**Real Time Fake News Detection using Objectivity Extraction and Analytics Approaches**

by

Chaowei Zhang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 7, 2021

Keywords: Fake news Detection, Real time data processing, Parallel computing, Memory optimization, Objectivity and subjectivity separation



Approved by

Ashish Gupta, Associate Professor of Department of Systems and Technology
Xiao Qin, Professor of Computer Science and Software Engineering
Kai Chang, Professor of Computer Science and Software Engineering
Nguyen Ahn, Assistant Professor of Computer Science and Software Engineering

Abstract

Fake news is playing an increasingly dominant role in spreading misinformation by influencing people's perceptions or knowledge to distort their awareness and decision-making. The growth of social media and online forums has spurred the spread of fake news causing it to easily blend with truthful information. This study provides a novel text analytics-driven approach to fake news detection for reducing the risks posed by fake news consumption. In this dissertation, we focus on addressing fake news detection tasks by establishing three analytics models.

In the first part, we first describe the framework for the proposed approach and the underlying analytical model including the implementation details and validation based on a corpus of news data. We collect legitimate and fake news, which is transformed from a document based corpus into a topic and event-based representation. Fake news detection is performed using a two-layered approach, which is comprised of detecting fake topics and fake events. The efficacy of the proposed approach is demonstrated through the implementation and validation of a novel FakE News Detection (FEND) system. The proposed approach achieves 92.49% classification accuracy and 94.16% recall based on the specified threshold value of 0.6.

We propose a computational approach in the second part for detecting fake news in real time. The proposed methodology utilizes event and topic extraction techniques along with a topic- merging mechanism to process real time news data and reduce the number of topics. This approach includes a two-stage procedure for improved memory management using a streaming framework. We report the findings from several computational experiments for benchmarking proposed methodology in different system settings. Our approach is more time- efficient in detecting fake news while also leading to a 19.76% reduction in the number

of topics and 26.92% reduction in the numbers of data clusters when compared to other Fake news detection systems.

Objective and Subjective separation(OSS) in text could benefit textual affective analysis fundamentally. Existing OSS approaches such as extracting perceptual pieces mainly concentrate on identifying subjectivity. Objectivity learning in language has been becoming a challenging task due to false knowledge and other misinformation news propagating over the internet. Finally, this dissertation presents a novel objectivity-subjectivity separation approach for short texts without using traditional subjective clues, referred to as '*private states*.' We accomplish this by leveraging three latent features (view point of subject and object, and tense) of extracted relational triple sets in sentences. In the model, we propose a group of algorithms to extract latent features and recognize subjective or objective patterns from datasets. We assess our approach via regrouping the three latent features as three two-elemental variables and a triple variable for comparing the distributions of these variables between objective and subjective datasets. The results indicate that model based on our proposed methodology has approx. 87.5% accuracy, and approx. 97% recall on evaluating extracted objective patterns.

Acknowledgments

This dissertation would not have been completed without invaluable guidance, experience sharing, constant support and encouragement from my advisors, people in our research group and family members during my study at Auburn University.

It is a great pleasure to thank those who made this dissertation possible.

First and foremost, I would like to give my most sincere and deepest gratitude to my advisors, Dr.Gupta and Dr.Xiao Qin, for their great efforts, trust and patience in my work. I will never forget their extensive knowledge in the field of information systems and inexhaustible enthusiasm for research, which keeps inspiring and driving me to accomplish my research. When working on the book chapters "FEND", "RT-FEFND", and "OSS". Their insightful advices and suggestions helped and enlightened me in setting up accurate motivations behind the research, building a EDOM and GreenDB clusters that can reduce energy consumption in database systems.

I am also tremendously grateful to be advised by my committee members Dr.Kai H. Chang, and Dr.Nguyen Ahn who reviewed my proposal and dissertation documents. They gave me a number of valuable suggestions, by which my dissertation had been substantially improved. I would like to show my appreciation for Dr.Amit Mitra as my university reader.

Working with our research group is fantastic. I owe my gratitude to Tathagata Bhattacharya, Xiaopu Peng, Jianzhou Mao, Ting Cao, Christian Kauten and Yi Zhou who helped me with paper writing, experimental result collection and group discussions. In addition, all the professors and students in the Department of Computer Science and Software Engineering are greatly appreciated, because an excellent atmosphere for study and research is created and maintained by everyone.

Finally and most importantly, the endless love from my family is the most powerful strength that keeps me fighting for my research. My parents, grandparents, and all of the other family members always stay with me, cheering for achievement and overcoming all difficulties. I really appreciate all your supports and encouragement while I am working on my Ph.D these five years.

Table of Contents

List of Figures

List of Tables

# Chapter 1

# Introduction

In the current era of internet and information, fake news often propagates over the social media through mimicking legitimate news media contents or fabricating false information [63]. As the novel corona virus spreads across the world in 2020, we are also seeing a simultaneous growth in various disinformation and misinformation related to COVID-19. Although the continuous legitimate reports about the increasing number of infected people invokes panic, the spreading of fake news only exacerbates the current situation by developing misunderstanding about the disease, its properties, causes and potential cure. For example, a widely spreading rumor on Weibo, the largest social media platform in China, falsely claims that ShuanghuangLian, a Chinese medicine used for heat-clearing and detoxifying, can cure COVID-19 [52]. A large number of people believed in this fake news and rushed to buy this medicine from pharmacy leading to increased risk of virus spread [128]. Fake news not only causes severe societal and trust issues but also negatively impacts individual's health and well-being. Fake news has become a global problem that needs to be dealt with in a timely manner to curb its spread and thereby making the detection efforts critically important for the overall well-being of individual, organization, society and economics.

Fake news can be defined "as the online publication of intentionally or knowingly false statements of fact [61]." In essence, the focus is on articles or messages posted online with the anticipation of the message going "viral". Fake news thrives on the false rumors, hoaxes, sensationalism, and scandal resulting from the dissemination of news articles through social media [42]. While intentional harm is debated, various incentives, - such as monetary, social, and political benefits - often drive the fake news spread.

## 1.1 Motivation of FEND Model

Recent proliferation in the use of social media as a vehicle for spreading fake news has significantly raised the risks imposed on individuals as well as organizations by the spread of misinformation (false information). For example, social platforms are frequently used to spread fake news via modifying authentic news or making fabricated news. Very recently, Berners-Lee, the inventor of the World Wide Web, claimed that fake news has been one of the most disturbing Internet trends that have to be resolved [118]. It is challenging, if not futile, to detect deceptive news due to the diversity and disguise of deceptions. Fake news may cause adverse influence coupled with damages. It influences an individual's decision-making and distorts one's perceptions about the real events by altering the information feeds that are utilized for news consumption. At the organizational level, the impact is more adverse as it poses risk to their brand names and can potentially affect on the consumption of their product or services [49]. News articles shared using social media further exacerbate this problem due to increased online media consumption and use of bots (e.g., twitter bots) that automate the spread of false information. A recent survey indicates that, of the known false news stories that appeared in the three months before the 2016 election, those favoring either one of the presidential candidates were shared approximately 38 million times on Facebook [4].

Contemporary developments in methods of news verification address the growing demand for automated means of discriminating real news from fake news among the immense volume of data [102]. In general, existing fake news detection approaches are categorized into two groups based on the underlying approaches, namely, linguistic, or network techniques. Linguistic approaches (e.g., natural language processing or NLP) are focused on news content, and aim to investigate fake news patterns by analyzing underlying semantics. In contrast, network approaches leverage existing knowledge networks to check facts of news (e.g., [58]). Recent operations research studies have started to utilize the capabilities of such text analytics and modeling approaches in various application domains. For example, text

analytics approaches have been used for suggesting improved design features for augmented reality health apps([64]). A few studies have applied such techniques in finance. For example, [122] predicts financial risks by using a finance-specific sentiment lexicon and regression and ranking techniques to explore the relations between sentiment words and financial risks based on a bag-of-words model. A few studies have applied text analytics approaches for detecting product defects([1]), sales forecasting [62], etc. Use of these techniques in operations research is increasing as researchers start to investigate the value of unstructured data for knowledge discovery with different industry sectors.

A growing number of techniques have been devised to verify news credibility([30] [59]). Existing fake news detection methods aim to detect intentionally deceptive news. Unfortunately, these approaches are inadequate to automatically and accurately pinpoint fake news from a massive amount of new data that is continuously generated by social media and web services. To address this gap, we propose a novel two-phase approach to detecting fake news. In phase one, we extract events from legitimate news, which are then categorized into an array of topic clusters. Each cluster is centered around a news topic. In phase two, a news item to be verified is classified into a topic cluster, where we validate the events reported in the news by comparing to those in the topic cluster. This approach is inspired by fake news detection demands([104]) and is reliant on text clustering and classification approaches ([65] [125]), as well as lexical databases([75] [126]).

Recently developed fact checking tools are adept at comparing news against a collection of knowledge represented as a network. However, such comparisons are very time consuming due to the volume and constant growth of the knowledge base. To speed up the detection performance, we propose to partition a large number of genuine and authenticate news (a.k.a., factual statements) into clusters, each of which is comprised of news sharing similar topics. To judge the credibility of a news, we classify the news into a topic cluster in which events are compared against those of the news. In case the news does not fall into any existing cluster, we mark the news as a deceptive one.

We demonstrate the implementation of the proposed approach for detecting fake news by carrying out two distinctive phases to discover deceptive news. First, trustworthy news are categorized into clusters according to topics. Each cluster is centered around common news topics. Second, we detect fake news by verifying events extracted from the news in a specific cluster.

The approach proposed in this study treats news as a fake one if (1) it is a news outlier (i.e., not classified in any topic cluster) or (2) the similarity between the news events and those of the cluster is below a specified threshold. A large number of authenticated news articles classified into news clusters based on topics and stored in a news database that periodically receives news updates by accumulating latest news stories from legitimate news sources such as CNN and Fox News that have been verified as legitimate by the research community. If an incoming news to be detected cannot be classified into any existing news cluster, it is marked as a candidate fake news. Otherwise, the incoming news is placed into the corresponding cluster for further analysis. The credibility of the incoming news is measured by comparing the events extracted from the news with those in the news cluster. When the news article's credibility is below a specified threshold, the news are classified as fake.

This study makes several contributions. First, a novel analytics-based approach for fake news detection that applies topic based classifying mechanism to group legitimate news into multiple topic clusters is presented. News in each cluster share common topics. An event-extraction mechanism is designed for extracting events from these news articles. Second, we propose and implement a credibility measure for evaluating the authenticity of any news by comparing events extracted from the news to those of the legitimate news. Third, based on the proposed approach, we present a framework for the development and validation of a novel system, FEND, to detect fake news by leveraging a large legitimate news database

that we built. Finally, we illustrate how to evaluate the performance of FEND using a real-world news dataset. The experimental results indicate that FEND achieves a high fake news detection accuracy.

### 1.2 Motivation of RT-FEND system

Fake news spread on the social media is usually created for misleading guidance of public opinion in order to achieve authors' financial or political goals and support their own beliefs. With the increase in online interactions and popularity of social media (e.g., self-media and webcast), the fake-news propagation becomes easier than in the age of traditional news media like newspapers or TV. This trend is fuelled by the low cost of maintaining social media and ease of use of social media [109]. A large number of websites don't authenticate users' personal information, implying that users are not held accountable for posting incorrect or biased information. It is prudent to detect fake news in a timely manner. News articles are time sensitive and created by agents or organizations (e.g., newspaper offices, TV stations, and other media). Prior to releasing news, journalists (e.g. reporters and news editors) complete a list of procedures, namely, news collection (e.g. personage interviews, incident tracking, purchase of original news materials), news editing, and news verification, etc. Time sensitivity is an extremely critical property of news for two reasons. First, a news organization maximizes benefits by taking the lead in reporting news. Second, the nature of people's curiosity and the pursuit of novelties is the fundamental reason to determine the value embodiment of time-sensitivity of news. A large volume of news items are generated every day and propagated at a fast speed in the era of social media. For example, more than 450,000 tweets are shared on Twitter and 46,740 photos are posted on Instagram in every minute [70]. Everyone who is engaged in social-media networks focuses on consuming updated news, which amplifies the need for real time detection of fake news. News may become quickly outdated, meaning that news no longer carries novelty factor or audience is no longer interested in the aging news.

Recent research on fake news has focused on developing new detection approaches. For example, BS-Detector is a browser plug-in for detecting fake news; PolitiFact is another fact-checking website that gives the credibility of claims by U.S officials [44]. However, these efforts remain rudimentary or fall short of handling the complex problem of detecting fake news in real time. Another fake news detection system embraces an offline data collection mechanism, which is inadequate for processing news in a real-time manner [136]. Real time detection of fake news is critical for curbing its spread through social media or various blogs and curtail its consumption by individuals. In this study, we focus on the conceptual and actual development of a real-time fake news collection and detection mechanism, which is capable of pruning outdated news.

One challenge in detecting fake news in a real-time manner is to deal with an excessive number of topics extracted from news articles. On average, approximately 23.1 topics are extracted from one news article. Topics tend to be diverse in different news. As a consequence, processing a large volume of news data is time consuming. To ensure the originality and integrity of news data, we stay away from the traditional dimension-reduction methods (e.g., feature pruning and principal component analysis) since topics in up-to-date news are likely to be unique.

In this study, we design a real-time fake-news detection system, which is capable of processing massive amount of news data by embracing salient features of topic reduction, event extraction, real-time processing, and parallel computing. The proposed system seamlessly integrates an array of processing modules to facilitate real-time data collection, news analysis, and fake-news detection. The experimental results indicate that the proposed system achieves a high fake news detection accuracy and high efficiency in comparison with other fake news detection baselines.

The main contributions of this study are summarized as follows. First, we apply lexical repositories and a topic-comparison method to reduce the dimensionality of training datasets. Second, we employ the real-time analytics framework to implement streaming data collection

and clustering to construct knowledge bases. Third, we design a novel two-stage algorithm-based procedure to efficiently manage the batch size of streaming data. Fourth, we develop a real-time fake-news detection model that includes a classifying mechanism based on topic and event-based filters. Fifth, we deploy our system on a up to 9 nodes computing cluster using virtual machine. Finally, we evaluate the performance of the system processing a real-world news dataset using a series of computation experiments and benchmark its performance against other systems.

Next section introduces and provides a comparative description of the prevalent approaches used for fake news detection. First, we summarize the current status of research on fake news detection and classify into different categories based on a simple benchmark. We then analyse and discuss their relative merit and weaknesses.

### 1.3 Motivation of OSS

Objectivity and subjectivity in language are latent cognitional features that are difficult to be recognized. Objective and subjective separation (OSS) playing an important role and is an essential step in the many natural language processing-based applications. Traditional OSS methods try to identify subjective pieces, or sentimental words and phrases within the data primarily for sentiment analysis. A statement can be classified as subjective based on its "*private states*" [127], which expresses opinions, rants, allegations, accusations, suspicions, and speculations [98]. In contrast, objective statements contain facts to describe an event or its nature that can not altered product [113]. There are various types of documents or publications such as history texts, scientific journals, news articles, encyclopedias, etc. [67] that massively utilize objective statements. Our knowledge accumulation predominantly relies on using such objective documents as these contribute towards improving our understanding of various social, technical, and societal aspects. Many NLP-based applications leverage objective documents, such as facts extraction [7], fact checking [95], knowledge bases construction [16], question answering(QA) [28], and deception detection [71] in transcripts.

Most existing OSS techniques are also referred to as subjective learning [98] [127], subjectivity detection [24], or subjectivity classification [99] [27]. These methods mainly aim to extract subjective clues from the text to perform automatic subjective analysis for different purposes. Subjective clues are either annotated manually or collected from subjective corpus or vocabularies. For example, Riloff, and Wiebe [98] developed a subjective corpus, called as *MPQA*, which has over 8,000 subjective clues that are either derived from an unlabeled corpora and subsequently annotated by human experts, or are automatically extracted from labeled subjective datasets. However, this approach has limitations due to inherent complexity of human language and embedded sentiments. For example, a statement comprising of subjective clues could sometimes be mistakenly classified as an objective statement or vice versa [66]. In contrast, subjective statements may not include any subjective clue [127]. The fundamental reason for this mis-classification is due to the fact that natural language and cognitive thinking have a many-to-many relationship. Explicitly, a language expression may deliver different message to audiences or readers as single idea could be expressed in multiple ways.

A majority of subjectivity detection approaches are precursor to sentiment analysis and are done to extract subjective statements [24] [21]. These sentiment analysis approaches subsequently perform emotion detection [92], positive & negative opinion extraction [13], morality classification [89], and polarity classification [60]. In contrast, objective detection approaches aim to improve the efficiency of tasks such as fact checking, deception detection etc. by cleaning the subjective statements. All objective pieces can be segregated from subjective statements a-priory to sentiment analysis techniques to improve the overall authenticity, reliability, and performance of the process.

In this work, we proposed a novel method for separating subjectivity from objectivity in text documents using OpenIE technique. The proposed method utilizes three parts of sentences (i.e., subject, predicate, and object) to investigate the influence of (1) view points of subjects' agents, (2) the tense of sentences, and (3) view points of objects' agents. All

combinations of these three feature are then evaluated to discriminate objective and subjective patterns with the sentences. This study makes significant contributions from the following five perspectives. First, view points detection algorithm is developed to detect view points of subjects and objects. Second, we designed a tense detection algorithm for classifying sentences into twelve tense categories. Third, the frequencies of each pair of three features are also applied to distinguish objective and subjective patterns. Forth, we design an algorithm to evaluate the performance of extracted patterns by leveraging view points detection algorithm's outputs, tense detection algorithm's outputs, and a given list of threshold. Finally, we provide validation of the proposed approach by evaluating the performance of the extracted patterns using cross validation.

Chapter 2

Literature Reviews

## 2.1 Fake News Detection

### 2.1.1 Fake News Risks

The initiation and spread of fake news presents significant risks from many different perspectives, including from a national security standpoint. A good example of this is deliberately misleading news that attempt to influence an individual's perception about another individual or election results. In politically divided environments, such as those being witnessed in the US and Europe, people tend to gravitate towards news from sources that are congenial to their belief or political taste. This may be attributed to confirmation bias or "tunnel vision" which involves one-sided case building based on preconceived notions or ideologies[80]. [90] report on three studies aimed at testing the propensity to think analytically and susceptibility to fake news. These studies find that, contrary to the confirmation bias theory, people are deceived by fake news as they fail to think analytically while consuming media, not because they think in a motivated manner. [112] discusses various cognitive biases that act as barriers in evaluating and correcting misinformation when humans process fake news, i.e., misinformation. The spread of fake news presents the risk of duping readers that takes disadvantage of the readers' preference for congenial news and the lack of analytical thinking while consuming news media.

The preference for agreeable news bits is further exacerbated with the "echo chamber" or the "filter bubble" phenomenon occurring with social media. On social media platforms, people tend to selectively associate with individuals of similar viewpoints and consume information appealing to their perspectives. The personalization features of social media amplifies

the effect[88]. Fake news functions as a catalyst to further intensify readers' point of views and runs the risk of information polarization. [50] demonstrate the information polarization effect due to differential consumption of fake news occurring through selective exposure to misinformation.

Often, instances of fake news are subsequently followed by fact-checks published on different media outlets. However, as shown by a study conducted by [108], partisan news consumers selectively evaluate and share fact-checking articles, again due to the "echo chamber" effect. Studies on political behavior have shown different results with respect to fact-correction phenomenon. [81] found that a "backfire effect" occurs when humans are presented with fact-checks of misinformation, in that they psychologically counter-argue and strengthen their initial false perceptions. However, a recent study by [129] has shown no evidence of factual backfiring. From a risk analysis perspective, although fact-checks may be effective in correcting the news for the record, they are practically ineffective in mitigating the risk of false information consumption and information polarization that occurs in the first place. This emphasizes a clear need for more objective fake news detection mechanisms that can serve to prevent the consumption of false or misinformation.

Clearly, fake news presents a keen risk of damaging the foundations of journalism ideals of veracity, objectivity and accountability. Fake news publishers risk accusations of crimes and violations of governmental regulations. [61] present a detailed survey of many legal and regulatory issues that fake news publishers may face. These may range from civil legal claims concerned with defamation, intellectual property law, or intentional infliction of emotional distress (IIED) to government violations and crimes such as cyber bullying. Also, such publishers may be in violation of social media platform account policies and search advertising restrictions. Savvy publishers act to proactively minimize the legal exposure and risks through mechanisms such as disclaimers and notices, website terms and conditions, and media liability insurance policies. In response, social media platforms have become cautious and have started incorporating detection mechanisms for fake news. However, cross-platform

mechanisms have received limited attention. Detecting fake news by originating from sources across multiple websites and platforms can serve as a useful tool for regulators.

To mitigate these risks, various detection approaches are discussed in the following section.

### 2.1.2 Fake News Detection

Fake news are created by fabricating nonexistent news or modifying legitimate news. The credibility of fake news are boosted by (1) imitating well-known authors' writing styles or (2) expressing opinions with a tone frequently used in real news. Very recently, an increasing number of fake news detection methods have been developed. All existing detection schemes may be grouped into two distinct classes, namely, linguistic-based methods and network-based methods[30]. Network-based approaches for fake news detection apply network properties as a supporting component for various linguistic-based approaches. Commonly used network properties include, but not limited to, website information, authors/subscribers information, time stamps, and the like. For example, [123] performs user behavior analysis to reduce the misinformation in online social networking forum related to Parkinson's disease. This study reports that misinformation embedded within the discussion thread depends on its content and users characteristics of the author. Another study proposes a model that focuses on investigating the quality of responses in an online crowd-sourced health, clarity of the thread questions, and the users' potential for making useful contributions[124]. The existing sentiment and syntax analysis schemes are customized for special data types, thereby being inadequate for fake news detection systems.

