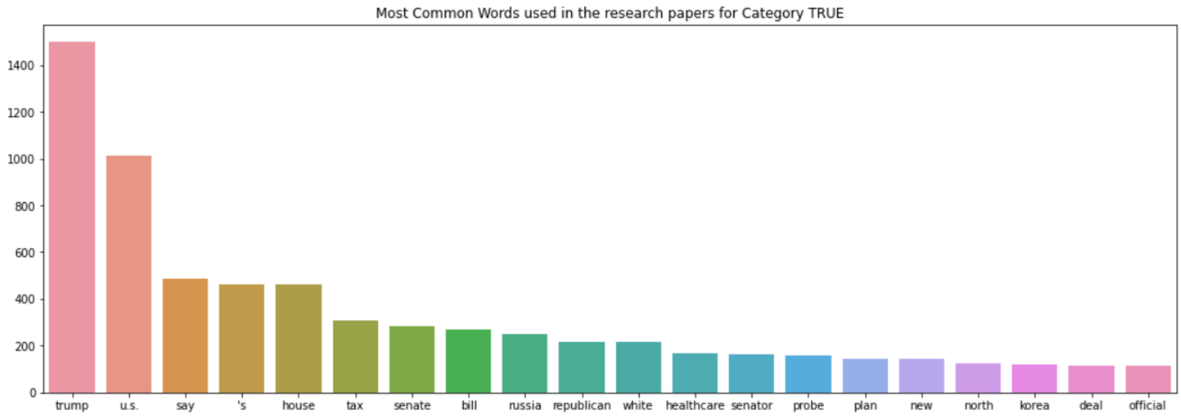


Fig 7. NER True Topics

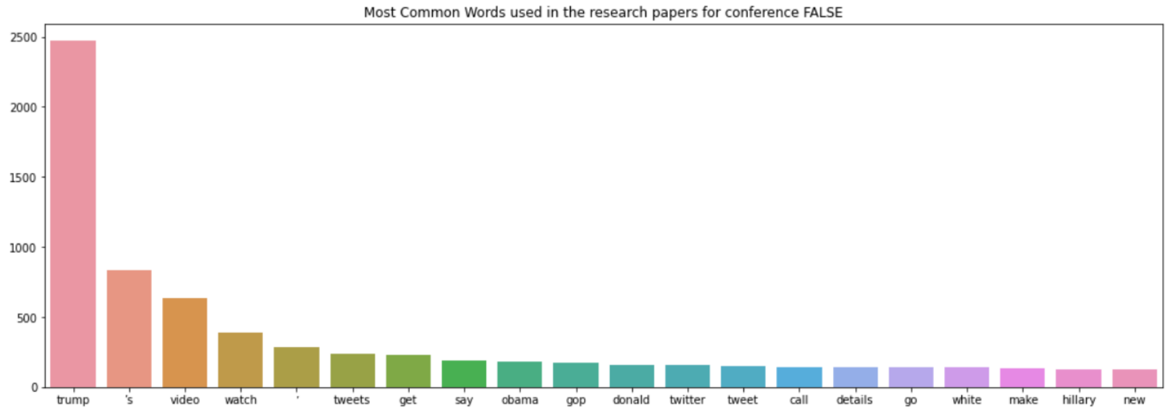
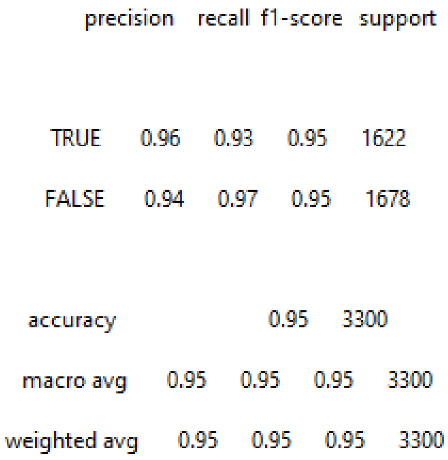


Fig 8. NER False Topics



```
              precision    recall  f1-score   support


        TRUE       0.96      0.93      0.95      1622

       FALSE       0.94      0.97      0.95      1678


    accuracy                           0.95      3300

   macro avg       0.95      0.95      0.95      3300

weighted avg       0.95      0.95      0.95      3300
```