A rumor detection model or CNT proposed by Qazvinian et al. adopts a variety of features such as content-based features (e.g., words and segments appearance, part of speech), network-based features (i.e., re-tweets or tweets propagation) and twitter-specific Memes (i.e., Hashtag or shared URLs). CNT orchestrates an array of strategies to select features to detect misinformation in microblogs[94]. Rubin et al. devised an SVM-based algorithm,

AHGNA, that embraces five predictive features (i.e., Absurdity, Humor, Grammar, Negative Affect, and Punctuation)[104]. After an assortment of feature combinations were evaluated using a total of 360 news articles, Rubin et al. illustrated that the best combination can detect satirical news with a 90% precision and 84% recall. Common fake news features discovered by these approaches may govern unethical writers to write fake news without exhibiting the de- tectable features. To address this weakness, [56] advocate a way of exploiting semantic knowledge from short texts. [56] scheme incorporates text segmentation, part-of-speech tagging, concept labeling, as well as a vocabulary database to harvest a collection of attributes, concepts and instances from a well-known knowledge base. This knowledge-intensive approach offers insights on short texts such a twitter. While a majority of fake news originates from websites, social media facilitates their spread. Larger text originating from news outlets and opinion threads offer deeper insights into the topic and provide richer contexts. Open information extraction (OIE) is a task of extracting factual information from textual data such as Twitter posts, spams, and articles in social media. More recently, OIE tools have been used for producing grammatical clauses, which can be used for topic extraction purposes. For example, [47] demonstrated their approach by extracting topics from a collection of 50,000 microblogs during a disaster event.

Similar to the above semantic knowledge based approach, our proposed approach aims to grasp an understanding of news through complete comparisons of news content. This approach no longer relies solely on statistical, sentiment, or syntax analysis to detect fake news but uses topic and event level analysis to understand patterns that are deeply embedded within the news for improved detection accuracy. Also, traditional fake news detection approaches pay more attention to reducing content leakage, which may provide misleading information when original articles are imitated or modified. As such, our proposed approach takes full advantage of in-depth semantic analysis of the sentences by incorporating OIE coupled with the other techniques to extract knowledge from news articles.

**Sentiment and Syntax Analysis**

Linguistic-based methods such as statistical analysis [33] [132], sentiment analysis [114], linguistic cues analysis [111] and deep syntax analysis [29] [74] have been deployed to detect abnormal information within the text data with high accuracy.

*Statistical analysis* [51] proposed an approach to examining the features of crime narratives. Their results show that psychopaths' speech contain a high frequency of disfluencies; psychopaths often use past tense and less present tense verbs in narratives. [105] proposed the rhetorical structure theory or RST to identify the discrepancy between real and fake textual data by applying the Vector Space Model (i.e., VSM) to assess the confidence level for each datum.

*Sentiment analysis* is a widely adopted strategy for detecting general deception, particularly deceptive Spams. [42] proposed PU-learning to detect deceptive spams by analyzing positive and negative opinions. PU-learning is a semi-supervised technique for building a binary classifier on the basis of positive (i.e., deceptive opinions) and unlabeled examples.

*Linguistic cues analysis* [111] demonstrate that linguistic cues derived from deception theories, in conjunction with content cues based on message content, can be quite effective in distinguishing between fraudulent and non-fraudulent projects on crowd funding platforms. Deception detection is shown to be particularly effective when both static communication (e.g., project description) is analyzed along with dynamic communication (e.g., forum messages). This study uses a similar approach of analyzing entire communication content, i.e., news, but differs from [111] study by focusing on the content similarity.

*Deep syntax analysis* Probability context free grammars or PCFG is a practical method that applies deep syntax analysis to separate sentences into rewrite trees representing syntax structures. For example, [41] investigated syntactic stylometry for deception detection, where features are derived from context free grammar (i.e., CFG) parse trees on hotel review data. Unfortunately, these existing detection schemes are tailored for special data types or specific contexts such as spam reviews detection [25] and spam mail detection (e.g., [57]); therefore,

are inadequate for a general-purpose fake news detection that could apply to wide ranging topics or issues.

#### Topic Extraction

TextRunner is one of the early, but highly scalable, OIE systems proposed by [12]. Since the inception of TextRunner, a few popular OIE approaches have been developed: ReVerb [37], OLLIE [38], and Stanford OpenIE [6]. In 2013, [32] proposed another OIE approach (ClausIE) that maintains information integrity of original textual data by decomposing sentences into a list of 'clauses'. Another study by [14] embraces similar extraction phases, but supplements the capabilities of the approach proposed in [32] by implementing contextual sentence decomposition to facilitate semantic searching. [131] presented a way to extract text relationships without verb expressions. Extraction results were validated by OLLIE and ClausIE. Their findings confirm that more extra relations are discovered by ClausIE than OLLIE, meaning that ClausIE has better performance than OLLIE.

### 2.1.3 Fake News Detection Applications

Recent efforts on fake news detection have focused on varied approaches. For example:

*B.S. Detector* - alerts users of unreliable news sources [97] by searching all links of a given webpage for sources that have been collected in a unreliable-news database, which includes samples of fake news, satire, extreme bias, conspiracy theory, rumor mill, state news, junk science, and the like. Although the database manages vital and rich information to facilitate fake-news detection, this approach only utilizes a knowledge base of untrustworthy links. Unlike the browser extension, our approach takes the news content and performs an ingrained analysis to quantify credibility scores.

*PolitiFact* - is a six-dimensional rating system developed to check facts. It is frequently used to rate the accuracy and credibility of claims made by US officials and others [101]. The PolitiFact system largely depends on human intervention, during which, journalists

assess information via watching TV, scanning social media, and evaluating reader comments. In contrast to PolitiFact, our system applies artificial intelligence models that utilize text analysis of news sources rather than interventions offered by a body of journalists.

*Fake News Detector AI* - identifies fake-news websites by measuring similarity to existing fake-news websites using artificial intelligence techniques as a blackbox [34]. This system uses a neural network–based feature analysis (e.g., headline, code structures, site popularity) approach on known websites, thereby yielding the credibility of the tested websites. Our system differs from this detection tool in the types of features. More specifically, Fake News Detector AI relies on network-based features, whereas our system employs semantic-based features.

### 2.2 Real Time Fake News Detection

Fake news intentionally spreads false information that could mislead the readers. However, fake news could also have high credibility that causes the readers to believe in them. Feature extraction has been the most popular method for fake news detection [87]. For example, [119] performed feature extraction from hoax and non-hoax posts based on a set of documents and users using logistic regression and Boolean crowdsourcing algorithms to achieve an accuracy of over 99%. Also, in [109], the author surveyed and summarized a variety of existing features based on content-specific features (i.e., author, publisher, headline, body, etc.), social context-specific features that include user-based cues (i.e., user profiles, characteristics, etc.), post-based cues (i.e., skeptical opinions, sensational reactions, etc.), and network-based cues (i.e., followers, relationship networks, etc.).

Linguistic-based methods are prominently used to understand fake news by capturing the deception from articles' content. We categorize different fake news detection approaches into three groups [29][9]. First Classification is based on statistical cues. This is the easiest way to obtain the feature from articles and is accomplished by counting the number of different type of elements, such as one or multi word frequency (Unigrams and Ngrams) that often

involves extracting the elements from bag of words representations and using TF- IDF(Term Frequency-Inverse Document Frequency) [104] [83] [91]. This is useful to distinguish fake from legitimate news, or extracting more complicated features such as readability [30] using models such as the Automatic Readability Index (ARI) [3], Flesch- Kincaid [110], Gunning Fog[8] and others. Second classification is based on sentiment-oriented cues [43] [100] [45]. With the subsequent improvement in the underlying dictionary of Linguistic Inquiry and Word Count (LIWC), it became a prevalent tool to support many NLP-based researches, especially for the deception detection [54]. Third classification is based on syntax-oriented cues. In many cases, features based on syntax are extracted using CFG (Context-free grammar), which is used for discovering lexicalized production rules. PCFG (Probability Context Free Grammars) is another advanced syntax analysis technique that divides sentences as a parse tree which is a set of rewrite rules for describing the syntax structure of sentences [137]. There are other syntax analysis applications such as the Stanford CoreNLP that could be used for deception discovery and includes syntax parser relevant functions [69].

In many cases, network properties and behaviors may serve as critical indicators of deception. Network analytic method usually focuses on people's social network properties (e.g., credit rating, number of followers and activity records), the source of articles (i.e. fake news websites) and public knowledge base (i.e. Wikipedia Knowledge Graph). Using social network behavior cues, [31] verified that information propagation through Twitter could be used to establish a veracity evaluation mechanism for the 2013 Australian election. Especially, the volume of retweets, hyperlinks in tweets, difference between original tweets and retweets were helpful in detecting phony online personas, fake bots and deceptive Twitter strategies. Other approaches that utilize network-based fact checking depend on existing knowledge base and publicly available structured data such as DBpedia ontology [18]. Google Relation Extraction Corpus (GREC) has been widely used for deception detection [55].

Recently, a few approaches have been developed and implemented for detecting fake news. For example, B.S. Detector [97], an online browser extension, alerts users to unreliable news sources by searching all links of a given webpage against a validated database of unreliable news. This will then be classified into one of the many categories: fake news, satire, extreme bias, conspiracy theory, rumor mill, state news, junk science, hate group, clickbait, proceed with caution, etc. In addition to checking against a static list of URLs in a pre-complied database, inability to perform semantic analysis is a major limitation of this approach. PolitiFact is another fact-checking website that rates the accuracy and credibility of claims using 'Truth-O-Meter', which is an instrument to assess a news on a scale of six [101]. A major limitation of this system is that it is only restricted to politics and requires human intervention and input. Fake News Detector AI is yet another neural network driven system that could identify fake news websites based on their similarity with existing fake news [97]. However, a major limitation of this system is lack of explanations for the result as this only generates a warning message if the given URL is unreliable.

In spite of effective usage of above methods, an obvious drawback these techniques are their instability. For example, document features such as number of paragraphs, sentiment-based word count, and syntax structure of sentences can be easily simulated. Additionally, these deception detection methods can only work for specific data types, such as spam reviews detection [25]. A recent study has proposed a model that abandoned traditional linguistic and Network- based methods while focusing on fact extraction from datasets [136] using open information extraction techniques and refine triples as formal features i.e., events comprising of subject (u), predicate (v) and object (o) or e = u, v, o. Topics are represented as t = u, o and, therefore, e = t, v. This method effectively avoids using statistical methods by using topics as features for clustering and focuses on the meaning extraction from text, which allows the models to understand the semantic differences among articles through vectorization of features. However, their proposed approach suffers from efficiency, extensibility topic redundancy, bad runtime environment; outdated dataset [136]. This study overcomes several

of the drawback stated above by utilizing real time streaming approaches for fake news detection and benefit from a distributed computing environment. Such approaches help keep the knowledge base updated for fact checking with more recent news having higher impact factor for successional news.

## 2.3 Objective and Subjective Separation

Objective and subjective parts in linguistic research are commonly referred to as *facts* and *opinions*, respectively [23]. A handful of studies have recently focused on attempting to extract opinions from articles using cutting-edge techniques such as sentiment polarity classification [77], opinion mining [22], and subjectivity detection [106]). In subsequent subsections, we perform extent review of literature on existing objective and subjective learning methodologies with a focus on data collection, data annotating, feature extraction, and objective/subjective learning.

### 2.3.1 Objective and Subjective Data Collections

Rapid growth in internet usage and aggressive consumption of social media platforms is leading to easy access to a vast volumes of information from various sources such as news reports, advertisement, blogs, etc. and increased information sharing through tweets, crowd-sourcing platforms, etc. Consequently, massive amount of textual data is becoming available for different NLP-related analysis. All textual datasets that may be used for objectivity and subjectivity analysis could be broadly classified into three categories: (1) public-oriented textual data originating from journalism sources or blogs (e.g., document-level news or debates); (2) personal-oriented relatively short texts from various social media (e.g., Instagram, Twitter, and Facebook); (3) commercial-oriented user reviews (e.g., movie or product reviews).

*Document-level datasets* consist of a list of textual data stored in different documents, where each document typically contains (1) relatively complete context of a story or an event;

(2) subjective perceptions of writers, subjects and objects, perhaps in the forms of quoted speech; (3) objective descriptions, which are unbiased statements. In an earlier study, [86] applies a '*subjectivity detector*' to filter subjective sentences from the original document-level dataset to boost document-level polarity classification. In this study, authors leverage coherence and proximity relationships among sentences to analyze the subjective similarity among text spans within discourse boundaries. Another study on subjectivity detection at the document-level employs a two-layered document-level sentiment classification approach to (1) extract subjective sentences; (2) detect the sentiment of the documents based on extracted subjective statements [134]. In summary, document-level datasets contain both objective and subjective pieces and require pre-processing and data cleaning tasks.

*Sentence-level datasets* can be generated from document-level textual data or personal-oriented short texts from social media. A few studies have annotated sentences as positive, negative, or neutral opinions e.g., [66]. The twitter datasets are frequently used for sentence-level analysis. For example, a study utilizes multinomial Naive Bayes classifier to determine positive, negative and neutral sentiments of tweets [84]. In this study, authors use *Tree-Tagger* to investigate the impact of using POS tags (e.g., n-grams) on sentiment evaluation performance. Another study explored using a different research design that uses a tree representation of tweets called *tree kernel* to avoid the need for performing extensive feature engineering on twitter datasets [2]. The study performed sentiment analysis using three different features namely, frequencies-based features, POS&polarity score-based features, and boolean-value-based features.

*Aspect/Entity-level datasets* are predominantly used for discovering the entities or aspects that an individual likes or dislikes [66]. Several studies have used review large textual dataset for subjectivity analysis. For example, movie review datasets [1] released by Cornell university that has been widely used for sentiment analysis [85]. [46] illustrates how subjectivity, informativeness, readability, and linguistic correctness of customers' reviews influence

[1]Cornell Movie review datasets, `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

product sales and customers' perceptions. This study applies Random Forest-based classifier to inspect the relative importance of various reviewer features, review subjectivity features, and review readability features to estimate the impact of different reviews on economic outcomes. Similarly, Zhao, et. al. [138] make use of 127,629 online hotels reviews[2] of customers to estimate customers' satisfaction. This work is based on using review attributes to discover how subjectivity, readability, and length of reviews can negatively affect customer ratings. On the other hand, customers' rating were found to be positively influenced by diversity and sentiment polarity within the data.

### 2.3.2 Subjectivity Annotations

Objectivity and subjectivity, being the latent features of text, are difficult to classify automatically due to complexity and diversity in natural language. Objectivity and subjectivity annotation tasks are initially performed manually by annotators. *Inter rater reliability* [72] is a commonly used measure to eliminate ambiguity across annotation outcomes derived from multiple annotators. Riloff and Wiebe [98] developed a rule-based subjectivity classifier by using known subjective vocabulary to collect subjective patterns from unannotated datasets. Such extracted subjective patterns can then be applied to identify subjective statements from other unannotated datasets. There are other subjectivity annotation schemes that have been developed at the discourse level [66]. For example, Asher, et. al. [9] applied five rhetorical features: contrast, correction, support, result, and continuation to build a sentiment annotation corpus that could be used for identifying opinion polarity. Another subsequent study proposed a three-step semantic analysis approach that includes: (1) categorizing text documents based on various opinion expressions such as judgment, advise, and sentiment expressions; (2) evaluating discourse segments using a shallow representation; (3) comparing and analyzing the distribution of categories in different types of multi-lingual datasets [10]. Another aspect-based two-level (i.e., sentence level and expression level) opinion annotation

[2]Hotel review dataset, `tripadvisor.com`

scheme is proposed in [121]. First, the sentence level annotation is implemented by using various attributes such as topic-relevant, opinions, polar-fact attributes. Second, the expression level scheme assigns five annotated types (i.e., polar-target, target, holder, modifier, and opinion expression) to mark spans.

### 2.3.3 Subjectivity Learning

*Subjectivity detection and classification* is a natural language process that aims to remove 'factual' or 'neutral' content from original articles [24]. Subjectivity detection techniques are geared towards understanding the psychological elements embedded within the writings by segregating objectives components from subjective pieces. Murray *et al.* proposed a pattern-based subjectivity detection approach, in which the subjective patterns are learned from both labeled and unlabeled data. This approach is implemented using n-gram word sequences with varying levels of lexical instantiation [79]. In this research, four subjectivity and polarity tasks were developed on spoken and written conversations. Marco *et al.* uses sentiment classification to discover summary sentences, or the short passages from a dataset of movie reviews [19]. This study extracted the overall sentiment of the review by filtering out potential noisy information.

*Opinion mining*, a sub-discipline at the crossroads of information retrieval and computational linguistics, is concerned with expressed opinion rather than topics of a text. *SENTIWORDNET* is a lexical resource in which each synset of *WORDNET* is associated with three numerical scores $Obj(s)$, $Pos(s)$, and $Neg(s)$, describing objective, positive, and negative levels of the terms contained in the synset [36]. Due to wide coverage and inclusion of qualifying labels, *SENTIWORDNET* has become an important online resource offering a practical interface for opinion mining applications [26]. Poria *et al.* [93] devised the first deep learning scheme, where a 7-layer deep convolutional neural network is utilized to tag each word in opinionated sentences as either aspect or non-aspect word.

*Polarity classification* focuses on distinguishing positive, negative, or neutral polarities of sentences in articles. Polarity classification has a wide range of applications such as tweets' sentiment analysis. For example, in the sentiment analysis of Italian tweets at the message level [39], Farias *et al.* provided participants a dataset, which includes two existing online corpus - *SENTI-TUT* [20] and *TWITA* [11] for the model training and testing purposes. Each of collected tweets is labeled with polarity (i.e., positive, negative, neutral, or mixed). Speriosu *et al.* improved tweets polarity classification methods by combining several knowledge sources with a noisily supervised label propagation algorithm [115]. The evidence shows that a maximum entropy classifier trained with distant supervision works better than a lexicon-based ratio predictor; the new classifier improves the accuracy for polarity classification on the held-out test set from 58.1% to 62.9%.

# Chapter 3

## Fake News Detection Model using Analytics Approaches

This chapter describes the conceptual and mathematical underpinnings of the proposed analytical model developed for establishing the credibility of news articles. We start This chapter by describing the composition of complete and incomplete sentences. Next, we formally define events and topics extracted from complete sentences. Boolean-value functions that distinguish fake events and topics from legitimate ones are subsequently described. Finally, we describe the mathematical formulation used for quantifying credibility of news articles.

### 3.1 Fake News Detection Model

#### 3.1.1 Topics and Events

Fake news could be detected through either topics or events. A news article $\alpha$ consists of a large number of sentences. We model article $\alpha$ as a set of $n$ sentences. Thus, we have

$$\alpha = \{\sigma_1, \sigma_2, ..., \sigma_n\}. \tag{3.1}$$

where each sentence (e.g., $\sigma_i$) is expressed as the following triple

$$\sigma_i = (U_i, V_i, O_i), 1 \leq i \leq n. \tag{3.2}$$

For the $i$th sentence $\sigma_i$ in (3.2), $U_i$ is a subject set; $V_i$ is a predicate set; and $O_i$ is an object set. Thus, we write these three sets as

$$U_i = \{u_i^1, u_i^2, ...u_i^{p_i}\}, 1 \leq i \leq n. \tag{3.3}$$

where $p_i$ is the number of subjects in subject set $U_i$.

$$V_i = \{v_i^1, v_i^2, ...v_i^{q_i}\}, 1 \leq i \leq n. \tag{3.4}$$

where $q_i$ is the number of predicates in predicate set $V_i$.

$$O_i = \{o_i^1, o_i^2, ...o_i^{r_i}\}, 1 \leq i \leq n. \tag{3.5}$$

where $r_i$ is the number of objects in object set $O_i$.

We categorize sentences in article $\alpha$ into complete sentences and incomplete sentences, depending on the existence of object set $O_i$ in the triple of the $i$th sentence. We refer to sentence $\sigma_i$ as a complete sentence if its object set $O_i$ does exist in the sentence triple; otherwise, sentence $\sigma_i$ is referred to as incomplete sentence (i.e., $O_i = \varnothing$). Hence, the set of sentences for article $\alpha$ can be rewritten as a combination of two disjoint sentence sets $S_{ic}$ and $S_{cp}$. Note that $S_{ic}$ is a set of incomplete sentences, whereas $S_{cp}$ is a set of complete sentences. Thus, we rewrite (3.1) as

$$\alpha = S_{ic} \ \cup \ S_{cp}, \ S_{ic} \ \cap \ S_{cp} \ = \varnothing, \tag{3.6}$$

where incomplete sentence set $S_{ic}$ is expressed as

$$S_{ic} = \{\sigma'_1, \sigma'_2, ..., \sigma'_m\}, \tag{3.7}$$

where incomplete sentence $\sigma'_i = (U'_i, V'_i, O'_i)$ has an empty object set. Thus, we have $O'_i = \varnothing$, and $1 \leq i \leq m$.