Fig 9. NER Confusion Matrix

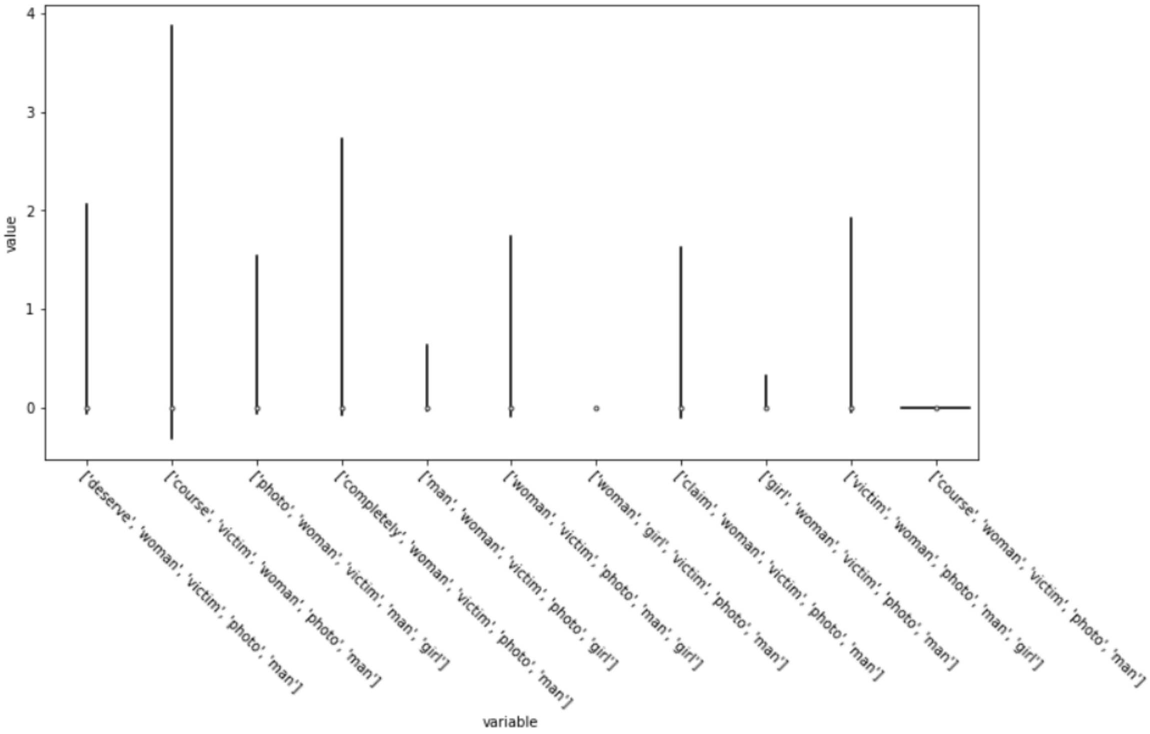
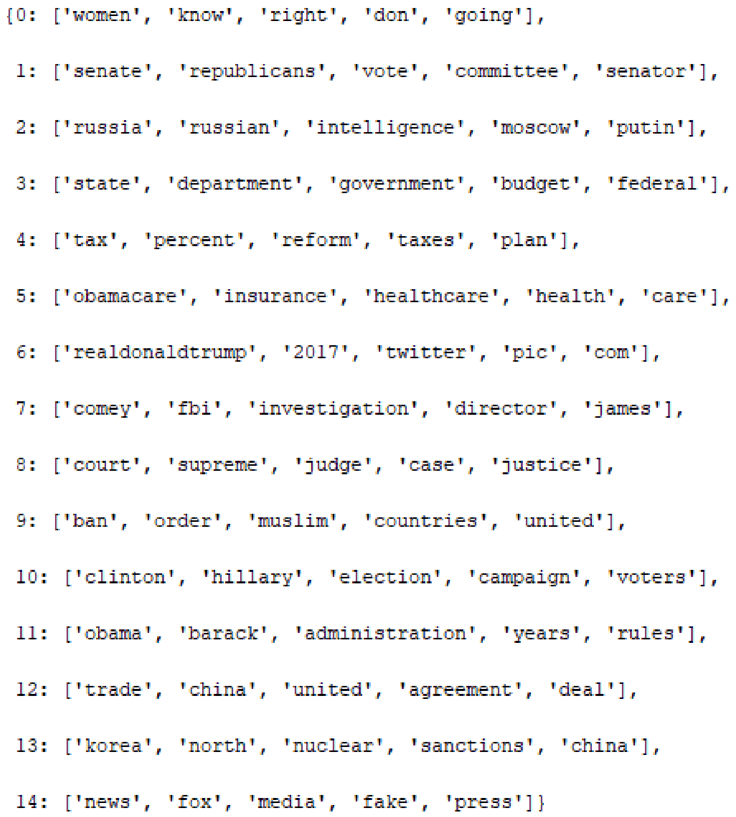