The models proposed in this study aims at detecting fake news from complete sentence set $S_{cp}$ rather than from incomplete sentence set $S_{ic}$; the reason is two-fold. First, incomplete sentences only bear information fragments due to the lack of objects. Second, among the four types of sentences from the language's perspective (i.e., *declarative sentences*, *interrogative sentences*, *imperative sentences* and *exclamatory sentences* [73]), declarative sentences – expressing statements – are incomplete sentence.

Let us consider three examples of incomplete sentences.

- *Incomplete Sentence 1:* Lucy is *lying*.
- *Incomplete Sentence 2:* It's *raining* outside.
- *Incomplete Sentence 3:* Water *evaporates* when it's hot.

These sentences have no objects because of the usage of intransitive verbs. The three incomplete sentences provide no details (e.g., Why Lucy lies? or what Lucy said?). In our proposed model, incomplete sentences $S_{ic}$ from article $\alpha$ are pruned during the pre-processing procedure.

Some sentences may contain fake information whereas others might have legitimate information. In what follows, we elaborate on the discrepancy between fake events and fake topics from the perspective of sentences. We start such a comparison by introducing events and topics in a formal way(Fig.3.1.1).

As described above, each event consists of a subject, a predicate, and an objects. Given sentence $\sigma_i = (U_i, V_i, O_i)$, an event set $E_i$ can be derived from subject set $U_i$, predicate set $V_i$, and object set $O_i$. Let us model such an extraction procedure as an event mapping function $x$, where $E$ is the Cartesian product of sets $U$, $V$, and $O$. Thus, we have

$$x : U \times V \times O \rightarrow E. \tag{3.8}$$

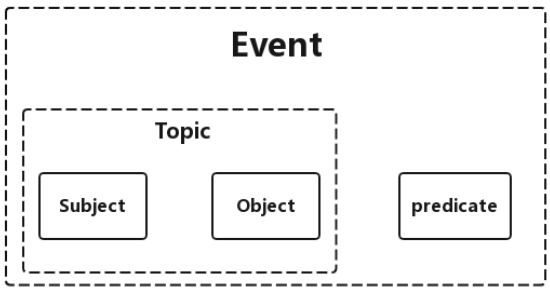



Figure 3.1: Relationship between event and topic

Suppose the sizes of sets $U_i$, $V_i$, and $O_i$ in sentence $\sigma_i$ are $p_i$, $q_i$, and $r_i$, respectively. The total number of events extracted from sentence $\sigma_i$ is a product of $p_i$, $q_i$, and $r_i$ (i.e., $p_i \times q_i \times r_i$). Hence, event set $E_i$ of the $i$th sentence $\sigma_i$ can be expressed as

$$E_i = \{e_i^1, e_i^2, ..., e_i^{w_i}\},\ w_i = \ p_i \times q_i \times r_i, \tag{3.9}$$

where $t_i$ is the total number of events extracted from sentence $\sigma_i$.

Let us make use of the following complete sentence (see also (3.2)) as an example to elaborate the definition of events extracted from sentences in the proposed model.

**Complete-Sentence Example 1:** A *computer* and a *car* require an *operator* and *power*.

The above complete sentence is expressed as triple $\sigma = (U, V, O)$, where subject set $U$, predicate set $V$, and object set $O$ are specified as $U = \{'computer', 'car'\}$, $V = \{'require'\}$ and $O = \{'operator', 'power'\}$. The set sizes $p$ (i.e., $|U|$), $q$ (i.e., $|V|$), and $r$ (i.e., $|O|$) are 2, 1, and 2.

In this example, sets $U$ and $O$ consist of multiple elements and set $V$ is a single-element set. We extract one element from each set to form an event. The total number $w$ of events extracted from $\sigma$ is four, because the product of $p$, $q$, and $r$ is 4 (i.e., $w = 2 \times 1 \times 2$).

The *event-set* $E$ of sentence $\sigma$ is expressed as $E = \{e^1, e^2, ..., e^4\}$, which is the Cartesian product of sets $U$, $V$, and $O$. Therefore, the events in set $E$ are written as

- $e^1 = \{'computer', 'require', 'operator'\}$,
- $e^2 = \{'computer', 'require', 'power'\}$,
- $e^3 = \{'car', 'require', 'operator'\}$, and
- $e^4 = \{'car', 'require', 'power'\}$,

The events modeled from a complete sentence (see also (3.2)) above articulate vital news information on what's happening. After events are extracted from complete sentences in a news article, *FEND* is positioned to compare the article's events against all the events from the legitimate news database. Some extracted events may identical to those in the database, whereas the others may be similar to the events in news database.

We now introduce the concept of *topics* to facilitate news article classification. Since topic-based clustering is more likely to group articles into a small number of clusters compared to event-based counterparts, topic-based clustering is more suitable than event-based news clustering. In our dataset, for example, multiple events are prone to sharing the same topic. This evidence implies that the number of events is larger than the number of topics in a given dataset. More specifically, the number of topics ranges from 4,987 to 29,877 in the top 20 clusters; the number of events in these clusters skyrockets to the range anywhere between 52,874 and 210,182.

Given a set $A$ of articles, we aim to classify all the news articles in $A$ into multiple news clusters in accordance to topics, which are defined as subject-object pairs. Let topic $t_i^j$ be the $j$th topic in sentence $\sigma_i$. Topic $t_i^j$ is created in the format of subject-object pair as

$$t_i^j = (u_i^j, o_i^j),\ u_i^j \in U_i\ \wedge\ o_i^j \in O_i \tag{3.10}$$

where $u_i^j$ is a subject in set $U_i$ and $o_i^j$ is an object in set $O_i$. Topics from sentence $\sigma_i$ form topic set $T_i$. Thus, we have

$$T_i = \{t_i^1, t_i^2, ..., t_{ii}^w\},\ w_i = \ p_i \times q_i \times r_i, \tag{3.11}$$

Each topic of sentence $\sigma_i$ (e.g., $t_i^j \in T_i$) can be directly derived from $\sigma_i$'s event set $E_i$ (see also (3.9)) by pruning the predicate of each event.

The relationship between event and topic is formally expressed below:

$$e_i^j = (t_i^j,\ v_i^{j_a}),\ 1 \leq j \leq w_i, \tag{3.12}$$

where $v_i^{j_a}$ is one component of **predicate-set** $v_i^j$.

Again, let us consider *complete-sentence example 1* (i.e., "A computer and a car require an operator and power"). Four events extracted from this complete sentence include $e_i^1, e_i^2, e_i^3$, and $e_i^4$ . The corresponding topics are listed below:

- $t_i^1 = \{'computer', 'operator'\}$,
- $t_i^2 = \{'computer', 'power'\}$,
- $t_i^3 = \{'car', 'operator'\}$,
- $t_i^4 = \{'car', 'power'\}$,

To articulate scenarios where multiple events may share the same topics, we consider another example.

**Complete-Sentence Example 2:** A *computer* and a *car require* and *consume power*.

In the above example, sets $U$ and $V$ consist of two elements; set $O$ is a single-element set. We obtain set $E = \{e^1, e^2, ..., e^4\}$, where we have

- $e^1 = \{'computer', 'require', 'power'\}$,

- $e^2 = \{'car', 'require', 'power'\}$,
- $e^3 = \{'computer', 'consume', 'power'\}$, and
- $e^4 = \{'car', 'consume', 'power'\}$,

The topic set in this example is $T = \{t^1, t^2, ..., t^4\}$; thus, we have

- $t_i^1 = \{'computer', 'power'\}$,
- $t_i^2 = \{'computer', 'power'\}$,
- $t_i^3 = \{'car', 'power'\}$,
- $t_i^4 = \{'car', 'power'\}$,

We show that topics $t_i^1$ and $t_i^2$ identical; similarly, $t_i^3$ and $t_i^4$ refer to the same topic. We conclude that in this example, there are two topics – (i.e. *('computer', 'power')* and *('car', 'power')*) – where each topic appears twice.

### 3.1.2 Fake Events and Fake Topics

Recall that an event is a triple containing a subject, a predicate, and an object (see (9)). The proposed model acquires a large number of legitimate news articles to build a knowledge base, which in turn assists in detecting untrustworthy articles in terms of credibility. In this study, we treat these legitimate articles as training data fed into the analytics model to build the knowledge base of legitimate news.

We introduce a boolean-valued function $f_E$ to detect if a given event is fake or legitimate. Thus, we have

$$f_E \ : \ U \ \times \ V \ \times \ O \ \rightarrow B_E, \tag{3.13}$$

where $B_E = \{0, 1\}$ is a boolean domain (i.e., 0 = fake event, 1 = legitimate event).

Similarly, we define a boolean-valued function $f_T$ to determine whether a topic is fake or not. Thus, we have

$$f_T \ : \ U \ \times \ O \ \rightarrow B_T, \tag{3.14}$$

where $B_T = \{0, 1\}$ is a boolean domain (i.e., 0 = fake topic, 1 = legitimate topic).

Let $f_V$ be a boolean-valued function to signify if a predicate is true or false. Hence, we have

$$f_V \ : \ V \ \rightarrow B_T, \tag{3.15}$$

where $B_V = \{0, 1\}$ is a boolean domain (i.e., 0 = false predicate, 1 = true predicate).

Given the $j$th event (i.e., $e_i^j$) of article $\alpha$, the value $f_E(e_i^j)$ is derived from the boolean-valued functions $f_T(t_i^j)$ and $f_V(v_i^j)$ as follows:

$$f_E(e_i^j) = f_T(t_i^j) \ \wedge \ f_V(v_i^j). \tag{3.16}$$

where event $e_i^j$ is comprised of topic $t_i^j$ and $v_i^j$.

### 3.1.3 Metric for Credibility and Performance Evaluation

The credibility of article $\alpha$ is computed through a function g($\alpha$), which is derived from boolean-value function $f_E$ (see (3.13)). The credibility of article $\alpha$ is measured as the percentage of legitimate events in the article. Thus, we have

$$g(\alpha) \ = \ \frac{\sum_{j=1}^{w_i}(f_E(e_i^j))}{w_i}. \tag{3.17}$$

where $\alpha$ is the test article, $w_i$ is the total number of events in article $\alpha$, $f_E(e_i^j)$ is the boolean-valued function defined in (3.13).

Given a news article $\alpha$, we apply equation (3.17) to quantify the article's credibility. If its credibility drops below a specified threshold (e.g., 0.6), article $\alpha$ will be treated as a fake news in *FEND*. For simplicity, we equally treat all events in articles during the credibility

calculation stage. In a real-world scenario, an article may tend to be a fake one if a key event isn't legitimate. The importance of each event might be represented by its frequency during the classification stage. Unfortunately, millions of events are generated during the fake-news detection phase. It may be assumed that events typically tend to be independent of one another. For example, when we test a dataset of 14221 articles, we extracted approximately $200,000$ topics; the number of events is in the order of magnitude larger that of topics. Consequently, it is impractical to rely on the weights of events to distinguish important events from unimportant ones. Alternatively, event importance could be specified by users, who can manually assign a large weight to an event that is more personally vital than others and vice versa.

Next, we introduce notation $d_{r\to r}$, $d_{r\to f}$, $d_{f\to r}$, and $d_{f\to f}$ to derive important performance metrics. Let $d_{r\to r}$ be the number of legitimate news articles truly verified as legitimate ones; $d_{r\to f}$ is the number of legitimate news falsely detected as fake news; $d_{f\to r}$ is the number of fake news treated as legitimate news; and $d_{f\to f}$ is the number of fake news correctly detected as fake ones. We summarize the notation in Table 3.1.

Table 3.1: Legitimate and fake news count notation for performance metrics

| Label \ Pred | real | fake |
|---|---|---|
| real | $d_{r\to r}$ | $d_{r\to f}$ |
| fake | $d_{f\to r}$ | $d_{f\to f}$ |

To measure the performance of the fake news detection system, we define four performance metrics using the notation listed in Table 3.1. These four measures, namely *accuracy*, *precision*, *recall*, and *F-score* are widely adopted in prior studies ([120, 133]).

Let $A$ be an *accuracy* rate, which is the percentage of news that are correctly identified as fake or real news among all news. Thus, $A$ is expressed as (3.18)

$$A = \frac{d_{r\to r} \ + \ d_{f\to f}}{d_{r\to r} \ + \ d_{r\to f} \ + \ d_{f\to r} \ + \ d_{f\to f}}, \tag{3.18}$$

$P$ denotes a precision rate, which is the fraction of accurately detected fake news among all the detected fake news. We express precision $P$ as (3.19).

$$P = \frac{d_{f \to f}}{d_{r \to f} \ + \ d_{f \to f}}, \tag{3.19}$$

$R$ represents a *recall* rate, which is the fraction of detected fake news among all the ground truth fake news. Hence, recall $R$ can be written as (3.20).

$$R = \frac{d_{f \to f}}{d_{f \to r} \ + \ d_{f \to f}}, \tag{3.20}$$

$F$, or *F-score*, is the harmonic mean of precision $P$ and recall $R$. Thus, we derive $F$ from $P$ and $R$ as (3.21).

$$F = \frac{2 \times P \times R}{P + R}. \tag{3.21}$$

## 3.2 Research Framework of FEND

In This section, we first introduce the framework describing the proposed methodological approach used for fake news detection. Next we describe the web crawler design with corresponding pseudocode. Third, we illustrate the pipeline of data processing. Finally, we describe the analytics approaches used for clustering and classification of fake news. The entire framework and various components are integrated together to develop a novel fake news detection system, referred to as *FEND* ($F$ak$E$ $N$ews $D$etection).

Figure 3.2 presents the framework that guides the design and development of *FEND*. *FEND* is driven by a ground-truth knowledge base comprised of legitimate-news clusters and corresponding verb lists. The model-training framework that drives the functioning of *FEND* judiciously creates clusters based on topics, meaning that news articles in the same cluster share a set of topics. Articles classified in separate clusters have distinctive topic sets.

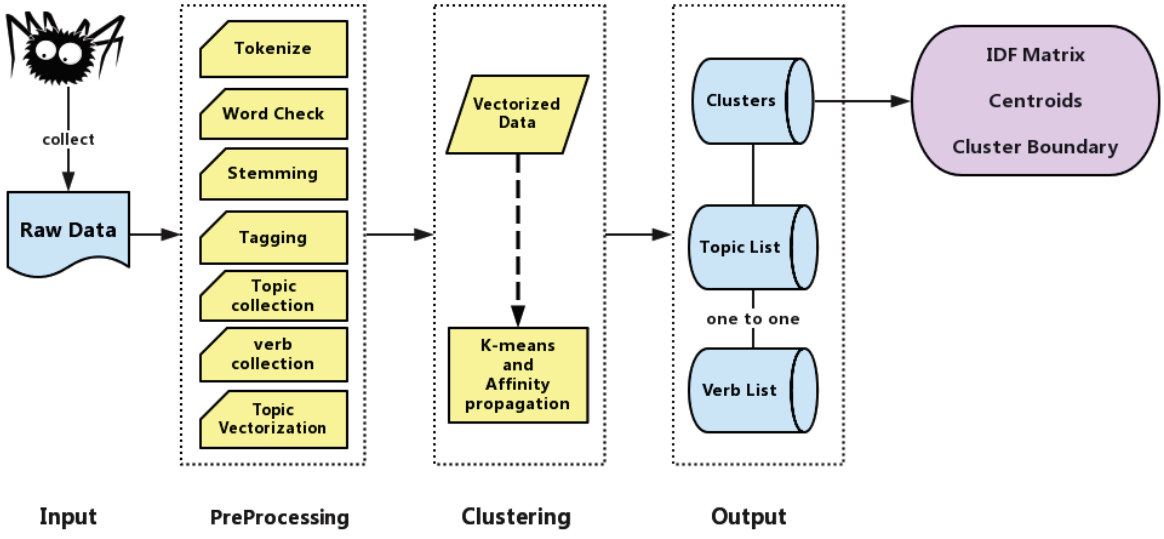



Figure 3.2: Model-training framework builds ground-truth knowledge bases by classifying legitimate news articles into news clusters.

Fake news is detected through two subsequent phases (see Fig. 6.6), namely, (1) fake-topics detection using news clusters and (2) the fake-predicate detection through verb comparisons. News clusters are assembled according to news topics; a news article is believed to be fake when (1) the news cannot be classified into any cluster or (2) its verbs have a low similarity level with the corresponding verbs in its news cluster.

Handling the synonyms of words is a critical issue to be addressed. Our proposed models address this issue in several ways by incorporating approaches such as lemmatization, stemming, and parts-of-speech tagging to ensure redundant or noisy data is removed during the pre-processing phase.. In particular, we employ a list of functions available from the WordNet library [75, 40] to detect synonyms of the predicate of an event in a verb list. Such a detection procedure is outlined as follows:

- the first step is to exploit all synonym arguments of predicate and each word in the verb list.

- the second step is to traverse and compare the current tested predicate with each verb in the contrasted verb list with respect to synonym arguments, thereby obtaining the largest similarity between argument pairs.

- the last step is to collect a specified number (e.g., 100) of synonym pairs to approximate a low similarity boundary (e.g., 86.6%), which is compared against the largest similarity obtained in step 2 to determine the synonyms.

To illustrate the above three steps, let us consider the following example. The word '*consume*' has six synonym arguments, namely, (1) '*devour.v.03*', (2) '*consume.v.02*', (3) '*consume.v.03*', (4) '*consume.v.04*', (5) '*consume.v.05*', and (6) '*consume.v.06*'. The argument format is '*Word.POS.Sense*', where *POS* is word type and *Sense* is the word's frequency count for a particular meaning of that word. The word 'expend' only contains two synonym arguments, including '*use.v.03*' and '*spend.v.02*'. Next, we compare and calculate the similarity of each argument pair, where one argument is from word '*consume*' and another one is from word '*expend*'. The last step is to greedily pick the largest value among all similarities as a reference to determine if these two words are synonyms. The framework presented in figure 3.2 and 3.3 present seamless integration of the training and the testing procedure.

The framework presents seamless integration of the training procedure and the testing procedure, and comprises three modules: the training data collection module, the data pre-processing module, and the news clustering module. The training data collection module acquires raw data from legitimate news websites and removes noise such as advertisements; we implement this module using a custom web crawler designed specifically for building the repository and performing in a data streaming fashion. The data pre-processing module integrates an array of text processing techniques to extract topics and events from the newly collected news data. The clustering module classifies the news articles into separate groups according to the extracted events. The output database maintains news clusters, each of which is coupled with a corresponding verb list. This output database serves as ground truth to validate the credibility of other incoming news articles.

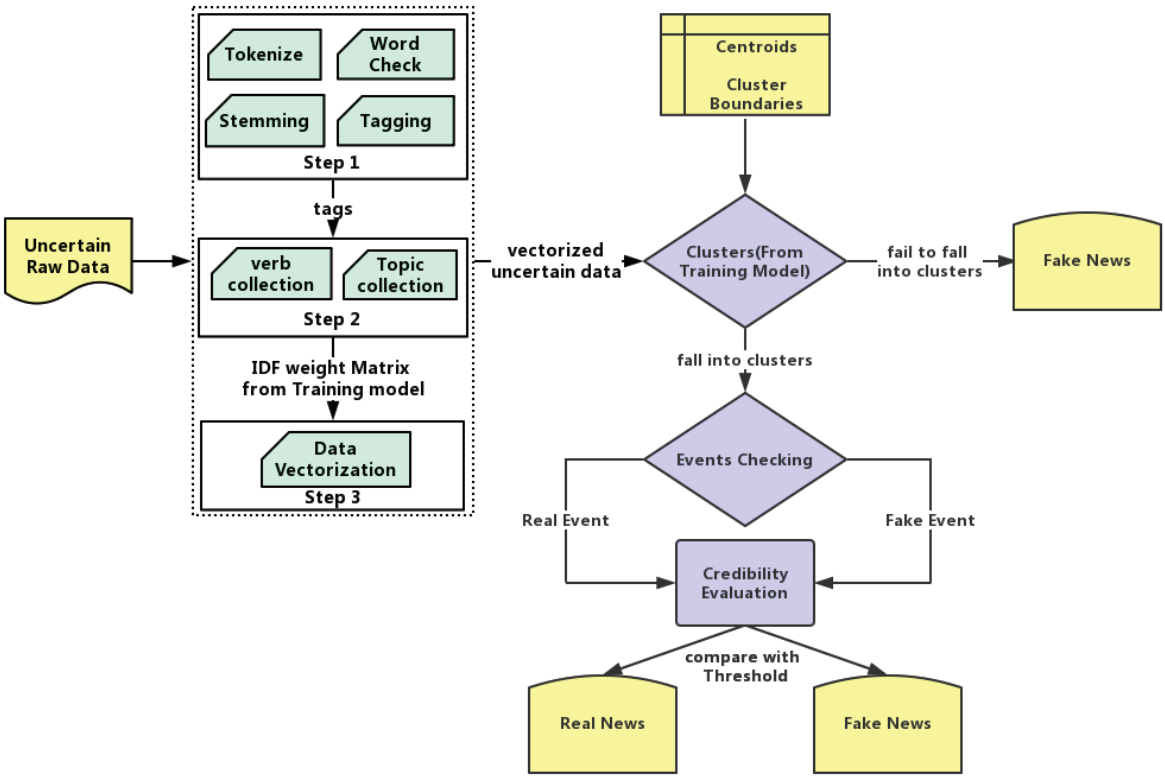



Figure 3.3: Fake news detection framework includes two filters: (1) the news that cannot be classified into any cluster or (2) its verbs have a low similarity level with the corresponding verbs in its news cluster.

Figure 3.3 outlines the framework that applies the trained model built from the existing legitimate news data to detect fake news. The fake news detection framework consists of a data pre-processing module, a filtering module, and a verification module. The input of the data pre-processing module is the same as that of the first module in the aforementioned model training framework. The input data (i.e., raw data) is either extracted from training data or collected from fake news websites using a web crawler. The pre-processing module not only deploys all the components of the training data pre-processing module, but also vectorizes topics of testing data using the *IDF weights matrix*(Inverse document frequency), which is extracted from the training procedure to ensure the consistency of vectorization. Similarly, the pre-processing module exploits topics and events of each testing data to produce vectorized data as well as the corresponding *verb list*. The second module (i.e., the

filtering module) embraces two-layer filtering procedure. The first layer filtrates testing data that fail in falling into any news cluster; these type of news are detected by the filter as fake news. The second layer of the filtering module is in charge of comparing the verb list of each remaining testing data with the verb list of the corresponding cluster to quantify the credibility of each of remaining testing data. The last module (i.e., the verification module) identifies fake news and real news using a threshold, which is specified in accordance to the credibility of all the testing data. The fake news detection framework discovers fake news and outlier news that falls outside of extracted news clusters.

We make use of the following example to shed light on how our algorithm can be executed from the training phase to the testing phase.

**Example 3**: Let us consider two clusters obtained in the training phase. We list the two clusters in which cluster 1 contains three articles (i.e., articles 1, 3, and 4) whereas cluster 2 is comprised of the other three articles (i.e., article 2, 5, and 6); article 7 is a testing sample. We summarize the two clusters and the tested news along with their topics and verbs in the following table.

Table 3.2: Two clusters are obtained in the training phase. Topics and verbs are extracted for the two clusters (i.e., clusters 1 and 2) and the test data (i.e., article 7).

| **Cluster** | **Articles** | **Topics** | **Verbs(In format: topiccorresponding verb list)** |
|---|---|---|---|
| Cluster 1 | 1, 3, 4 | 'abc', 'abd', 'acd' | a{a'}, b{b', b"}, c{c', c", c""}, d{d', d", d"', d""} |
| Cluster 2 | 2, 5, 6 | 'aef', 'bef', 'cef' | a{a'}, b{b', b"}, c{c', c", c""}, e{e', e"}, f{f', f", f"'} |
| | 7 (tested article) | 'defg' | d{d', d""'}, e{e"'}, f{f", f"', f""}, g{g', g"} |

Now we evaluate the credibility of the tested data in the above example. The tested data (i.e., article 7) belongs to cluster 2, in which article 7's credibility can be quantified as follows. In topic listd, e, f, g, element topic d includes one legitimate and one fake verb. Similarly, we determine the number of legitimate and fake verbs for the other topics (i.e., e, f, and g) in the list. Thus, we have e(0 legitimate & 1 fake), f(2 legitimate & 1 fake). In this example, topic g turns out to be an emergent topic in which all the verbs (i.e., g' and g") are treated as fake. Hence, we have g(0 legitimate & 2 fake). For tested article 7, its total

number of verbs (a.k.a., events) is eight (8), among which three (3) are legitimate verbs. Consequently, the credibility of the tested data (i.e., article 7) is 3/8 or 37.5%.

The fake news detection framework discovers fake news including outlier news by identifying news items that fail to fall into any news clusters. The *FEND* system can be applied in two phases, namely, fake topics detection and fake events detection. We subsequently implement this two phase framework to develop the *FEND* system that can be applied for detecting fake topics and detecting fake events. Subsequent sections describe various components of the fake news detection framework and *FEND* system.

The three metrics applied to decide if a news is in a cluster include IDF Matrix, coordinates of centroids (a.k.a, vector of centroids), and boundary of centroids (a.k.a, the largest distance between centroids and points in their cluster). IDF matrix is derived from feature weights using the TF-IDF technique explained in section 4.2. In what follows, we summarize the procedure utilized for deciding if a given tested news is in a cluster.

First step is to extract topics from the given news. This step is also referred to as 'feature extraction'. Second step is to vectorize the news using IDF-Matrix and its topics retrieved from Step 1. Third step is to calculate the distance between vectorized news (i.e., the tested one) and the centroid of a current cluster. Finally, the last step is to decide if this news belongs to the compared cluster or not. The news belongs to the cluster if the distance calculated in the previous step is smaller than cluster-1's boundary from centroid. Otherwise, this news is an outlier.

### 3.3 Data Processing and Clustering

In the fake news detection framework (see also Fig. 3.2), the raw data aggregated by the web crawler drives the development of the ground-truth and fake news database. To conduct extensive experiments, we develop a universal web crawler to retrieve news from a various websites to be tested by the fake news detection framework. This web crawler – described in detail in Appendix A – facilitates the pre-processing phase of *FEND* with input data as

a set of text files, where each file is an individual news accompanied by author information and published data.

We construct a word-processing pipeline (see also Fig. 3.2) that is capable of extracting events and topics by the virtue of triple extractions using OIEs, word tokenization, word verification, word stemming, word property tagging, event collection, event decomposition (i.e., topic collection), and topic vectorization.

Raw data acquired using the web crawler in the previous steps is subsequently subjected to a series of pre-processing transformations for annotation as well as topic and event extraction. Stanford CoreNLP library [68], which provides a pipeline architecture for performing a sequence of linguistic annotation procedures namely, tokenization, tagging, word check, stemming and part-of-speech tagging, was used with Natural Language Toolkit [17], a python based library for natural language processing. Stanford CoreNLP library is among the most popular and advanced open-source libraries available for performing the pre-processing of raw corpus data. Two separate corpus comprises of ground truth and fake news datasets.

The tokenization algorithm segments each document from the ground truth and fake news corpus into a sequence of sentences, each of which is then rendered into a series of 'tokens' – i.e., a single word or a combination of continuous multiple characters. The output of tokenization algorithm is fed to the stemming algorithm, which performs the morphological analysis of each token generated from the tokenization process. This approach removes the redundancy in word frequency counts by truncating the words back to their roots. For example, separate occurrence of words 'know' and 'knowing' within a document is counted as two instances of occurrence of the word 'know' as 'ing' is stripped from the end. The output from stemming is then passed onto part-of-speech (POS) tagging where each tokenized sentences is further annotated with POS tags for entity and relationship detection. These steps help annotate the data into triple representation, which is a combination of subject, predicate and object as defined in equation 3.2.

The process of vectorization converts tokens into numerical vectors for subsequent topic generation. We used Term Frequency-Inverse Document Frequency (TF-IDF) weighted term approach to accomplish this task [130]. This approach allows weight the term frequency of tokens by the appropriate weight base don their importance for the document. The TF-IDF approach is integrated with document pre-processing pipeline and uses scikit learn library, which is a python based library for machine learning algorithms. TF-IDF approach evaluates the product of term frequency (i.e., TF) measure of each topic occurrence within a document weighted by its importance (i.e., IDF). TF represents the term frequency of each topic while IDF computes the importance of the topic and generates weight matrix for each topic within the dataset. The raw values of TF-IDF are evaluated using below equation

$$tf\text{-}idf(t, d, D) = tf(t, d) \times idf(t, D) \tag{3.22}$$

$t$, $d$ and $D$ denotes a topic, an article and a set of articles in the dataset respectively. $tf(t, d)$ is used to calculate the frequency of each topic appeared in each article. IDF is evaluated using the below computation

$$idf(d, t) = log[\frac{n}{df(d, t)}] + 1 \tag{3.23}$$

Finally, the Euclidian norm is then applied to the raw values of TF-IDF for normalization.

Figure 3.4 illustrates the process of generating events and topics from news articles. The triple data store is an internal store connecting the OIE tools and the word-processing pipeline. An event dataset aggregates events extracted by the word-processing pipeline. We implement a module to split the event dataset into a topic dataset and a verb dataset. This enables the classification of news articles in subsequent stages (also see (3.10) in Section 3.1.1). In a later phase, the topic dataset drives topic-based news clustering.

We apply two clustering methods, namely k-means and affinity-propagation, to train the models on the three datasets. This strategy allows us to validate our fake news detection

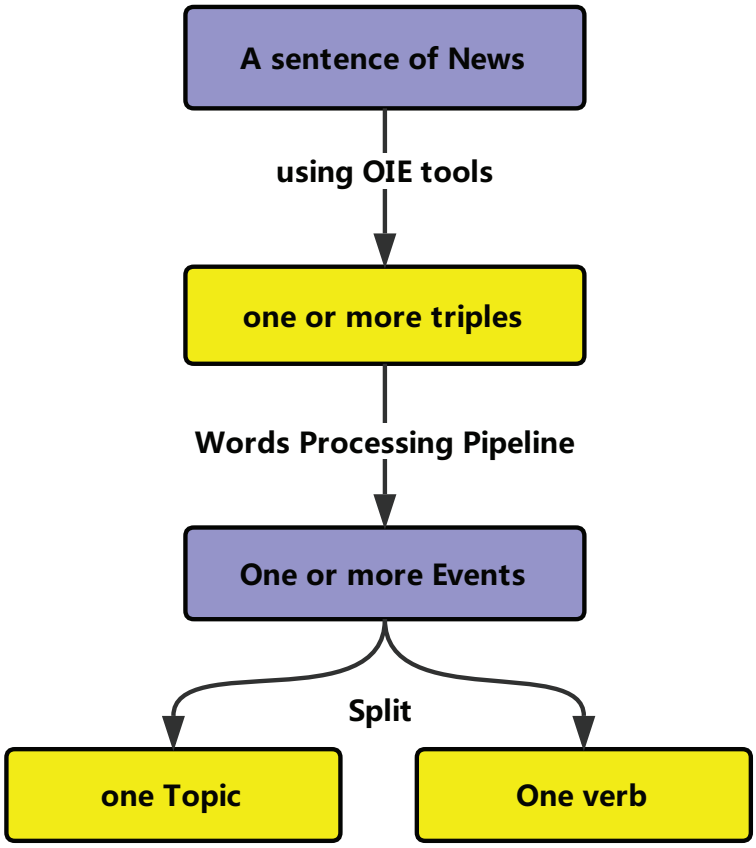



Figure 3.4: Process of generating events and topics from news articles: The triple data store connects the OIE tools and the word-processing pipeline. The topic and verb datasets are derived from the event dataset and a verb dataset.

theory that treats topics as features of news articles. The rationale behind deploying these two algorithms is two-fold. First, from perspective of implementation, these two clustering algorithms offer ease of implementation in comparison to the other complicated ones. Second, we pick one supervised algorithm (i.e., K-means) and one unsupervised algorithm (i.e., affinity propagation) as an epitome of each algorithm category.

K-means is a classic clustering algorithm in data mining and divides a dataset into $k$ clusters by setting the value of $k$ in advance. K-Means identifies the best centroids by alternating between assigning data points to clusters based on centroids and choosing centroids determined by data points to clusters. The process of the k-means clustering algorithm proceeds as: (1) Select $k$ cluster centroids randomly. (2) calculate the Euclidean Distance between each point and centroids, then save the current clusters. (3) re-evalulate the distance of data point in each cluster and select the new centroids. (4) Repeat Steps 2 and 3 $n$ times, or until the clusters converge.

Affinity Propagation, proposed by [35], is a popular technique due to its simplicity, ease of applicability, and performance. This scheme relies on the concept of message passing among data points until convergence. Unlike K-means, affinity propagation doesn't require a-priori specification of the number of clusters. It measures similarity between the pairs of data points while simultaneously considering all the data points as potential exemplars. Real-valued messages are exchanged among data points until a high-quality set of exemplars and corresponding clusters gradually emerges. We articulate the affinity-propagation clustering algorithm by first calculating the responsibilities. Responsibility $r(i, k)$ reflects the accumulated evidence for how well-suited point $k$ is to serve as the exemplar for point $i$, taking into account other potential exemplars for point $i$. Responsibility is sent from data point $i$ to candidate exemplar point $k$. Next, we calculate availability. Availability $a(i, k)$ represents the accumulated evidence for how appropriate it would be for point $i$ to choose point $k$ as its exemplar, taking into account the support from the other points that point $k$ should be an exemplar. Availability is delivered from candidate exemplar point $k$ to point $i$.

In our experiments, we employ the affinity-propagation (AP) algorithm to perform data clustering. After obtaining the number of news clusters, we apply the number of clusters to the value of k to configure the K-means algorithm. This experimental sequence is important, because K-means algorithm takes the number of clusters as an input parameter. After comparing the clustering results of the K-means and AP algorithms, we discover that the

two algorithms yield identical clustering results for input datasets. As such, the credibility of tested news articles remains unchanged regardless of AP or K-means deployed in the training phase. Consequently, the comparisons between these two clustering algorithms are ignored.

In summary, this chapter describe the design of the proposed FEND system, which comprises of a training module and a testing module. In the training module, we make use of a list of text processing techniques to extract topics and events of real news, then utilize TF-IDF and K-means algorithm to process the extracted topics and events as a knowledge base. In the testing module, there are two fake news detection filters are included: (1) fake topic detector (2) fake event detector. Both of the two detector can detect fake news according to the extracted topics and events respectively by comparing with the knowledge base.

Chapter 4

A Computational Approach for Real Time Detection of Fake News

## 4.1 The Analytic Model of RT-FEND

In this study, a news article is modeled as a list of events extracted from completed sentences using Stanford's CoreNLP [69]. Given event lists, topics and verbs are separated from events to create features and verb lists.

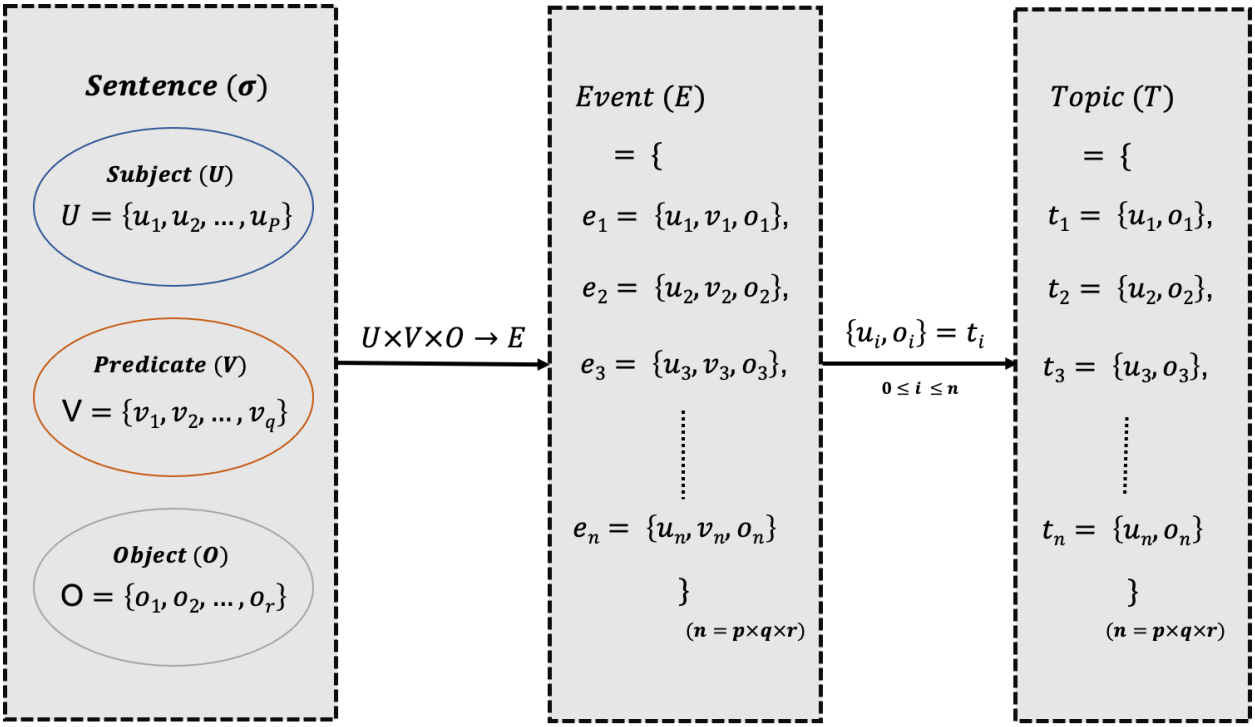



Figure 4.1: A example to explain the process of system testing. Subject set is $U$, predicate set is $V$ and an object set is $O$. $p$ is the number of subjects in subject set $U$, $q$ is the number of predicates in predicate set $V$, $r$ is the number of objects in object set $O$, $E$ is the event set of sentence $\sigma$, and $t_i$ is the corresponding topic of event $e_i$.

A news article comprises of several sentences. A completed sentence includes all of the three components, namely, subject, object and predicate. Figure 4.1 describes the formal

representation of sentences, subject sets, predicate sets, object sets, event lists, and topic lists, where $p$, $q$, and $r$ represent the number of subjects in subject set $U$, the number of predicates in predicate set $V$, and the number of objects in object set $O$, respectively. The number of all topics, inclusive of duplicates, in set $E$ is equal to $p \times r$. Each event $e_i$ in set $E$ are independent of the other events unless all the components in two contrasted events are identical. A similar assumption applies to topics. The number of events in set $E$ is equal to $p \times q \times r$. As a result, the numbers of events and topics surges dramatically along with a growing number of data points. For instance, given a data set containing a total of 14,231 news articles, the number of topics is approximately 329 thousands; the number of events exceeds 1.8 millions. To address this problem referred to as 'the curse of dimensionality', we reduce the number of topics by consolidating similar topics into a single but large topic group. This goal is achieved by applying the *WordNet* library [75] to facilitate topic similarity comparison using the following equation:

$$S_{ij} = \frac{s_{ij}^{u} + s_{ij}^{o}}{2} \tag{4.1}$$

where $s_{ij}^{u}$ is subjects similarity between topic $i$ and topic $j$, $s_{ij}^{o}$ is objects similarity between topic $i$ and topic $j$.

We apply a topic-merging mechanism to merge topics with high similarity in WordNet (see also Fig. 4.1). This mechanism is implemented by recursively (1) calculating the similarity of randomly selected 1000 topics and (2) obtaining synonymous topics from these topics. The mechanism chooses the lowest similarity of synonymous topics as a threshold (e.g., 0.85) to prune redundant topics. This procedure of the topic-merging mechanism is better illustrated through an example. Let's consider two topics: Topic 1 includes $'investigation, solution'$ and Topic 2 includes $'survey, answer'$. *investigation* and *survey* are often used interchangeably in many textual environments; the same applies to*solution* and *answer*. More specifically, the similarity score of subjects *investigation* and *survey* is measured as 0.94 (i.e., $s_{12}^{u}$), the similarity score of objects *solution* and *answer* is quantified

as 0.88 (i.e., $s^o_{12}$). Therefore, the similarity score between Topic 1 and Topic 2 (i.e., $S_{12}$) can be derived from scores $s^u_{12}$ and $s^o_{12}$. Thus, the similarity between the two topics in *Example 1* is written as $S_{12} = \frac{0.94+0.88}{2} = 0.93$ (see also Eq. 4.1).

We illustrate this concept by demonstrating an example that provides insight into the model and explains the underlying processing used by the proposed system for fake news detection using real world data. Figure 4.2 demonstrates various training module processes such as triple sets extraction, event extraction, topic merging, textual data vectorization and data clustering. The process starts with the collection of a list of legitimate news items for the purpose of knowledge base construction. First, NLTK- and Stanford openIE- based techniques are applied to clean data and extract triple sets(e.g., 5 and 5 triple sets are extracted from two messages) from each sentence of pre-processed news articles. Then, we apply other NLP-based techniques such as word stemming, lemmatization, stop words, and nouns selection to generate events(e.g., 11 and 5 events are extracted) and corresponding topics (e.g., 10 and 4 topics respectively are extracted). In the use case, the number of extracted events for each data are 11 and 5 respectively(i.e., e, f, g, ...,h). The number of extracted topics are 10 and 4 respectively. After applying the topic merging mechanism, the total number of topics is reduced from 14 to 13. Next, we vectorize news data based on their topics using TF-IDF algorithm, which applies weights to the term frequency based on how frequently a term appears in other news documents [130]. For example, the term frequency of a particular term is weighted down if its frequency of occurrence is high in other news documents. $n$ is the number of topics in the dataset, where $n >> i, j, k, ..., l$ and $n \leq i + j + k + ... + l$. Finally, we collect all verbs of each topic in dataset to build verb lists and apply clustering algorithms to gather similar news to generate data clusters for building knowledge base.

Figure 4.3 shows various testing module processes that include events extraction, vectorization, clusters checking, and verb check. In testing module, we randomly select an uncertain news article to test the trained model. Similar to training module, Figure 4.3

performs extraction of triples, events and topics from news data, which then lead to the development of data clusters in the subsequent steps of the training process. The resulting weighting matrix (step 4 in figure 1 ) can be used to vectorize new textual data. Then we identify centroid coordinate of clusters and corresponding boundary(see step 6 in figure 4.2) to examine if the uncertain news can fall into the clusters. If it can't, we treat the uncertain news as fake. Otherwise, the verb lists will be used to calculate the credibility of the news item. Finally, the testing module compares this credibility with a preset threshold.

## 4.2 Analytics Framework of RTFEND

This section describes the framework of the proposed methodology, for real-time collecting, processing, managing and analysis of fake news. The framework (Fig. 4.4) contains a flow of streaming data transferred and processed in the three modules, namely, (1) real-time data collection, (2) streaming-data processing, and (3) knowledge-base construction.