Fig 10. Text Summarization Topics



```
{0: ['women', 'know', 'right', 'don', 'going'],

 1: ['senate', 'republicans', 'vote', 'committee', 'senator'],

 2: ['russia', 'russian', 'intelligence', 'moscow', 'putin'],

 3: ['state', 'department', 'government', 'budget', 'federal'],

 4: ['tax', 'percent', 'reform', 'taxes', 'plan'],

 5: ['obamacare', 'insurance', 'healthcare', 'health', 'care'],

 6: ['realdonaldtrump', '2017', 'twitter', 'pic', 'com'],

 7: ['comey', 'fbi', 'investigation', 'director', 'james'],

 8: ['court', 'supreme', 'judge', 'case', 'justice'],

 9: ['ban', 'order', 'muslim', 'countries', 'united'],

 10: ['clinton', 'hillary', 'election', 'campaign', 'voters'],

 11: ['obama', 'barack', 'administration', 'years', 'rules'],

 12: ['trade', 'china', 'united', 'agreement', 'deal'],

 13: ['korea', 'north', 'nuclear', 'sanctions', 'china'],

 14: ['news', 'fox', 'media', 'fake', 'press']}
```

Fig 11. Text Summarization Topic Modelling

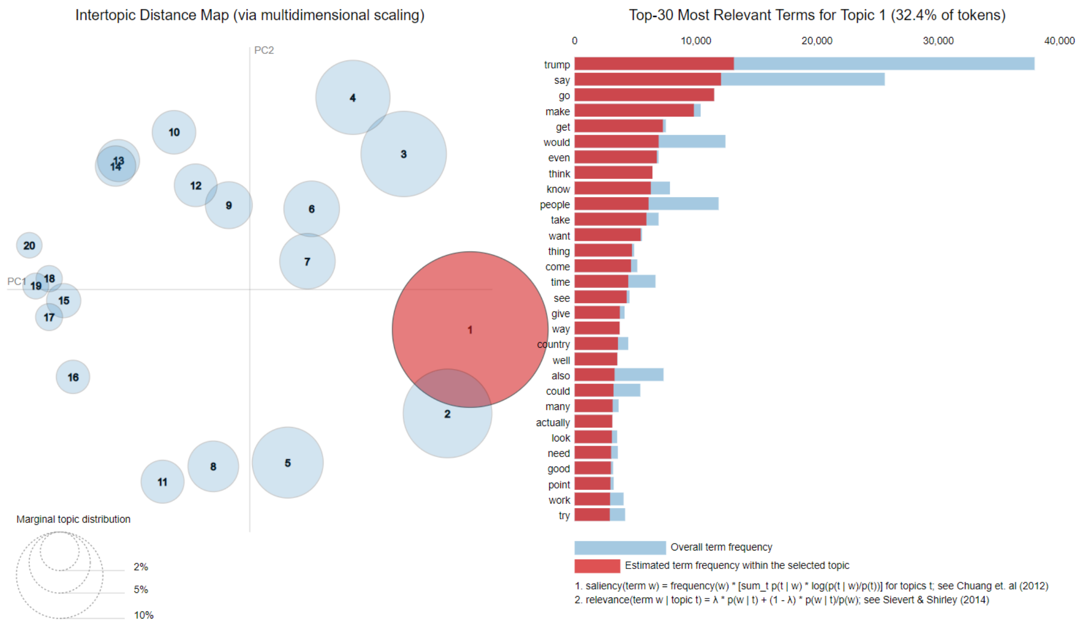
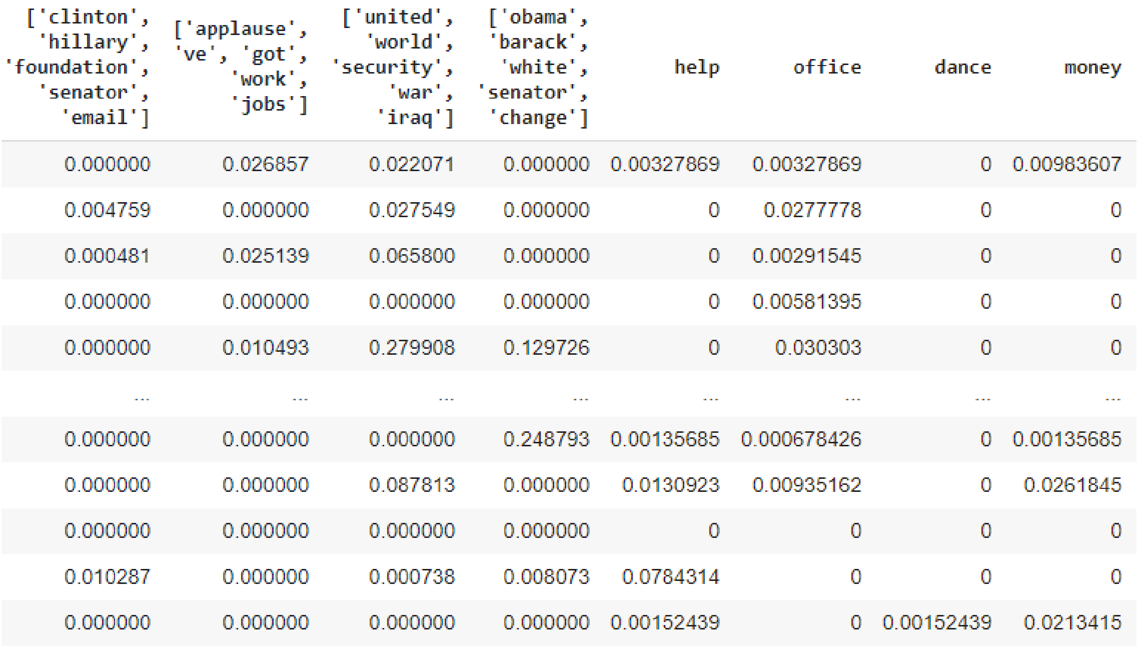