News data acquisition is performed by the real-time data collection module, which forwards newly harvested data to the streaming data processing module. After processing the streaming data, the knowledge base construction module builds legitimate news knowledge base from the raw news data. The first batch of data in the Resilient Data Definitions (RDD) data structure format (see also the streaming-data processing sub-system) makes use of an array of five data pre-processing submodules to create an initial knowledge base, which is comprised of article clusters and corresponding verb lists. The techniques implemented in the five pre-processing modules include, but not limited to, NLP-related lemmatization and stemming, events and topics extraction, and topic-based articles vectorization. The knowledge base is repeatedly updated after subsequent data are fed into the knowledge base construction module from preceding modules. The subsequent data are represented as the 2nd batch and the like in the streaming-data processing sub-system as demonstrated in Fig 4.4.

### 4.2.1 Real Time Data Collection and Streaming-Data Processing

The data-collection module periodically acquires news data from the Internet using a web crawler, storing and archiving collected data stored in a no-sql database, Hadoop distributed file system (HDFS). The web crawler uses a customized algorithm to collect real-time data from a wide variety of legitimate news websites. The acquired real-time news data are dispatched to the streaming-data processing module, which converts the data format from the data-frame into the $RDD$. The data format conversion is critical and indispensable, because output data of the real-time data collection module is managed in the data-frame format whereas the knowledge base construction subsystem simply handle the $RDD$ format.

In addition to data-format conversion, another vital functionality of the streaming-data processing subsystem is to uniformly divide streaming data into equal-sized batches. Each batch of data is scheduled and handled in a given sliding window. For example, in the streaming-data processing subsystem in Fig. 4.4, *time 1* and *time 2* are two sliding windows for the first and second batches of data, respectively. The first batch of data serves as an initial data point for the next subsystem (i.e., knowledge-base construction). Given the first batch of news data in sliding window 1, an initial knowledge-base is constructed in a static manner. In other words, the proposed system generates a temporary knowledge-base derived from all the news articles in the first batch of data handled in the first sliding window (a.k.a, *time 1*). Upon the arrival of the second batch of data, the knowledge-base construction module statically updates the news knowledge base by processing the second batch. Similarly, as impending batches become available, the real-time data collection subsystem sequentially processes all the data in a streaming fashion.

### 4.2.2 Knowledge-base Construction

We articulate the process of building a knowledge bases by transferring collected data, converting data types, and analysing data in a high-performance computing environment.

As shown in figure 4.5, news articles that have been converted into data frames in the preceding modules are first fed to this module as *RDD*. All *RDD*-formatted articles are evenly split into '$n$' pieces before dispatching to multiple slave nodes on distributed computing environment such as Apache Spark, where '$n$' is the number of slave nodes in the computing cluster. Next, *RDD*-based raw data (i.e., news articles) are transformed as corresponding *RDD*-based topics and verbs and dispatch the topics and verbs into the slave nodes. Being treated as features of articles, all the topics guide the clustering process carried out in the slave nodes. The name node collects and gathers sub-results from the slave nodes, followed by grouping similar vectors into one slave for the re-clustering process. These steps facilitate the implementation of clustering algorithm using real time processing.

The streaming process mechanism is mandatory for iteratively upgrading centroids for developing clusters. Such a mechanism allows our model to manage streaming news data to update the knowledge base in a real-time manner [5]. The coordinates for the current centroid ($c_(s+1)$) is evaluated as

$$c_{s+1} = \frac{c_s p_s \alpha + b_s q_s}{n_s \alpha + q_s} \tag{4.2}$$

$$p_{s+1} = p_s + q_s \tag{4.3}$$

where $c_s$ is the previous center for a news cluster, $p_s$ is the number of points assigned to the news cluster so far, $b_s$ is the new cluster center from the current processing batch, and $q_s$ is the number of points added to the cluster in the current batch of data. The decay factor $\alpha$ controls how much past data should be incorporated in the current computing cycle and is analogous to an exponentially weighted moving average. For example, if $\alpha$ is set to 1, then all data will be used from the beginning; if $\alpha$ equals to 0, then only the most recent data will be evaluated.

## 4.3 Performance Optimization

We utilize Spark, a memory-based parallel computing framework, to fetch the data and feed into the main memory for processing. Improving the performance of memory is extremely critical for real time processing. Various memory management techniques such as execution and storage utilization in Spark speed up the efficiency of data processing in our system [135].

### 4.3.1 Average Cold-Start Ratio

We make use of existing legitimate news to simulate a streaming-data-collection environment, in which the proposed system is required to process a news dataset in a batching mode. Time spent in such data processing largely depends on batch size, which in turn affects the system's performance. It is arguably true that electing an optimal value for batch size is reliant on empirical studies.

We introduce a novel method to formally configure the batch size for optimal memory allocation. This model is comprised of a list of experiments based on a sampling dataset to select the best batch size for optimizing the performance of the proposed system. Thus, we have

$$\Pi = \{\pi_1, \pi_2, ..., \pi_n\} \tag{4.4}$$

where $\pi_j$ is the $j$th experiment set and $n$ is the total number of experiments sets when the number of slave nodes is varied. For example, $n$ will be equal to 4, if there are four options to setup the number of nodes (e.g., 0, 2, 4, and 8) in a computing cluster. Each experiment set (e.g., $pi_j$) contains a group of experiments. More formally, the $j$th experiment set $\pi_j$ can be expressed as

$$\pi_j = \{\beta_{1j}, \beta_{2j}, ..., \beta_{mj}\}, 1 \leq j \leq n. \tag{4.5}$$

where $m$ is the total number of candidate batch sizes. For instance, setting m to 3 signifies that there are three batch-size options (e.g., 10MB, 20MB, and 30MB) in experiment set $\Pi$. Hence, the total number of experiments in set $\Pi$ is $m \times n$.

The number of batches and batch size are correlated. Given the total data volume, we derive the number of batches by dividing the total data volume by the batch size. Let $K$ be the total size of a dataset; we denote $k_i$ as the batch size of the $i$th experiment. The number of batches $l_i$ is obtained from dataset size $K$ and batch size $k_i$ as

$$l_i = \frac{K}{k_i} \tag{4.6}$$

In practice, we first configure an optimal value for batch size $k_i$, followed by determining the number $l_i$ of batches using Eq. 4.6.

Given experiment $\beta_{ij}$ in set $\pi_j$, we denote $T_{ij}$ as the total time spent in processing news data in experiment $\beta_{ij}$ when the number $l_i$ of batches is obtained from Eq. 4.6. The total processing time $T_{ij}$ is a summation of the processing time of each batch in experiment $\beta_{ij}$. Let $t_{ij}^s$ be the spending time of the $s$th batch in experiment $\beta_{ij}$. Thus, time $T_{ij}$ can be written as

$$T_{ij} = t_{ij}^1 + t_{ij}^2 + ... + t_{ij}^{l_i} = \sum_{s=1}^{l_i} t_{ij}^s. \tag{4.7}$$

An intuitive expectation is that all the batches in experiment $\beta_{ij}$ share a similar processing time. Surprisingly, the processing time of the first batch is an outlier. For example, time $t_{ij}^1$ is smaller than the other processing times of the subsequent batches (e.g., $t_{ij}^2, ..., t_{ij}^{l_i}$). To capture the discrepancy between $t_{ij}^1$ and the other processing times in experiment $\beta_{ij}$, we introduce the following cold-start ratio $\alpha_{ij}^s$, which is a ratio between $t_{ij}^1$ and $t_{ij}^s$ $(1 < s \leq l_i)$. Thus, we have

$$\alpha_{ij}^s = \frac{t_{ij}^1}{t_{ij}^s}, 1 < s \leq l_i. \tag{4.8}$$

where $t_{ij}^s$ is the processing time of the $s$th batch in experiment $\beta_{ij}$ of the $j$th experiment set.

Because the processing times of all the batches except the first one are very similar in length, we conclude that cold-start ratios are dependent of data-mining algorithms and modeling frameworks. Hence, given a batch size and a settled cold-start ratio, our system can predict the processing time of future batches. We observe the distribution of all the cold-start ratios, followed by calculating the average cold-start ratio in (4.9).

$$A_{ij} = \frac{\sum_{s=2}^{l_i} \frac{t_{ij}^1}{t_{ij}^s}}{l_i - 1} = \frac{\sum_{s=2}^{l_i} \alpha_{ij}^s}{l_i - 1} \tag{4.9}$$

where $A_{ij}$ is an average cold-start ratio of $\beta_{ij}$

If one updates computer-system configuration, the time spent in processing batches will be changing. For example, a batch of 20MB requires 100-second processing time on an 8-node cluster; this batch size must consume 180 seconds on a 4-node cluster. Nevertheless, a surprising finding is that the cold-start ratio remains unchanged in the two different system configurations. We conclude that cold-start ratios are independent of system configurations. Thus, we have

$$A_i = \frac{\sum_{j=1}^{m} A_j}{m} \tag{4.10}$$

where $A_i$ is the average cold-start ratio of experiment set $\forall j \subset m, \beta_{im}$ for a given batch size $k_i$.

### 4.3.2 Experimental Validation

We conduct a sequence of 12 experiments to illustrate our proposed holistic approach of choosing the most appropriate batch size to optimize the overall performance in the proposed system. Table 4.1 summarizes the configurations (e.g., batch size and memory size) of the experiments.

Table 4.1: The number of nodes, batch sizes, and memory sizes of the 12 experiments. $\beta_{ij}$ is the $i$th experiment in group $j$ (see also (4.5)).

| **Experiment No.** | **Num of Nodes** | **batch size** | **Memory Size** |
|---|---|---|---|
| $\beta_{11}$ | 1 | 10MB | 8GB |
| $\beta_{21}$ | 1 | 20MB | 8GB |
| $\beta_{31}$ | 1 | 30MB | 8GB |
| $\beta_{12}$ | 3 | 10MB | 16GB |
| $\beta_{22}$ | 3 | 20MB | 16GB |
| $\beta_{32}$ | 3 | 30MB | 16GB |
| $\beta_{13}$ | 5 | 10MB | 24GB |
| $\beta_{23}$ | 5 | 20MB | 24GB |
| $\beta_{33}$ | 5 | 30MB | 24GB |
| $\beta_{14}$ | 9 | 10MB | 32GB |
| $\beta_{24}$ | 9 | 20MB | 32GB |
| $\beta_{34}$ | 9 | 30MB | 32GB |

Eqs. 4.4-4.9 suggest that experiment set $\pi_1$ includes experiments 1-3(i.e., $\beta_{11}$ - $\beta_{31}$), where the number of slave nodes is set to 0; the other experiments are grouped in the same manner. While conducting a set $\Pi$ of 12 experiments, we observe that the time spent in processing the first batch is approximately 23% less than that of the other batches. Thus, we have $A \approx 1.23$ when$n$ and $m$ are set to 4 and 3, respectively. $t_{ij}^{s}$ is the processing time of the $s$th batch in the $i$th experiment of the $j$th experiment set (see (eq.4.8)). For example, $t_{31}^{1}$ denotes the processing time of the fist batch in the third experiment of experiment set 1. Regardless of experiments, the average cold-start ratio between the first batch's processing time and those of the other batches is around 1.23. Thus, we have

$$\forall j \in n, \forall i \in m, s > 1 : \frac{t_{ij}^{1}}{t_{ij}^{s}} \approx 1.23. \tag{4.11}$$

$T_{ij}$ represents the total processing time of a multiple-node computing cluster, where $j$ is a constant in the $i$th experiment; $K$ and $k_i$ represents the total data size and batch size, respectively. Thus, we can re-write (4.7) as

$$T_{ij} \approx t_{ij}^{1} \ + \ \frac{t_{ij}^{1}}{1.23}(\frac{K}{k_i} - 1) = t_{ij}^{1} \ + \ \frac{t_{ij}^{1}}{1.23}(l_i - 1), \ l_i > 1. \tag{4.12}$$

Given a fixed dataset (i.e., $K$ = 300MB), the aforementioned model shows that the processing times of experiments $\beta_{11}$, $\beta_{21}$, and $\beta_{31}$ are:

- $T_{11} \approx t^1_{11} + \frac{t^1_{11}}{1.23}(l_1 - 1)$,
- $T_{21} \approx t^1_{21} + \frac{t^1_{21}}{1.23}(l_2 - 1)$, and
- $T_{31} \approx t^1_{31} + \frac{t^1_{31}}{1.23}(l_3 - 1)$.

To optimize the performance, we simply select the least processing time among above $T_{11}$, $T_{21}$, and $T_{31}$ for experiments $\beta_{11}$, $\beta_{21}$, and $\beta_{31}$. Since the overall processing time is reliant on the first batch's processing time, we have to compare $t^1_{11}$, $t^1_{21}$, and $t^1_{31}$ obtained from the three experiments (i.e., $\beta_{11}$, $\beta_{21}$, and $\beta_{31}$).

Recall that (see (4.6)) the number of batches is derived from the total data size and batch size. We compute the numbers of batches for $\beta_{11}$, $\beta_{21}$, and $\beta_{31}$ as 30, 15, and 10 (i.e., $l_1 = \frac{300}{10}$, $l_2 = \frac{300}{20}$, and $l_3 = \frac{300}{30}$), respectively.

Let us compare $T_{11}$ and $T_{21}$ obtained in experiments 1 and 2 (i.e., $\beta_{11}$ and $\beta_{21}$). We have

$T_{11} - T_{21} = t^1_{11}(1 + \frac{l_1 - 1}{1.23}) - t^1_{21}(1 + \frac{l_2 - 1}{1.23}) = 38.66 \times t^1_{11} - 19.18 \times t^1_{21}$.

When we change batch size from 10MB to 20MB, we observe that the first batch's processing time is increased by 309 seconds. Thus, we have $t^1_{11} = t^1_{21} - 309$.

$T_{11}$ for experiment 1 is less than $T_{21}$ if $t^1_{11}$ is less than 304.11 (i.e., $t^1_{11} < 304.11$ derived from $\frac{t^1_{11}}{t^1_{11}+309} < 0.496$).

Similarly, $T_{21}$ is less than $T_{11}$ if $t^1_{11}$ is larger than 304.11 (i.e., $t^1_{11} > 304.11$ derived from $\frac{t^1_{11}}{t^1_{11}+309} > 0.496$).

Now, we select the best scheme from experiment set $\{\beta_{11}, \beta_{21}, \beta_{31}\}$ to optimize the performance when the number of nodes is fixed to one. In addition, we need to compare the discrepancy among the experiments in set$\{\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}\}$ when we vary the number of nodes while keeping batch size unchanged. Because of the number of nodes in the cluster represents computing power, the computing cluster with more computing-nodes leads to less

processing time than clusters equipped with fewer nodes. In this example, the processing time of the first batch in experiment set$\{\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}\}$ should be sorted as: $t^1_{11} > t^1_{12} > t^1_{13} > t^1_{14}$. We expect that the average cold-start ratio is a constant regardless of system configuration. Unlike our expectation, increasing the cluster's computing capacity varies the average cold-start ratio. Therefore, we conclude that cold-start ratio largely depends on the number of nodes in a cluster.

### 4.3.3 Two-Stage Procedure to Optimize Batch Size

The above findings motivates us to propose a two-stage procedure (see Algorithms 1 and 2 below) to select an optimal batch size from a list of candidate batch sizes, thereby boosting performance of a given computing cluster.

**Algorithm 1:** Computing Cold-Start Ratio.

**Input :** Size of dataset $K$, batch size $k$
**Output:** average cold-start ratio for experiment set $\pi_j$

**for** *experiment* $\beta_{ij}$ *in experiment set* $\pi_j$ **do**
  number of batch $l_i = \frac{K}{k_i}$ ;
  **for** *the sth batch in* $l_i$ *of experiment* $\beta_{ij}$ **do**
    cumulated cold-start ratio $\alpha^s_{ij} += \frac{t^1_{ij}}{t^s_{ij}}$ ;
  **end**
  average cold-start ratio $A_{ij} = \frac{\alpha^s_{ij}-1}{s-1}$
**end**
**return** *average cold-start ratio for* $\pi_j$*:* $A_j = \frac{A_{ij}}{i}$

In Algorithm 1 (a.k.a., Stage 1), we pre-set a list of candidate batch sizes, one of which offers the best performance. Then, we calculate the number of batches by dividing the overall data size by a chosen batch size. Next, we cumulate cold-start ratios for all the batches including the first batch. Finally, Algorithm 1 outputs the average cold-start ratio of experiment set $\pi_j$. Average cold-start ratio $A_j$ is used to predict the processing time in experiments, which have different batch sizes under the same system settings. Therefore,

the proposed approach carries out Algorithm 2 (a.k.a., Stage 2) to estimate the minimal processing time.

**Algorithm 2:** Estimating Minimal Processing Time.

**Input :** Size of dataset $K$, batch size $k$, cold-start ratio $A_j$
**Output:** minimum consuming time $T_{min}$ in experiment set $\pi_j$

$T_{min} = 10{,}000$ ;
number of batch $l = \frac{K}{k}$ ;
**for** *experiment* $\beta_{ij}$ *in experiment set* $\pi_j$ **do**
    number of batch $l_i = \frac{K}{k_i}$ ;
    expected time consuming for experiment $\beta_{ij}$: $T_{ij} \approx t_{ij}^1 + \frac{t_{ij}^1}{A_j}(l_i - 1)$ ;
**end**
**if** $T_{ij} < T_{min}$*:* **then**
    $T_{min} = T_{ij}$ ;
**end**
**else**
    continue
**end**
**return** $T_{min}$

In the second stage, we apply the average cold-start ratio and the list of candidate batch sizes as inputs. Then, Eq. 4.11 is applied to predict the total processing time in each experiment. Finally, Algorithm 2 selects an optimal batch size that gives rise to the minimal processing time.

Recall that we design a model (see Section 4.3) to govern the election of the most appropriate batch size from a candidate list to optimize the efficiency of the proposed approach. Our model (see Section 4.2) coupled with the novel two-stage procedure (see Section 4.3.3) is adroit at optimizing main memory usage to speed up system performance for real time analytics. For more details about memory management within Spark, we refer the reader to the memory management and garbage collection procedures within Spark documentation.

Additionally, a few external techniques in the Spark Streaming framework judiciously optimize memory usage. These techniques include, for example, estimating memory consumption (i.e., SizeEstimator) and avoiding extra feature overhead to reduce memory consumption. In summary, we apply the Spark Streaming memory management methods such

as data serialization and memory tuning to effectively manage system's memory usage. We also designed a novel two-stage batch size selection procedure by calculating the average cold-start ratio to select the best batch size from a list of candidates to minimize processing time. In the next section, we demonstrate that given a system configuration, our proposed methodology is adept at optimizing the proposed method's performance.

In summary, this chapter introduces the development of the proposed RT-FEND system, which keeps using the design of FEND system described in Chapter 3. There are some of news components are involved: (1) topic merging mechanism that can merge highly similar topics; (2) Streaming data processing that allow the system to collect and pre-process data in real time manner; (3) two-stage procedure to optimize batch size of data. In other words, RT-FEND system optimizes FEND system in perspectives of data collection, data pre-processing, processing platform, and optimization of memory allocation.

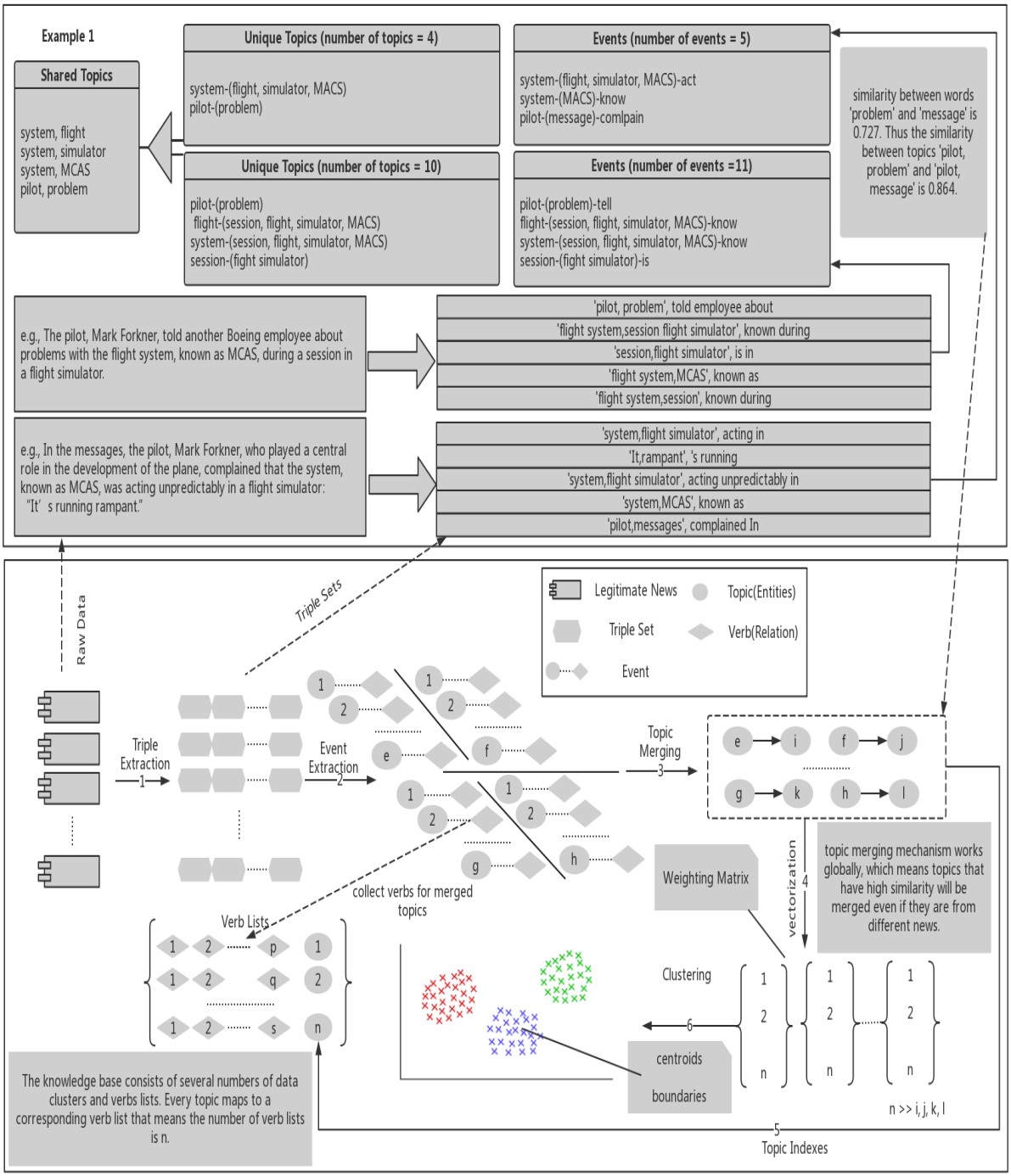



Figure 4.2: A example to explain the process of knowledge base construction.