Fig 12. LDA Topic Modelling

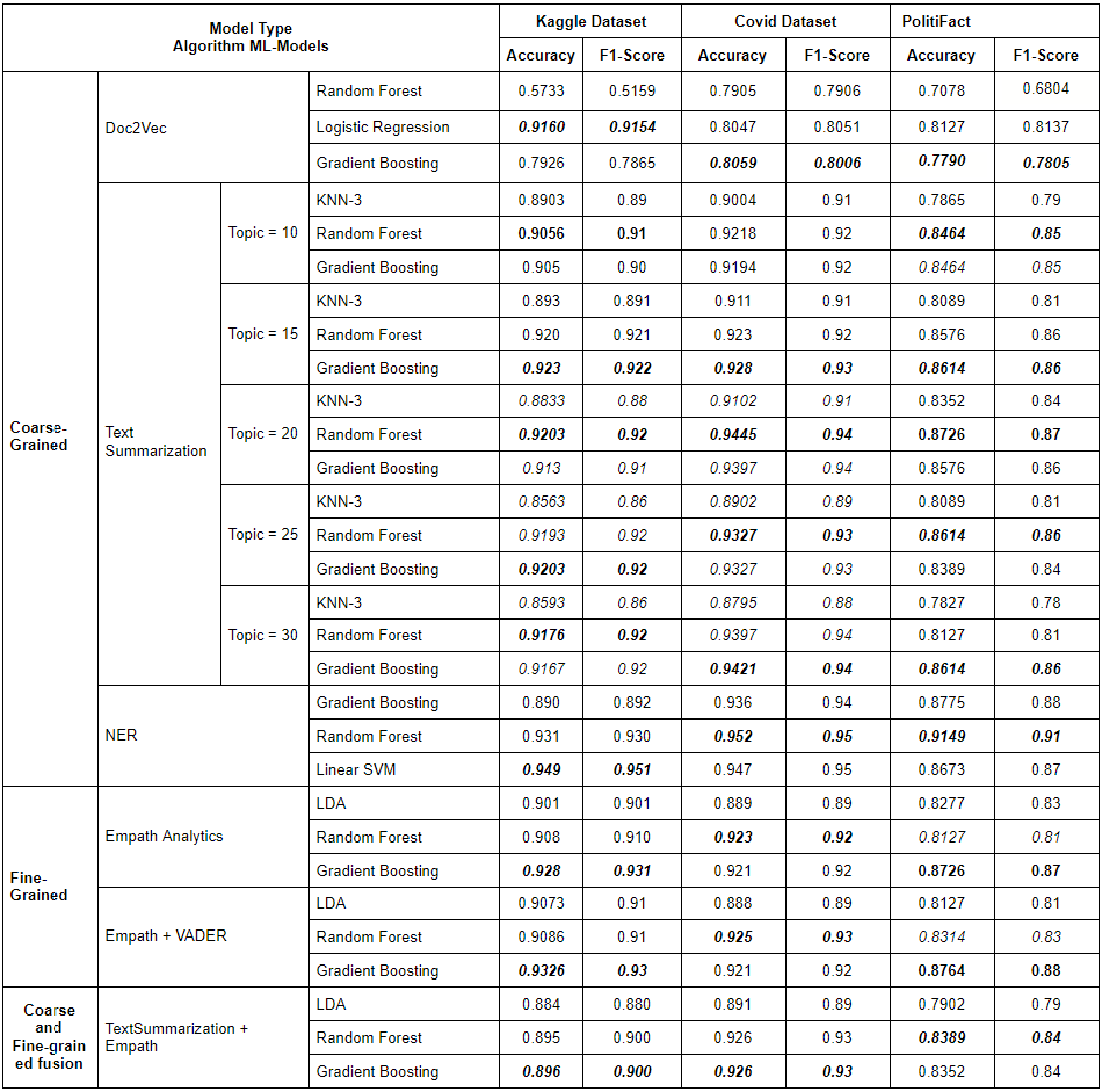
4.2.3 Fusion Fine-Coarse Features



['clinton', 'hillary', 'foundation', 'senator', 'email']	['applause', 've', 'got', 'work', 'jobs']	['united', 'world', 'security', 'war', 'iraq']	['obama', 'barack', 'white', 'senator', 'change']	help	office	dance	money
0.000000	0.026857	0.022071	0.000000	0.00327869	0.00327869	0	0.00983607
0.004759	0.000000	0.027549	0.000000	0	0.0277778	0	0
0.000481	0.025139	0.065800	0.000000	0	0.00291545	0	0
0.000000	0.000000	0.000000	0.000000	0	0.00581395	0	0
0.000000	0.010493	0.279908	0.129726	0	0.030303	0	0
...	...	...	...	...	...	...	...
0.000000	0.000000	0.000000	0.248793	0.00135685	0.000678426	0	0.00135685
0.000000	0.000000	0.087813	0.000000	0.0130923	0.00935162	0	0.0261845
0.000000	0.000000	0.000000	0.000000	0	0	0	0
0.010287	0.000000	0.000738	0.008073	0.0784314	0	0	0
0.000000	0.000000	0.000000	0.000000	0.00152439	0	0.00152439	0.0213415