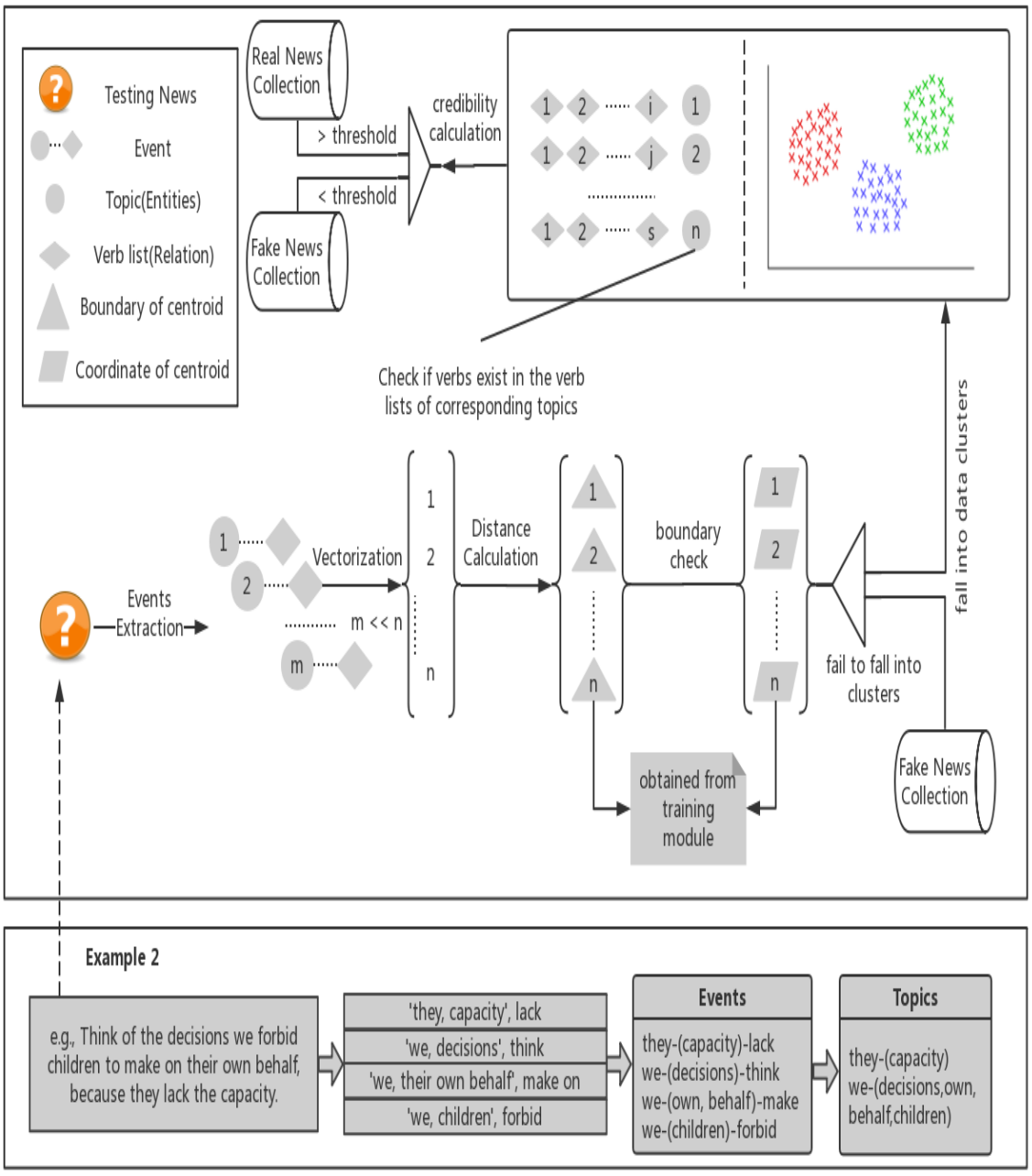



Figure 4.3: A example to explain the process of system testing.

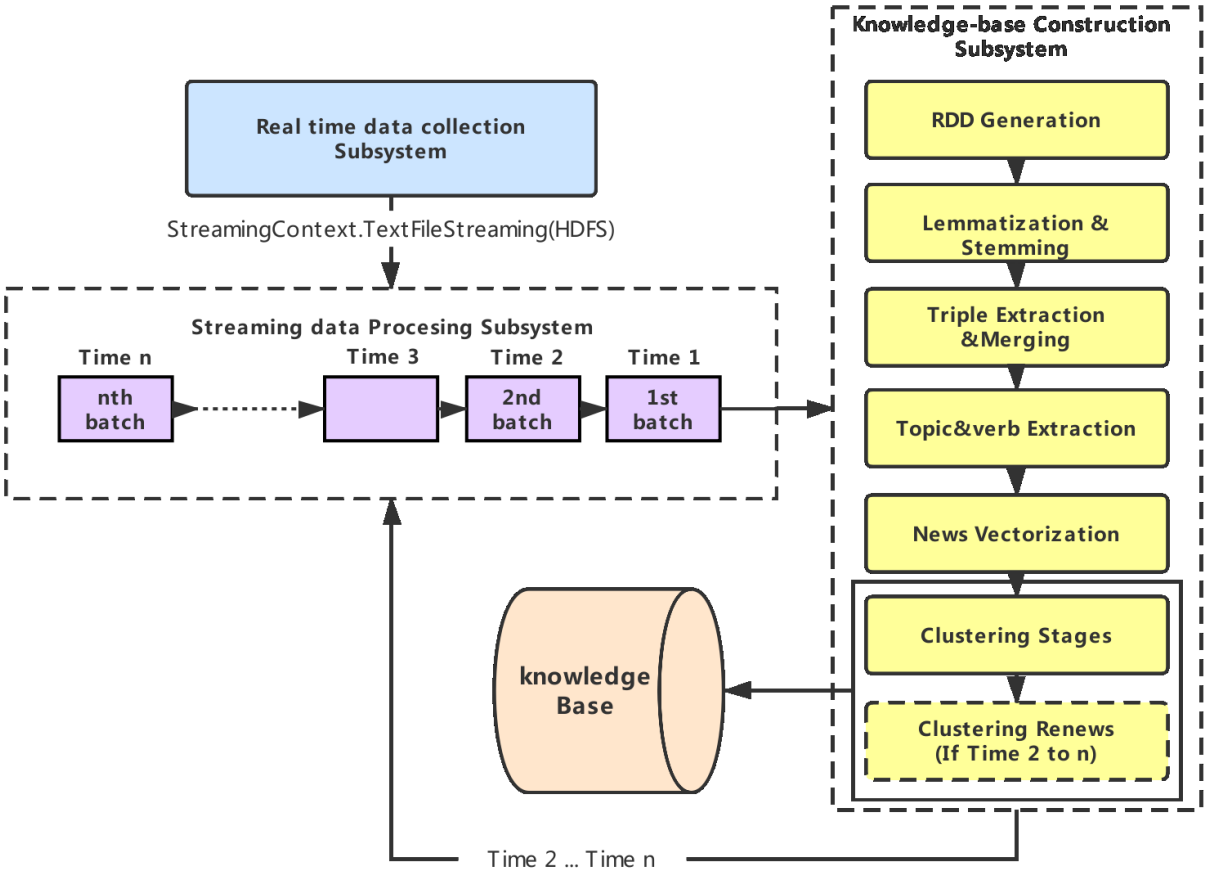



Figure 4.4: A system framework for knowledge-base constructions.

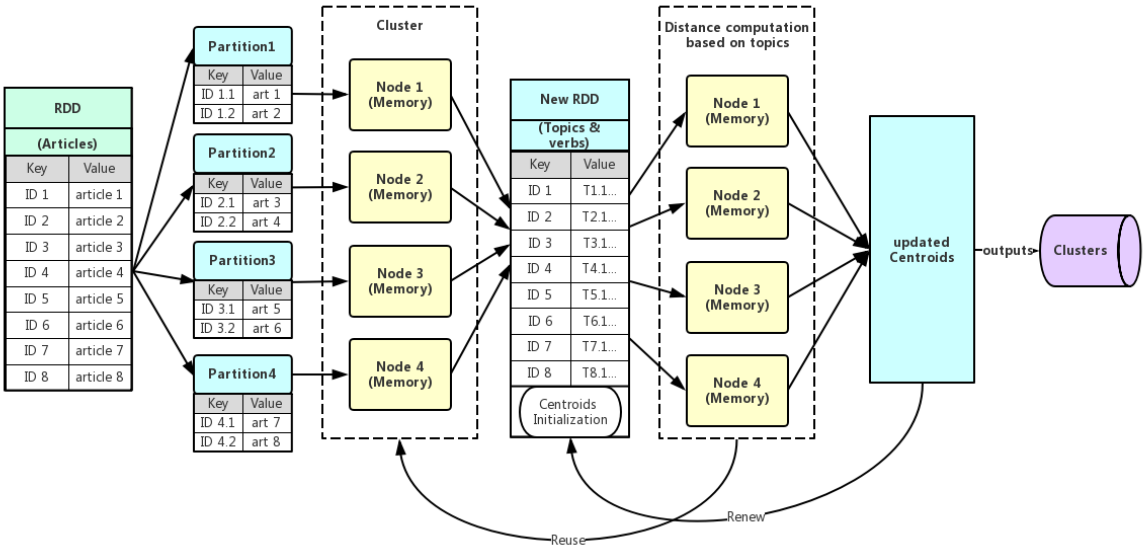



Figure 4.5: Data flow execution for updating knowledge base.

Chapter 5

Towards Identifying Objectivity and Subjectivity in Short Text

## 5.1 Feature Extractions and Expressions of OSS

In this section, we describe the systematic procedure for analyzing objective and subjective short texts at the sentence level. According to the event expression concept proposed by Zhang *et al.* [136], a sentence can be represented by one or more events, where each *event* is a triple set consisting of a subject, a predicate, and an object ($e_i = \{u_i, v_i, o_i\}$). Thus, we select subjects ($u$), predicates ($v$), and objects ($o$) of sentences as raw elements as a way to represent sentence-level documents. We subsequently apply the view point of a subject, tense of predicate, and the view point of an object as fundamental features to investigate the efficacy of these three parameters in identifying subjective and objective patterns in short texts. To accomplish this, we propose four new algorithms in this study. Section 5.2 presents algorithms 3 and 4 to detect viewpoints of subjects and objects, respectively. Section 5.3 presents Algorithm 5 for detecting tense of sentences. Finally, we design Algorithm 6 in Section 5.4 to elect objective and subjective patterns using outputs from algorithms 3, 4 and 5. We demonstrate the performance of these algorithms and our proposed approach using a set of threshold values.

We illustrate our proposed model, which manipulates these four algorithms to extract subjective and objective patterns by demonstrating a case study based on real data (See Fig. 5.1). We use three objective and three subjective sentences from Cornell Movie dataset to describe various sequences in the proposed model. The model first extracts relational triples from these six labeled sentences by applying *OpenIE*. Second, the algorithm 3 is applied to (1) pick the initial subject of sentences from each of the extracted triples; (2) confirm the view point of the selected subject. For example, the sentence *"I was perplexed*

*to watch it unfold with an astonishing lack of passion or uniqueness."* contains two triples: "*I, was perplexed, to watch it*", and "*It, unfold with, lack of passion*". The location of these two subjects - "*I*", and "*It*" are identified as *0*, and *5*, respectively with "*I*" being the initial subject of the sentence. In the next step, the model takes two inputs (1) the outputs from algorithm 1, and (2) relational triples to select the final subject-related object by applying algorithm 4. The output is the view point of the selected object. In the fourth step, the tenses of the sub-obj-related predicates is detected by applying algorithm 5. Finally, based on pattern distribution between the two datasets, extracted triple sets are elected as objective patterns and subjective patterns 6. In the figure, the pattern (3, 8, 0) is classified as an objective pattern because it doesn't occur in subjective samples. The pattern (4, 8, 0) occurs once in both objective and subjective samples leading to be classified as an uncertain pattern.

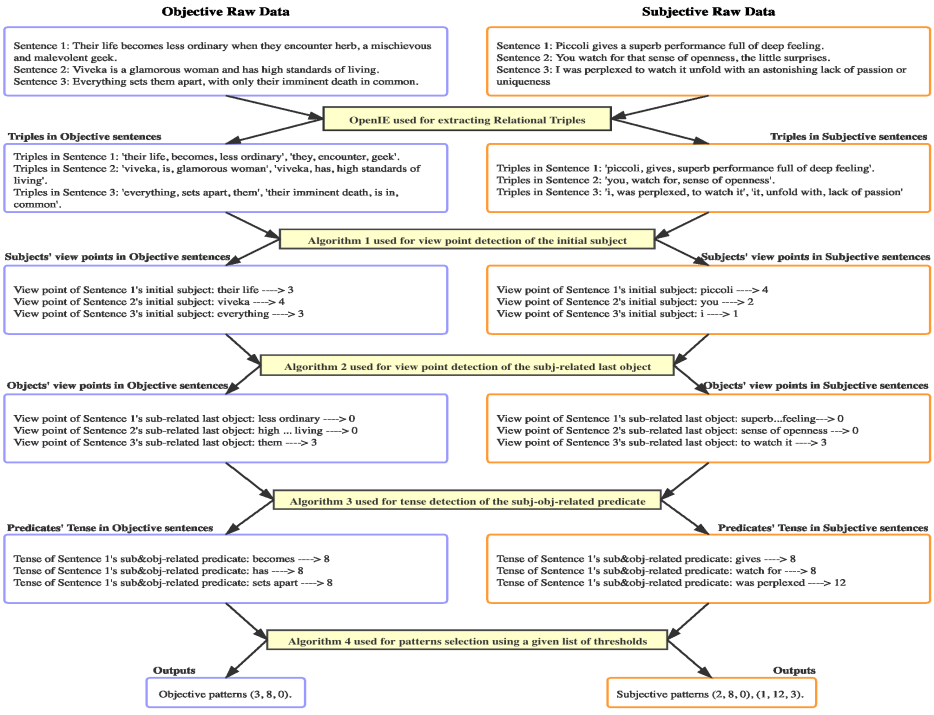



Figure 5.1: Sequential Steps in Proposed Model Demonstrated Using an Example

**Algorithm 3:** Confirming view point of subject

**Input :**
1. A sentence $s_i$,
2. Pre-defined pronouns lists $L_1, L_2, L_3$ for first, second, third view points

**Output:** selected subject and corresponding view point $(u_i, p_i^u)$

```
Event Exaction Function EE() ;
Name Entity Recognition Function NER() ;
all_sub_views = [] ;
for e in EE(s_i) do
    u_j = e[0] ;
    if len(set(u_j) & set(L_1)) != 0 then
        p_j^u = 1
    end
    if len(set(u_j) & set(L_2)) != 0 then
        p_j^u = 2
    end
    if len(set(u_j) & set(L_3)) != 0 then
        p_j^u = 3
    end
    if len [n for n in NER(u_j) n != 'O'] then
        p_j^u = 4
    end
    else
        p_j^u = 0
    end
    all_sub_views.append((u_j, p_j^u, s.index(u_j)))
end
min_u_i = min[i for u, v, i in all_sub_views] ;
uv = [u, v for u, v, i in all_sub_views if i == min_u_i] ;
u_i = uv[0] ;
p_i^u = uv[-1] ;
return subject and its view point (u_i, p_i^u)
```

## 5.2 View Point Detection

Generally speaking, sentences are classified as complete sentences and incomplete sentences. A complete sentence has to be constructed by at least three elements (i.e., a subject, a predicate, and an object). An incomplete sentence may have absences of any single or combination of three elements. The following examples depict some cases of incomplete sentences from the perspectives of three elements.

**Example 1: three examples about absences of three elements**

- *"Don't bother to answer".* [1]
- *"He was so pleased".*
- *"Not bad so far".*

In the first example, there is no subject and no object; the sentence only contains a predicate *"bother to answer"*. The second example, which lacks an object, only includes a subject *"He* along with a predicate *"was pleased"*. The third case only contains a predicate "bad" with an auxiliary verb (e.g., am, is) being absent. In our approach, we set the viewpoint value of a sentence to 0 if the sentence has any missing element. For example, in *Sentence 1*, the values of subject and object are set to $p_i^u = 0$ and $p_i^o = 0$, where $p_i^u$ and $p_i^o$ denote the view points of subject $u$ and object $o$, respectively.

To cover all possible subject- and object-related cases of sentences, we introduce a set $P = \{0, 1, 2, 3, 4\}$, where first view points, second view points, third view points, and the other cases using non-pronouns are represented as 1, 2, 3, and 4, respectively. Value 0 in set $P$ indicates that no subject or object is identified from input sentences. Suppose sentence $s_i$ is expressed as a triple set $\{u_i, v_i, o_i\}$, where $u_i$ denotes a subject, $v_i$ represents a predicate, and $o_i$ is an object. Function $F_{s\_view}$ returns the view point $p_i^s$ of subject $u_i$, and Function $F_{o\_view}$ outputs the view point $p_i^o$ of object $o_i$. Both $p_i^u$ and $p_i^o$ must belong to value set $P$. More formally, we have

[1] A fragment of a sentence in the Cornell movie review subjective dataset.

$$(u_i, p_i^u) = F_{s\_view}(s_i, L_1, L_2, L_3), \text{where} \quad p_i^u \in P = \{0, 1, 2, 3\} \tag{5.1}$$

$$(o_i, p_i^o) = F_{o\_view}(s_i, L_1, L_2, L_3, (u_i, p_i^u)), \text{where} \quad p_i^o \in P = \{0, 1, 2, 3\} \tag{5.2}$$

where $L_1$, $L_2$, and $L_3$ are manually collected words vocabularies that are comprised of all pronouns in first view points, second view points, and third view points, respectively.

If neither a subject nor an object of sentence $s_i$ is initially discovered, then $p_i^s$ or $p_i^o$ is tentatively set to 0. when sentence $s_i$ embraces a subject (an object), the value of $p_i^u$ ($p_i^o$) should be set to 1, 2, 3, or 4. The purpose of view-point detection motivates us to devise the following algorithms 3 and 4 to automatically confirm the view points of subjects and objects. These algorithms are underpinnings of the process of identifying objective and subjective statements.

Algorithm 3 detects subject view points. This algorithm carries out the following four steps to determine view points of an subject in a given sentence. First, an event extraction function $EE()$ is applied based on *Stanford OpenIE* [69] to extract *events* from sentences, where event is a triple set that consists of a subject, a predicate, and an object (i.e., $e = \{u, v, o\}$). A name entity recognition function $NER()$ provided by *Spacy* [53] is deployed to detect if there exists any name entities in subjects of events (Line 1-2). Second, the algorithm checks the view point of pronoun-based subject of all the events (Line 6-16). Third, function $NER()$ is invoked to recognize any non-pronoun name-entities used in subjects (Line 15-16). Finally, the first initial subject of the sentence is selected to represent the subject of the input sentence (Line 23-27), and the chosen subject coupled with its view point are ejected by the algorithm.

Algorithm 4, which is similar to Algorithm 3, is in charge of detecting object view points. Algorithm 4 not only utilizes all the parameters of Algorithm 4, but also picks the

**Algorithm 4:** Confirming view point of object

**Input :**

1. A sentence $s_i$,
2. Pre-defined pronouns lists $L_1, L_2, L_3$ for first, second, third view points
3. $(u_i, p_i^u)$ from algorithm 3

**Output:**

view points of subject and object ($p_i^u$, $p_i^o$)

```
n = len(S) ;
Event Exaction Function EE() ;
Name Entity Recognition Function NER() ;
all_obj_views = [] ;
for e in EE(s_i) do
    o_j = e[2] ;
    if u_i == e[0] then
        if len(set(o_j) & set(L_1)) != 0 then
            p_j^o = 1
        end
        if len(set(o_j) & set(L_2)) != 0 then
            p_j^o = 2
        end
        if len(set(o_j) & set(L_3)) != 0 then
            p_j^o = 3
        end
        if len [n for n in NER(o_j) n != 'O'] then
            p_j^o = 4
        end
        else
            p_j^o = 0
        end
    end
    else
        Continue
    end
    all_obj_views.append((o_i, p_i^o, s.index(o_j)))
end
max_o_i = max[i for o, v, i in all_obj_views] ;
ov = [o, v for o, v, i in all_obj_views if i == max_o_i] ;
o_i = ov[0] ;
p_i^o = ov[-1] ;
return object and its view points (o_i, p_i^o)
```

outputs of algorithm 3 as an input. This algorithm consists of the four steps, among which the first three steps detect view points of input-subject-related objects ($p_j^o \in \{0, 1, 2, 3, 4\}$) (Line 6-26). As a final step, the algorithm elects the last input-subject-related object as the object of the sentence; the selected object accompanied by corresponding view point become algorithm's outputs.