Fig 13. Feature Fusion Table

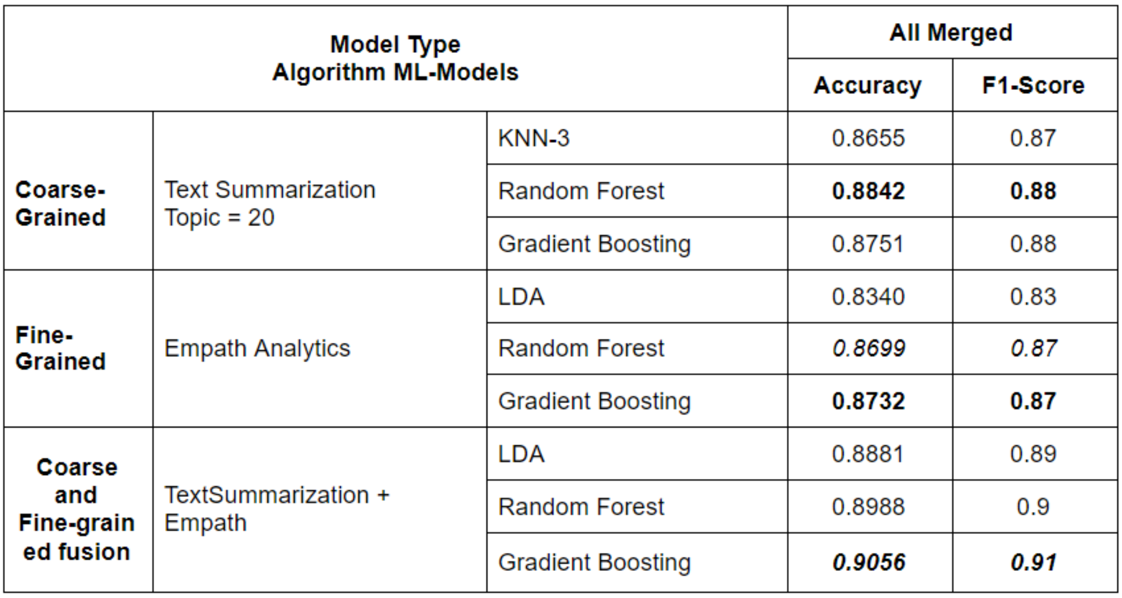
4.3 Inference Table



Model Type Algorithm ML-Models				Kaggle Dataset		Covid Dataset		PolitiFact	
				Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
Coarse-Grained	Doc2Vec		Random Forest	0.5733	0.5159	0.7905	0.7906	0.7078	0.6804
			Logistic Regression	***0.9160***	***0.9154***	0.8047	0.8051	0.8127	0.8137
			Gradient Boosting	0.7926	0.7865	***0.8059***	***0.8006***	***0.7790***	***0.7805***
	Text Summarization	Topic = 10	KNN-3	0.8903	0.89	0.9004	0.91	0.7865	0.79
			Random Forest	**0.9056**	**0.91**	0.9218	0.92	***0.8464***	***0.85***
			Gradient Boosting	0.905	0.90	0.9194	0.92	*0.8464*	*0.85*
		Topic = 15	KNN-3	0.893	0.891	0.911	0.91	0.8089	0.81
			Random Forest	0.920	0.921	0.923	0.92	0.8576	0.86
			Gradient Boosting	***0.923***	***0.922***	***0.928***	***0.93***	***0.8614***	***0.86***
		Topic = 20	KNN-3	*0.8833*	*0.88*	*0.9102*	*0.91*	0.8352	0.84
			Random Forest	***0.9203***	***0.92***	***0.9445***	***0.94***	**0.8726**	**0.87**
			Gradient Boosting	*0.913*	*0.91*	*0.9397*	*0.94*	0.8576	0.86
		Topic = 25	KNN-3	*0.8563*	*0.86*	*0.8902*	*0.89*	0.8089	0.81
			Random Forest	*0.9193*	*0.92*	***0.9327***	***0.93***	***0.8614***	***0.86***
			Gradient Boosting	***0.9203***	***0.92***	*0.9327*	*0.93*	0.8389	0.84
		Topic = 30	KNN-3	*0.8593*	*0.86*	*0.8795*	*0.88*	0.7827	0.78
			Random Forest	***0.9176***	***0.92***	*0.9397*	*0.94*	0.8127	0.81
			Gradient Boosting	*0.9167*	*0.92*	***0.9421***	***0.94***	***0.8614***	***0.86***
	NER		Gradient Boosting	0.890	0.892	0.936	0.94	0.8775	0.88
			Random Forest	0.931	0.930	***0.952***	***0.95***	***0.9149***	***0.91***
			Linear SVM	***0.949***	***0.951***	0.947	0.95	0.8673	0.87
Fine-Grained	Empath Analytics		LDA	0.901	0.901	0.889	0.89	0.8277	0.83
			Random Forest	0.908	0.910	***0.923***	***0.92***	*0.8127*	*0.81*
			Gradient Boosting	***0.928***	***0.931***	0.921	0.92	**0.8726**	**0.87**
	Empath + VADER		LDA	0.9073	0.91	0.888	0.89	0.8127	0.81
			Random Forest	0.9086	0.91	***0.925***	***0.93***	*0.8314*	*0.83*
			Gradient Boosting	***0.9326***	***0.93***	0.921	0.92	**0.8764**	**0.88**
Coarse and Fine-grained fusion	TextSummarization + Empath		LDA	0.884	0.880	0.891	0.89	0.7902	0.79
			Random Forest	0.895	0.900	0.926	0.93	***0.8389***	***0.84***
			Gradient Boosting	***0.896***	***0.900***	***0.926***	***0.93***	0.8352	0.84

Table 4.1: Individual Dataset Results

Finally, when all the datasets are merged and applied on the best algoriths of granularity and its fusion, the following table lists down our conclusions for metric evaluations.

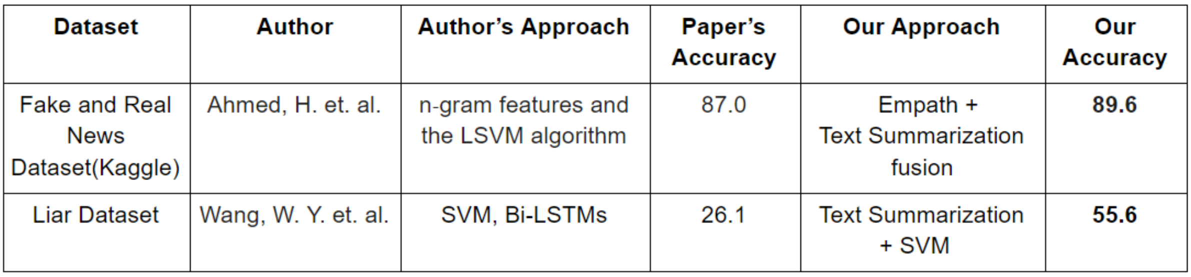


Model Type Algorithm ML-Models			**All Merged**	
			Accuracy	**F1-Score**
Coarse-Grained	Text Summarization Topic = 20	KNN-3	0.8655	0.87
		Random Forest	**0.8842**	**0.88**
		Gradient Boosting	0.8751	0.88
Fine-Grained	Empath Analytics	LDA	0.8340	0.83
		Random Forest	*0.8699*	*0.87*
		Gradient Boosting	**0.8732**	**0.87**
Coarse and Fine-grained fusion	TextSummarization + Empath	LDA	0.8881	0.89
		Random Forest	0.8988	0.9
		Gradient Boosting	***0.9056***	***0.91***

Table 4.2: Merged Dataset Results

4.4 Base Line Comparisons

After comparing our results with standard published papers, we put forth the following table of comparisons. We also list down the efficiency and approach we executed to attain the same. Fake and Real News Dataset (Kaggle) [7] and Liar Dataset [33] are used for the basis of comparison with referenced papers.



Dataset	**Author**	**Author's Approach**	**Paper's Accuracy**	**Our Approach**	**Our Accuracy**
Fake and Real News Dataset(Kaggle)	Ahmed, H. et. al.	n-gram features and the LSVM algorithm	87.0	Empath + Text Summarization fusion	**89.6**
Liar Dataset	Wang, W. Y. et. al.	SVM, Bi-LSTMs	26.1	Text Summarization + SVM	**55.6**

Table 4.3: Baseline Comparisons

4.5 Summary

On the analysis of the textual news on the dataset, emotions,semantics,factual features were classified accordingly. Around 200 fine attributes were considered for fine grain and topics were modelled based on coarse grain algorithms. Finally, both algorithms ware fused to generate another efficient model for testing. To clarify the results, analysis were carried out on 3 datasets and a combined dataset with around 13000 news data.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We conclude granularity concepts and its implementations, ie. Fine Grain and Coarse Grain on textual news. Data Preprocessing was a core part along with selection of Empath library that drives deeper into word level features for analysis of the required news. NER, Text Summarization and LDA topic modelling was an integral part for coarse grain sentence level analysis. Merging fine level analysis and coarse grain analysis to build an exterior model which is efficient enough to predict news classes. Comprehensive experiments are designed and implemented based on three existing standard datasets. Experimental results show that the proposed models achieve better detection performance on the mixed dataset than corresponding baselines.

5.2 Future Work

Analysed sentence and word level features to be combined to predict the fakeness of news appropriately so as to prevent any loss due to false predictions by model, if any.. Finally, building an application that can be used by all. Chrome extension is one of the planned idea for its user friendly nature.

References

[1] N. Cao, S. Ji, D. K. Chiu, M. He, and X. Sun, "A deceptive review detection framework: Combination of coarse and fine-grained features," *Expert Systems with Applications*, vol. 156, p. 113465, 2020.

[2] A. Oehmichen, K. Hua, J. A. D. Lopez, M. Molina-Solana, J. Gomez-Romero, and Y.-K. Guo, "Not all lies are equal. a study into the engineering of political misinformation in the 2016 us presidential election," *IEEE Access*, vol. 7, pp. 126305–126314, 2019.

[3] A. Drif, Z. Ferhat Hamida, and S. Giordano, "Fake news detection method based on text-features," 08 2019.

[4] B. Markines, C. Cattuto, and F. Menczer, "Social spam detection," in *Proceedings of the 5th international workshop on adversarial information retrieval on the web*, pp. 41–48, 2009.

[5] V. Qazvinian, E. Rosengren, D. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 1589–1599, 2011.

[6] A. Gupta, H. Lamba, and P. Kumaraguru, "$1.00 per rt# bostonmarathon# prayforboston: Analyzing fake content on twitter," in *2013 APWG eCrime researchers summit*, pp. 1–12, IEEE, 2013.

[7] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*, pp. 127–138, Springer, 2017.

[8] N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[9] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, "Phi. sh$ ocial: the phishing landscape through short urls," in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, pp. 92–101, 2011.

[10] E. Fast, B. Chen, and M. S. Bernstein, "Empath: Understanding topic signals in large-scale text," in *Proceedings of the 2016 CHI conference on human factors in computing systems*, pp. 4647–4657, 2016.

[11] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001, 2001.