### 5.3 Tense Detection

After confirming view points of subject $p_i^u$ and object $p_i^o$, we propose algorithm 5 to figure out sentence's tense - a vital information for extracting objective and subjective patterns. Before designing the algorithm, we extend three basic tenses - 'future', 'present', and 'past' - into twelve tenses using three tense-based parameters - 'Perfect', 'Continuous', and 'Continuous Perfect'. Such twelve tenses cover all the sentence cases summarized in Table 5.1 accompanied with the POS tagging schemes.

Let $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 0\}$ denote a set of twelve tenses containing future perfect continuous tense (i.e., 1), future perfect tense (i.e., 2), future continuous tense (i.e., 3), future simple tense (i.e., 4), present perfect continuous tense (i.e., 5), present perfect tense (i.e., 6), present continuous tense (i.e., 7), present simple tense (i.e., 8), past perfect continuous tense (i.e., 9), past perfect tense (i.e., 10), past continuous tense (i.e., 11), past simple tense (i.e., 12), and undetected tense (i.e., 0). We devise function $E_{tense}$ (see also algorithm 5), which takes (1) all the input parameters of algorithm 3, and (2) outputs of algorithms 3 4 - ($u_i$, $p_i^u$), and ($o_i$, $p_i^o$) as input parameters. Function $E_{tense}$ returns tense $q_i$ of the chosen event (i.e., $(u_i, v_i, o_i)$). Intuitively, the value of $q_i$ is an element in set $Q$. More specifically, we have

$$\begin{aligned} q_i &= E_{tense}(s_i, u_i, p_i^u, o_i, p_i^o), \text{where} \\ q_i \in Q &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 0\}. \end{aligned} \tag{5.3}$$

We implement algorithm 5 to effectively detect tense of an input sentence. In the algorithm, We employ CoreNLP POS Tagger (i.e., $CPT()$) approach [69], which is nested in the Stanford CoreNLP toolkit to identify part of speech of words (Line 2). Next, we collect all the twelve tense schemes in a list (i.e., *Scheme*) in an order of the labels in table 5.1 (Line 3). Then, the program searches matched predicate using the outputs of Algorithm 3 and Algorithm 4. Finally, if predicate $v_i$ has a scheme within the scheme list, the tense of $q_i$ will be returned according to the index of scheme list; otherwise, $q_i$ is set to 0 meaning that no tense is diagnosed in the case.

**Algorithm 5:** Detecting Tenses of Sentences

**Input :**
1. A sentence $s_i$,
2. $(u_i, p_i^u)$ from algorithm 3,
3. $(o_i, p_i^o)$ from algorithm 4

**Output:** The tense $q_i$

```
1  Event Exaction Function EE() ;
2  CoreNLP POS Tagger CPT() ;
3  A list of all tense schemes Scheme ;
4  all_matched_tense = [] ;
5  for e in EE(s_i) do
6  |   if u_i == e[0] and o_i == e[2] then
7  |   |   v_i = e[1] ;
8  |   |   if set(CPT(v_i)) & set(Scheme) != 0 then
9  |   |   |   matched_scheme = set(CPT(v_i)) & set(Scheme) ;
10 |   |   |   matched_tense = Scheme.index(matched_scheme) ;
11 |   |   |   all_matched_tense.append(matched_tense)
12 |   |   end
13 |   end
14 end
15 if len(all_matched_tense) != 0 then
16 |   q_i = min(all_matched_tense) ;
17 end
18 else
19 |   q_i = 0
20 end
21 return (v_i, q_i)
```

## 5.4 Objective and Subjective Patterns Extractions

Algorithm 6 shows the process of electing subjective patterns and objective patterns governed by a threshold $\varphi$. First of all, we make use of the outputs of algorithm 3, algorithm 4, and algorithm 5 to construct triples in form of $(p_i^s, q_i, p_i^o)$, where $p_i^s$ represents the view point of sentence $i$'s subject, $q_i$ denotes the tense of sentence $i$, and $p_i^o$ is the view point of sentence $i$'s object. The triples are fed into algorithm 6 as an input. A given list of thresholds $\varphi \in \{70.00\%, 72.50\%, 75.00\%, 77.50\%, 80.00\%, 82.50\%,$
$85.00\%, 87.50\%, 90.00\%, 92.50\%\}$ are applied to repeatedly go through the entire algorithm 6 to evaluate the effectiveness of different thresholds on electing two types of patterns. The algorithm first counts which triples appeared in the two datasets, followed by tracking the number of each triple for the datasets separately (See line 5, 6). Then, the union set ($all\_triples$) of the two datasets are used to traverse the number of each triple in the objective and subjective dataset for possibility calculations (See line 7). If the numbers of a triple in the subjective dataset and objective dataset are zero and non-zero, the current triple will be collected as an objective pattern, and vice versa (See line 8-12). If the numbers of a triple are non-zero in both datasets, the ratio of the triple (See line 15) will be calculated to compare with a given threshold $\varphi$. If the ratio is larger than $\varphi$, the triple will be classified as a subjective pattern (See line 16-17); otherwise, the triple will be treated as an objective one if the ratio is smaller than $\frac{1}{\varphi}$ (See line 19-20).

This chapter summarizes the structure of our proposed objective and subjective separation approach. We tried to investigate the insights of three components of the external event of short sentences: viewpoints of subjects and objects, and tense of predicates in determining subjective sentences and objective sentences. Meantime, we design four algorithms to address this purpose. The first two algorithms are designed to detect viewpoints of subjects and objects.The tense of sentences can be detected by using the third algorithm. Finally, the results of the three algorithms are fed to the last one to extract subjective and objective patterns of sentences.

**Algorithm 6:** Capturing subjective patterns and objective patterns.

**Input :** objective and subjective datasets $SV, OV$ in forms of triple set $(p_i^s, q_i, p_i^o)$, Threshold $\varphi$

**Output:** Subjective patterns set $SP$, objective patterns set $OP$

```
count_SV = Counter(SV).items() ;
count_SV = Counter(OV).items() ;
all_triples = SV.union(OV);
for triple in all_triples do
    num_triple_S = [item_num for item, item_num in count_SV if item == triple] ;
    num_triple_O = [item_num for item, item_num in count_OV if item == triple] ;
    num_triple_O = num_triple_S[0] == 0 ;
    if num_triple_S[0] == 0 and num_triple_O[0] !=0: then
        OP.append(triple)
    end
    if num_triple_S[0] != 0 and num_triple_O[0] == 0: then
        SP.append(triple)
    end
    else
        ratio = num_triple_S[0] / (num_triple_S[0]+num_triple_O[0]) ;
        if ratio ≥ φ then
            SP.append((triple,ratio))
        end
        if ratio ≤ 1/φ then
            OP.append((triple,ratio))
        end
    end
end
return SP, OP
```

Table 5.1: All twelve tenses accompanied with corresponding POS tagging schemes and labels

| **Tenses** | **POS Schemes** | **Example** | **labels** |
|---|---|---|---|
| Future Perfect Continuous | "MD" + "VB" + "VBN" + "VBG" | I will have been doing | 1 |
| Future Perfect | "MD" + "VB" + "VBN" | I will have done | 2 |
| Future Continuous | "MD" + "VB" + "VBG" | I will be doing | 3 |
| Future Simple | "MD" | I will do | 4 |
| Present Perfect Continuous | "VBD" + "VBN" + "VBG" | I have been doing | 5 |
| Present Perfect | "VBD" + "VBN" + "VBG" | I have done | 6 |
| Present Continuous | "VBD" + "VBN" | I am doing | 7 |
| Present Simple | "VBD" | I do | 8 |
| Past Perfect Continuous | "VBP"/"VBZ" + "VBN" + "VBG" | I had been doing | 9 |
| Past Perfect | "VBP"/"VBZ" + "VBN" | I had done | 10 |
| Past Continuous | "VBP"/"VBZ" + "VBG" | I was doing | 11 |
| Past Simple | "VBP"/"VBZ" | I did | 12 |

# Chapter 6

# Preliminary Results and Discussions

## 6.1 Results of FEFND

In this section, we discuss preliminary results collected in the experiments conducted to quantitatively demonstrate the performance strengths of *FEND* using two real-world news datasets. To carry out performance evaluation, we compare our approach with two existing data-mining algorithms with respect of accuracy, precision, recall rate, and *F-score*.

### 6.1.1 Datasets

To quantitatively evaluate the effectiveness of our system, we make use of legitimate news available from *CNN* and *Ne+w York Times* for model training. Detection accuracy is assessed using various *test datasets*, which are summarized in Table 6.1. In this study, our news resources such as CNN and New York Times are regarded as legitimate based on the classification done by a large scientist group. They also have wide circulation among larger audience compared with their counterparts [76, 15, 82]. For example, CNN is accessed by approximately 96 million pay-television households, representing 82.8% of households with at least one television set in the U.S. [116]. A few viewers may not treat CNN as a legitimate resource because CNN's content is inconsistent with the viewers' opinions. Nonetheless, our system contains a flexible mechanism that allows users to plugin any legitimate news database. Users may choose their trustworthy news outlets for constructing a legitimate news database. Regardless of news database, our system is adept at detecting fake news.

We explore a group of websites classified as fake news domains [50] to acquire fake news as ground truth. These websites include, but are not limited to, *www.greenvillegazette.com*, *www.politicops.com*, *www.advocate.com*, and *www.naturalnews.com*. The first

Table 6.1: Fake and legitimate news article counts from the website sources.

| Type | realNews | fakeNews |
|---|---|---|
| CNN | 8897 | 0 |
| New York Times | 5334 | 0 |
| advocate | 0 | 6444 |
| naturalnews | 0 | 2402 |
| politicops | 0 | 3066 |
| greenvillegazette | 0 | 1525 |

three websites publish Pro-Clinton fake news, whereas the last two websites circulates Pro-Trump fake news. News acquired from both Pro-Clinton and Pro-Trump websites drive our unbiased and fair comparison experiments. Note that *Natural News* is a website gaining its popularity from scientific fake news and various conspiracy theories.

Table 6.2: Article cluster collection for ground truth

| Clusters No. | Subjects (selected) | Number of topics (repeated topics included) |
|---|---|---|
| 1 | foreign, residents, workings, leaders, stocks, obama, governments, groups, americans | 29877 |
| 2 | president, country, policy, children, family, University, government, administration, court, republicans, clinton, immigration | 25318 |
| 3 | report, Washington, Facebook, criticism, investigation, department, circumstances, organization | 24476 |
| 4 | weapon, chief, authorities, surveillance, peace, contest, Penitentiary, corrections, robbery, violent, surveillance, prisoner, murder | 20386 |
| 5 | ISIS, adolescent, politico, warning, evil, oceans, deliberation, insult, opponents, correctness, slaughter, panelists, apparatus | 19973 |
| 6 | media, trump, campaign, democracy, california, politician, verdict, senate, cnn, nominee, truth, victims | 14323 |
| 7 | environment, mediterranean, photographs, hospitals, mountaineers, earthquake, migrant, artists, villages, sight | 12033 |
| 8 | devices, databases, company, amazon, business, market, website, engineer, value, worldwide, Google, competitors | 10879 |
| 9 | culture, story, police, director, space, blackness, challenge, popularity, experiences, expectations, deficit, motivation, exhibition | 9894 |
| 10 | ambassador, north, korea, speech, quarantine, surveillance, crisis, policy, conflict, meeting, intelligence, reconnaissance | 9435 |
| 11 | entrepreneurs, business, startups, guests, experts, incubator, university, recruits, questions, authorities, investigation, photographs, activities | 8997 |
| 12 | refugees, country, western, violence, police, asylum, extremists, Federal, arrests, Pakistan, psychologist, pursuit | 6962 |
| 13 | hurricane, Miami, emergency, crosshairs, residents, carolina, economy, Alabama, rumors | 6795 |
| 14 | gravel, road, accident, license, professor, law, convictions, prosecutors, evidence, instructions, investigators, justice | 6580 |
| 15 | secretary, military, implementation, tweet, country, defence, fighters, sailors, patriot, freedom, opportunity, democracy, oppression | 6439 |
| 16 | biases, whites, neighborhoods, relationships, foundation, segregation, marriages, minority, economist, associations, laboratory, interactions | 5971 |
| 17 | reporters, facebook, survivor, community, firefighter, Twitter, evidence, winners, deputy, photographer, attorney, warrants | 5881 |
| 18 | olympics, medal, coach, ceremony, champions, reporters, committee, athletes, recognition | 5601 |
| 19 | sugar, calories, health, vegetables, nutrients, disease, attraction, breaks, textures, deprivation, antioxidants, balance | 5033 |
| 20 | actress, theory, nominations, film, tribute, portrayal, rewards, felicity, spirit, characters, actions | 4987 |

### 6.1.2 Evaluation

As described earlier, we extract news topics as features using two different clustering techniques – K-means and Affinity Propagation (AP) – for purposes. Both algorithms produced similar number of clusters. However, the composition of the clusters produced using different techniques could have small variation. Both of these approaches apply the euclidean distance as the measure for classifying the data into different clusters. For our dataset, both algorithms produced similar results. We, therefore, focus on clusters produced using AP technique.

Table 6.2 illustrates selected subjects of 20 clusters that were identified, along with the total number of topics in each cluster. Cluster 1 (see Table 6.2) is the largest cluster with approx. 29,900 topics and includes subjects such as "foreign", "residents", "workings", "leaders", "stocks", "Obama", "governments", "groups", "Americans", etc. Each of these subjects are associated with various verbs, which make it easier for classifying the articles. On the other end, cluster 20 is the smallest cluster with approx. 5000 topics and includes subjects such as "actress", "theory", "nominations", "film", "tribute", "portrayal", "rewards", "felicity", "spirit", "characters", "actions", etc.

Table 6.3: The three topic sub-groups created by the news-topic clustering algorithms.

| **Topic Zones** | **Topics** |
|---|---|
| 1. Red | Foreign - ( representative, KCNA, media, meet, Moscow, Monday, state )..<br>Residents - ( area, wave, roadsCity, employees, equipment, advance, storm, clock, system, Coast, East, West, roadsCity )..<br>Workings - ( Putin, Vladimir, Russia, Hezbollah, position, border, strike, weeks, Israeli ).. |
| 2. Green | Leaders - ( gains, policy, children, program, skills, doctrine, grade, success )..<br>Money - ( energy, time, legislation, year )..<br>Stocks - ( rotations, brigade, armor, intervals, Europe, United, States )..<br>dogs - ( Europe, Asia, wolves, Stone, Age, Old ).. |
| 3. Blue | Obama - ( victory, increase, tax, research, funding, wake, rate, debate, year, Brooke )..<br>Governments - (predecessor, President, Moon, scandal, corruption, arrest, power, nothing)..<br>Groups - ( terror, group, Obama, administration, US, air, campaign, stretch )..<br>Americans - ( years, age, hypertension, part, work, health )..<br>journalists - ( figuring, tally, rights, culture, war, crime, business ) |

Figure 6.1 illustrates a network visualization of all the topics belonging to cluster 1, which is the largest cluster. All the topics belonging to the cluster could be grouped into

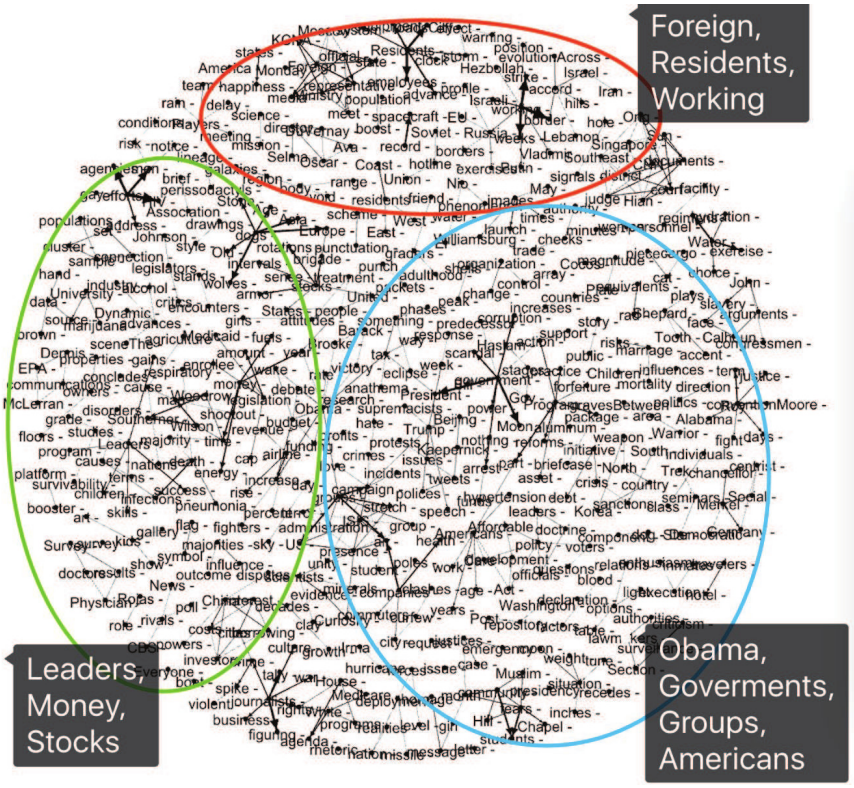



Figure 6.1: Network-based topic visualization

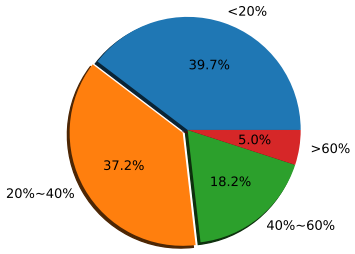
three sub-groups or topic zones. Bolder arrows represent topics occurring with high frequency. In Table 6.3, we extract almost all of the topics with high frequencies to show the relationship among topics and categorize them based on their sub-groups. For example, "foreign" (representative, KCNA, media, meet, Moscow, Monday, state) identifies one topic (subject-object pair) within the subgroup. Each topic within cluster 1 has relation to the other topics. These results suggest that a particular topic may appear in one article, but may also be identified across multiple similar articles. Our approach performs ingrained fake news classification using two layers of sequential filtering. The first layer of filtering uses the clustering results from the training model to detect the fake news based on any existence of fake topics. The second layer of filtering screens the fake news based on event extraction and credibility evaluation scores. Table 6.4 summarizes the results after applying the first layer of filtering that is based on topic extraction.

Table 6.4: The number of fake news detected by the first filter.

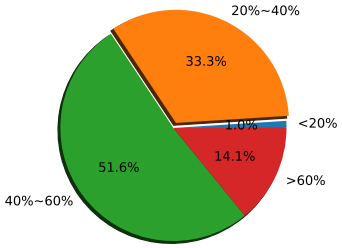
| | advocate | naturalnews | politicops | greenvillegazette |
|---|---|---|---|---|
| Uncertain News | 6444 | 2402 | 3066 | 1525 |
| Fake Topics | 4312 | 506 | 133 | 478 |
| Remaining Data | 2132 | 1896 | 2933 | 1047 |

Table 6.5: Credibility Score Descriptive Statistics for 'Remaining' Fake news.