[12] C. M. Fuller, D. P. Biros, and R. L. Wilson, "Decision support for determining veracity via linguistic-based cues," *Decision Support Systems*, vol. 46, no. 3, pp. 695–703, 2009.

[13] P. L. Prasanna and D. R. Rao, "Text classification using artificial neural networks," *International Journal of Engineering & Technology*, vol. 7, no. 1.1, pp. 603–606, 2018.

[14] C. Naik, V. Kothari, and Z. Rana, "Document classification using neural networks based on words," *International Journal of Advanced Research in Computer Science*, vol. 6, no. 2, 2015.

[15] M. Whitehead and L. Yaeger, "Sentiment mining using ensemble classification models," in *Innovations and advances in computer sciences and engineering*, pp. 509–514, Springer, 2010.

[16] R. E. Schapire, “The boosting approach to machine learning: An overview,” *Nonlinear estimation and classification*, pp. 149–171, 2003.

[17] B. Snyder and R. Barzilay, “Multiple aspect ranking using the good grief algorithm,” in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 300–307, 2007.

[18] K. Xu, F. Wang, H. Wang, and B. Yang, “Detecting fake news over online social media via domain reputations and content understanding,” *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 20–27, 2019.

[19] S. Castelo, T. Almeida, A. Elghafari, A. Santos, K. Pham, E. Nakamura, and J. Freire, “A topic-agnostic approach for identifying fake news pages,” in *Companion proceedings of the 2019 World Wide Web conference*, pp. 975–980, 2019.

[20] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, “Text summarization techniques: a brief survey,” *arXiv preprint arXiv:1707.02268*, 2017.

[21] S. Sriram, “An evaluation of text representation techniques for fake news detection using: Tf-idf, word embeddings, sentence embeddings with linear support vector machine.,” 2020.

[22] J.-S. Shim, H.-R. Won, and H. Ahn, “A study on the effect of the document summar ization technique on the fake news detection model,” , vol. 25, no. 3, pp. 201–220, 2019.

[23] J. Li, A. Sun, J. Han, and C. Li, “A survey on deep learning for named entity recognition,” *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[24] A. Ritter, S. Clark, O. Etzioni, *et al.*, “Named entity recognition in tweets: an experimental study,” in *Proceedings of the 2011 conference on empirical methods in natural language processing*, pp. 1524–1534, 2011.

[25] A. Savelieva, B. Au-Yeung, and V. Ramani, “Abstractive summarization of spoken andwritten instructions with bert,” *arXiv preprint arXiv:2008.09676*, 2020.

[26] I. Bíró and J. Szabó, “Latent dirichlet allocation for automatic document categorization,” in *Joint european conference on machine learning and knowledge discovery in databases*, pp. 430–441, Springer, 2009.

[27] S. A. Alasadi and W. S. Bhaya, “Review of data preprocessing techniques in data mining,” *Journal of Engineering and Applied Sciences*, vol. 12, no. 16, pp. 4102–4107, 2017.

[28] W. McKinney *et al.*, “pandas: a foundational python library for data analysis and statistics,” *Python for High Performance and Scientific Computing*, vol. 14, no. 9, pp. 1–9, 2011.

[29] M.-h. Zhu, “Research on data preprocessing in exam analysis system,” in *Communication Systems and Information Technology*, pp. 333–338, Springer, 2011.

[30] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *International conference on machine learning*, pp. 1188–1196, PMLR, 2014.

[31] P. Tijare, “A study on fake news detection using naïve bayes, svm, neural networks and lstm,” 07 2019.

[32] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, “Confusion matrix-based feature selection.,” *MAICS*, vol. 710, pp. 120–127, 2011.

[33] W. Y. Wang, “" liar, liar pants on fire": A new benchmark dataset for fake news detection,” *arXiv preprint arXiv:1705.00648*, 2017.

[34] H. Ahmed, I. Traore, and S. Saad, “Detecting opinion spams and fake news using text classification,” *Security and Privacy*, vol. 1, no. 1, p. e9, 2018.

Acknowledgements

At the outset, we thank the God Almighty for the grace, strength and hope to make our endeavor till now a success. We express our sincere gratitude to our guides Dr. Dipti P Rana, Assistant Professor, Computer Engineering Department, SVNIT and Ms. Isha Agarwal, PhD Scholar, Computer Engineering Department, SVNIT for their exemplary guidance, monitoring and constant encouragement without which this report would not have been possible.

We would like to thank Dr. Mukesh A. Zaveri, Head of Dept. Computer Engineering, SVNIT for support provided.

We would like to thank all who directly or indirectly contributed to this report.