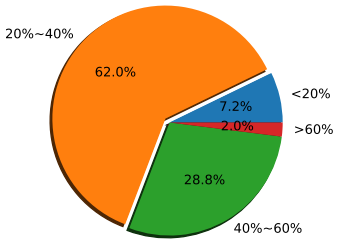
| **Fake News** | Mean | **Range** | **Std. Deviation** |
|---|---|---|---|
| greenvillegazette | 0.33 | (0.0, 0.88) | 0.178 |
| politicops | 0.345 | (0.1, 0.85) | 0.155 |
| naturalnews | 0.366 | (0.0, 0.67) | 0.15 |
| Advocate | 0.36 | (0.0, 1.0) | 0.156 |



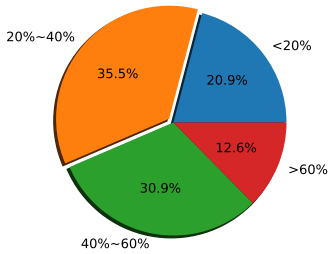

(a) advocate




(b) naturalnews




(c) politicops




(d) greenvillegazette

Figure 6.2: the Credibility Distribution of News of all Four Fake News Datasets

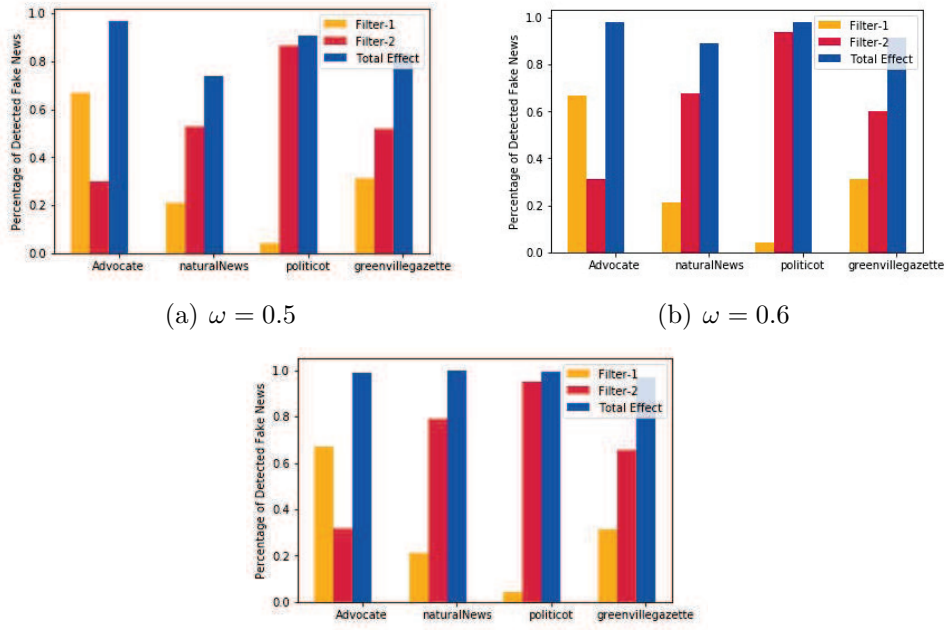
Among all the fake news summarized in Table 6.4, 66.9% news in `advocate.com`, 21.1% news in `naturalnews.com`, 31.4% news in `greenvillegazette.com`, and 4.4% news in `politicot.com` are detected as type-1 fake news by the first-level filter. Results suggest that the detection rate of the first-level filter varies significantly across different datasets and shows a wide detection range. For example, the detection rate is the highest for the `advocate.com`

dataset (i.e., 66.9%) and the lowest for the politicot.com dataset (i.e., 4.4%). Over 65% news in advocate.com are correctly detected by the first-level filter, meaning that a large portion of news posted in advocate.com have fake topics (i.e., type-1 fake news). The type-1 fake news detection rates for naturalnews.com and greenvillegazette.com are medium low (i.e., 21.1% and 31.4%), indicating that a majority of news published on naturalnews.com and greenvillegazette.com have legitimate topics. For the news found on politicot.com, only 4.4% are detected as type-1 fake news, suggesting that almost of all the politicot.com news contain trustworthy topics. Results suggest that first filtering layer provides a good mechanism for detecting the reliability of a news source. The output of the first layer feeds to the second layer filter.

The second-level filter detects the credibility of fake news and facilitates the comparison of the individual fake news scores with the threshold $\omega$. Table 6.5 provides descriptive statistics of remaining unclassified fake news documents. The average credibility of the fake news is found to have a mean of 0.35, average range of $(0.025, 0.85)$ and average standard deviation of 0.16.

Figure 6.2 describes the distribution of credibility scores for each of the four fake news datasets. Each of fake new datasets were found to contain a relatively small proportion of high credibility articles (real news) legitimate articles. For example, advocate dataset contains 5% legitimate news having over 60% credibility, while 77% of news' have credibility lower than 40%. Naturalnews has 14% legitimate news with credibility over 60%, 86% with less than 60% credibility, 34% with less than 40% credibility, and 52% news with credibility between 40% and 60%. politicot is found to have only 2% news articles with over 60% credibility. Greenvillegazette has more dispersed distribution with 12.4% news articles demonstrating 60% credibility scores. The distribution patterns of fake news suggest that advocate and politicot could be mostly classified as fake news source with fewer than 5% and 2% real news articles respectively. On the other hand, naturalnews and greenvillegazetee has 12-14% real news articles but a more distributed credibility news score. Such websites

could be considered as having tendencies for generating fake news content. Figure 7 shows the performance of two filtering approaches separately as well as collectively. With $\omega = 0.6$, the overall fake news detection performance ranges between 90-97% whereas with $\omega = 0.7$, the overall fake news detection reaches an overall average of 97%.



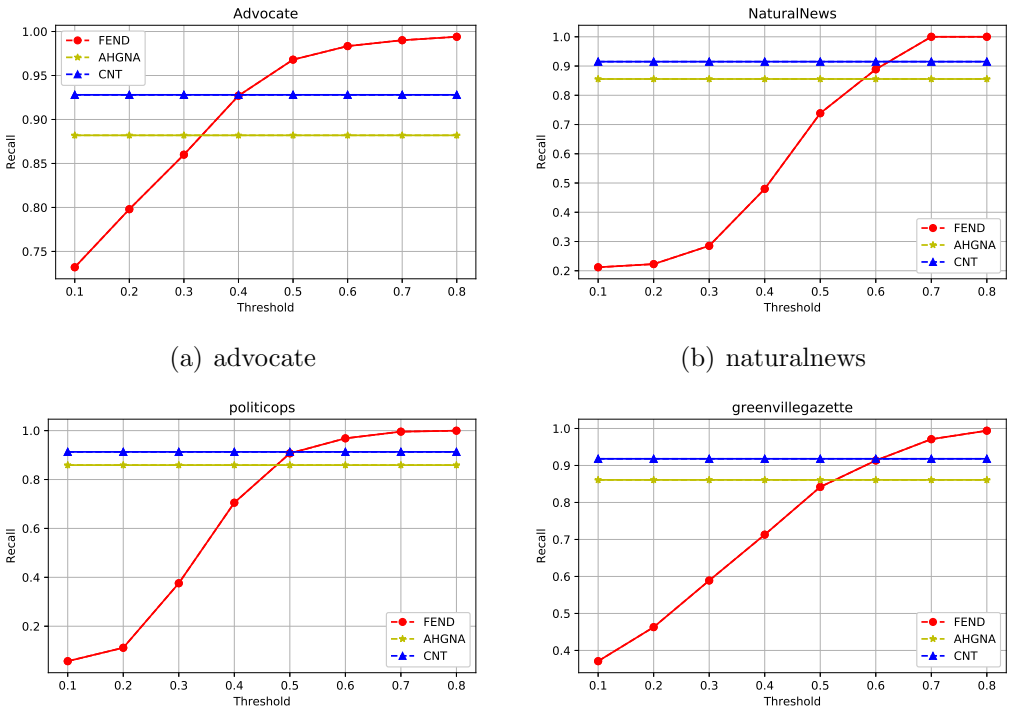

(a) $\omega = 0.5$ (b) $\omega = 0.6$

(c) $\omega = 0.7$

Figure 6.3: The performance of two filters in the model

Next, we carry out four experiments following the three steps. (1) We select 75% news articles (i.e., 10,674 CNN and New York Times news) from the source as training data. (2) We build a model using the training data. (3) The remaining 25% news perform as testing data (i.e., 3,557 news) to validate the model constructed in step 2. The purpose of this cross-validation procedure to evaluate our system's capability of estimating credibility scores of various testing datasets.

Each of the four folds of the testing dataset is populated with the random assignment of news articles drawn from real as well as fake news. Table 6.6 shows four cross validation data sets.




(a) advocate

(b) naturalnews

(c) politicops

(d) greenvillegazette

Figure 6.4: Performance Comparisons among CNT, AHGNA and proposed *FEND* approach based on fake news datasets.

Table 6.6: Cross Validation Dataset

| **Cross Validation Fold** | **Corresponding Fake News Dataset** | **RealNews** | **FakeNews** |
|---|---|---|---|
| 1 | advocate | 3557 | 6444 |
| 2 | naturalnews | 3557 | 2402 |
| 3 | politicops | 3558 | 3066 |
| 4 | greenvillegazette | 3557 | 1525 |

Table 6.7 shows the result of cross validation for mixed datasets. The accuracy ranges between 93.77% and 89.55%, precision ranges between 92.77% and 83.13%, recall rate ranges between 97.91% and 92.42%, and F-score ranges between 94.91% and 88.31%. Based on the accuracy and recall definitions from equations 3.18 and 3.20, recall values for pure fake news and mixed datasets are equal to the the accuracy values of pure fake news datasets.

We further investigate the recall and accuracy of the proposed approach by comparing its efficacy with two other approaches. The first approach is referred to as CNT, a multi-feature detection technique that is based on detecting features: content-based features (such

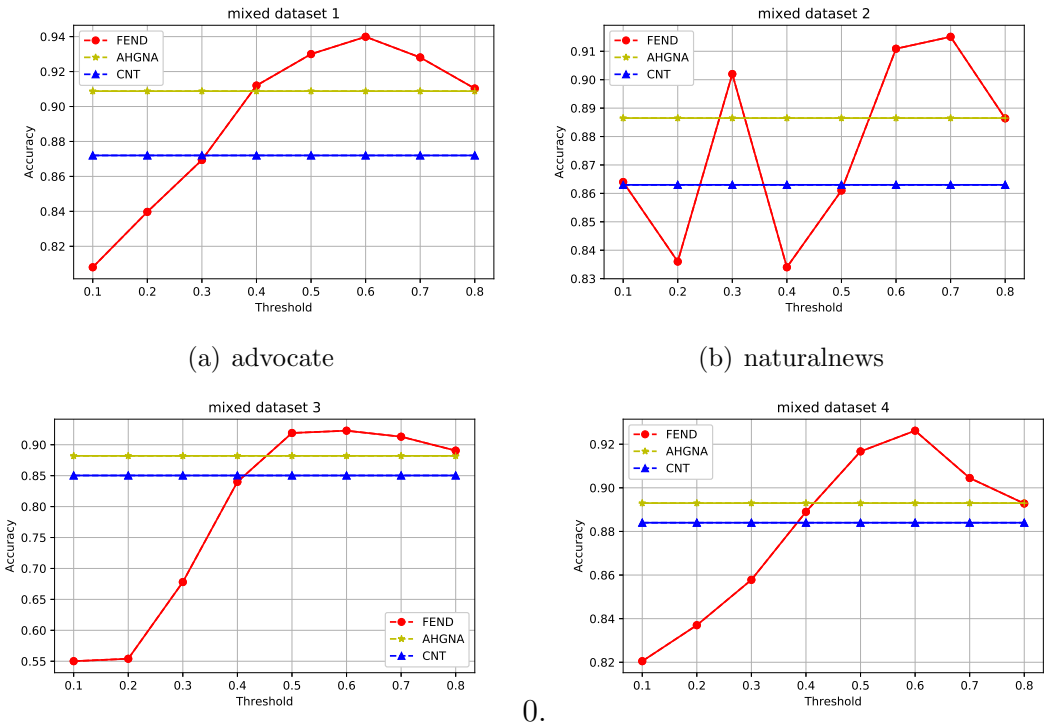
Table 6.7: Performance Evaluation( use $\omega = 0.6$ as the default threshold for mixed dataset)

| Cross Validation Fold | Accuracy | Precision | Recall | F-score |
|---|---|---|---|---|
| 1(advocate) | 93.99% | 92.77% | 98.34% | 95.47% |
| 2(naturalNews) | 91.09% | 89.00% | 88.88% | 88.94% |
| 3(politicops) | 92.27% | 86.92% | 98.08% | 92.16% |
| 4(greenvillegazette) | 92.62% | 85.13% | 91.34% | 88.12% |

as lexical patterns, part-of-speech pattern), network-based features (e.g. user behavior) and twitter specific memes (e.g. hashtag) [94]. The second is AHGNA, which uses Absurdity, Humor, Grammar, Negative, and Affect as features for segregating fake or satire from the real [104]. Figure 6.4 shows the comparison of three approaches based on the recall measure. For higher values of $\omega$, the system built on the proposed approached outperforms CNT and AHGNA approaches. For example, the recall of *FEND* exceeds CNT and AHGNA for $\omega$ higher than 0.4 for fake news originating from advocate. Similarly, *FEND* performed better for $\omega$ exceeding 0.52 while for politicot news. Finally, for $\omega$ greater than 0.6, *FEND* worked better for naturalnews and politicot. Additionally, we found the singled-tailed t-test to be significant at 0.05 as well as 0.01 significance level assuming homoscedasticity. For FEND vs. CNT, p-value is 0.00278 and 0.00015 for FEND vs. AHGNA.

Figure 6.5 shows the accuracy comparisons between the proposed method and two baseline methods (i.e., CNT and AHGNA) using the mixed datasets. For a high $\omega$ value (e.g., 0.6), our proposed approach outperforms both CNT and AHGNA. For example, Figure 6.5(a) reveals that the accuracy of FEND exceeds CNT and AHGNA when the threshold $\omega$ is higher than 0.4. More often than not, the threshold in real-world scenarios is configured in a window between 0.5 and 0.8.

Figure 6.6 demonstrates the *F-score* comparisons among FEND, AHGNA, and CNT when threshold $\omega$ is configured to 0.6 in the case of four mixed datasets. Figure 6.6 illustrates that FEND outperforms AHGNA and CNT in terms of the *F-score* measures. For example, FEND improves AHGNA's *F-score* by up to 4.6% with an average of 2.4%; FEND enhances the CNT's *F-score* by up to 6.2% with an average of 3.3%.




(a) advocate (b) naturalnews

(c) politicot (d) greenvillegazette

Figure 6.5: Performance Comparisons among CNT, AHGNA and proposed *FEND* approach based on mixed datasets.

## 6.2 Experimental Results of RTFEND

In this section, we first describe the details of an experimental testbed. Then, we carry out a group of experiments to demonstrate efficiency improvements from the perspectives of applying the two-stage procedure described in the previous section and increasing the numbers of slave nodes. Next, we compare processing times between the proposed system and the baseline counterpart that doesn't employ memory management. Finally, we compare our model with models deploying other approaches for detecting Fake news in terms of the number of topics, number of data clusters, and system performance. To demonstrate the efficacy of the proposed approach, we compare the performance of the proposed model with the baseline model developed by [136], which is not a real time fake news detection model. This ad-hoc fake news prediction model is a two-layered fake-news detection system. The first layer (i.e., the fake-topic detection layer) detects fake news that are outliers from the

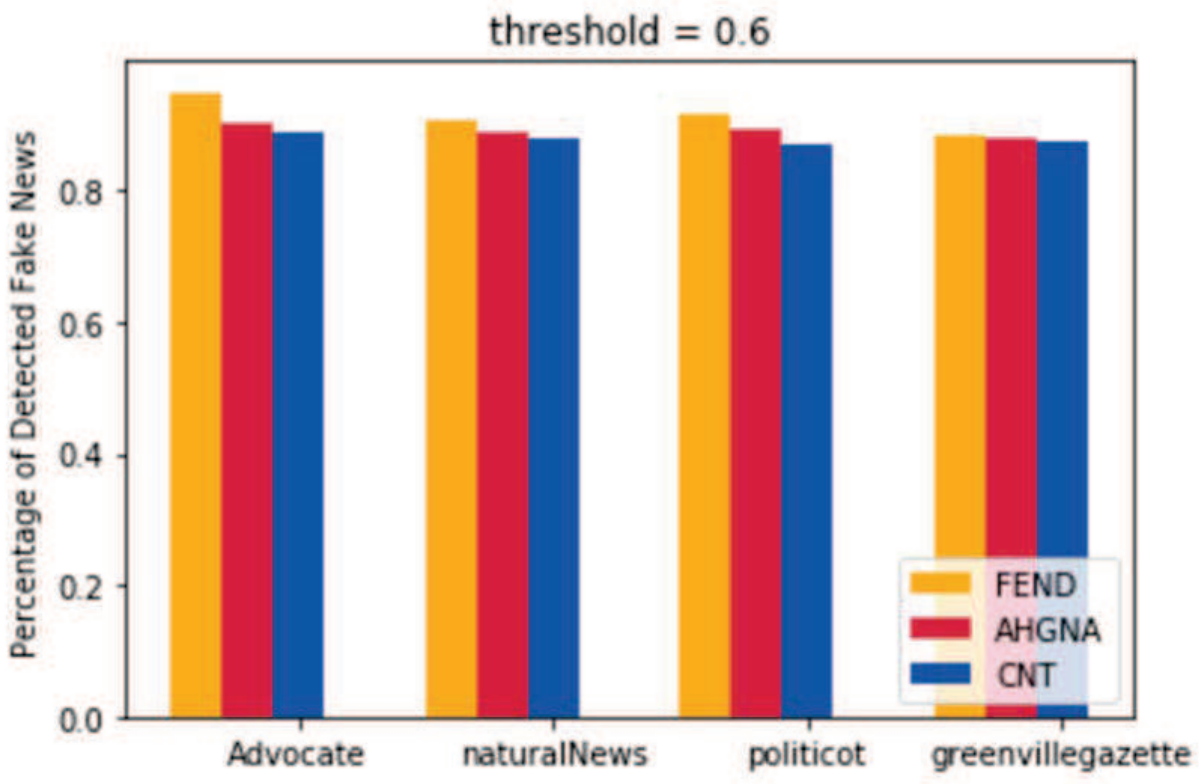



Figure 6.6: F-score comparison between FEND, AHGNA, and CNT when $\omega = 0.6$

perspective of the knowledge base (a.k.a., ground truth), which is comprised of data clusters built by acquiring legitimate news articles sharing similar topics. The second layer (i.e., the fake-event detection layer) is responsible for detecting fake news through an event analysis. Although the two layers appear similar in both models, the proposed model in this study utilizes different data processing mechanism based on real time streaming and algorithm optimization processes. These two components provide novelty to the approaches proposed in this study.

For performance comparison, we apply two laptops that include 32GB 1666MHz DDR3 memory, 32GB 2133MHz DDR3 memory, an Intel 2.4Ghz quad-core *i5* processor, and an Intel 2.6Ghz quad-core *i7* processor to construct a Hadoop & Spark-based compute cluster (i.e., one master node and up to eight slave nodes) using virtual machine environment to quantitatively evaluate the performance of the proposed model and compare it with baseline models. The master node has the fixed configuration that includes 8GB main memory

capacity and 2.6 Ghz quad-core *i7* processor. The configuration of slave nodes may change from perspectives of number, CPU and memory capacities according to different experiment requirements. All the nodes are orchestrated by the Rocks system framework, which is an open-source Linux cluster distribution. Apache Spark is set up as the distributed computing framework on this cluster.

### 6.2.1 Topic and Data Cluster Reduction

The goal of the first group of experiments is to measure the topic-reduction performance of the proposed approach. Since the baseline FEND model lacks the topic-reduction module, we evaluate the topic-reduction performance by comparing our model and FEND in terms of the number of topics as well as the number of data clusters. Evidence shows that FEND extracts approximate 200 thousand topics from 14,221 articles. These topics are independent of each other. Such an excessive number of topics inevitably triggers the high dimensionality issue in FEND. To overcome the deficiency of baseline FEND model, we introduced topic-merging mechanism that is embedded in our model in Section 4.1 (see also Example 1 for a similarity score between two topics). In this group of experiments, the similarity threshold is set to 0.85 in the topic-merging module.

Based on the four different fake-news datasets, we illustrate the improvements of the proposed model over FEND [136] in terms of the number of topics (see Fig. 6.7(a)) and the number of data clusters (see Fig. 6.7(b)). Among all the four datasets, the *NaturalNews* dataset enjoys the best topic-reduction performance achieved by the proposed model. More specifically, the topic- and cluster-reduction rates are 25.03%(i.e., 15,011 topics) and 30.95% (i.e. 13 clusters), respectively. Compared with FEND, the proposed approach reduces the number of topics in the *Advocate* dataset by 17.59% (i.e., around 18,000 reduced topics); the approach slashes the number of clusters by 25% (i.e., 14 reduced clusters). For the *greenvillegazette* dataset, our approach's topic- and cluster- reduction rates are 23.53% (i.e., 17,505) and 27.91% (i.e., 12). The poorest reduction performance was observed with the

*politicot* dataset; nevertheless, in this case the proposed approach reduces the numbers of topics and clusters by 12.89% (i.e., 5,910) and 23.81% (i.e., 5), respectively.

In summary, our proposed approach cuts back the numbers of topics and data clusters on averages of 19.76% and 26.92%, respectively. The results confirm that our approach judiciously reduces the number of topics by repeatedly (1) calculating the similarity of randomly selected topics, (2) refining synonymous topics, and (3) choosing the lowest similarity from these synonymous topics. The findings unveil that topic reductions help in cutting back the number of data clusters and; therefore, time spent in the data-processing stage and significantly shortening the model-testing stage.

### 6.2.2 Percentage of Detected Fake News

Fig. 6.8 plots the percentages of detected fake news in the two layers in the context of the four fake-news datasets; the threshold in our model is configured to 0.5, 0.6, 0.7, and 0.8, respectively in figs 6.8(a) 6.8(b) 6.8(c) 6.8(d).

We make two intriguing observations from Fig. 6.8. The first observation is that regardless of threshold value, the performance of the proposed model largely depends on input datasets. For example, among all the four datasets, the *Advocate* dataset benefits from the proposed model most; the *naturalNews* dataset leads to the lowest accuracy; the detection accuracies of the *politicot* and *greenvillegazette* datasets are somewhere between. The second observation is that there is no clear winner between Filter-1 and Filter-2. In the *Advocate* dataset, Filter- 1 is superior to Filter-2; in the *politicot* case, Filter-2 is better than Filter-1 in terms of accuracy; when it comes to the *naturalNews* and *greenvillegazette* datasets, Filter-1 is on par with Filter-2. These two observations suggest that the accuracy performance of the poroosed model largely depends on input datasets. There are some inherent differences among the four datasets. First, the detection rate of the filter-1 varies significantly across the four datasets. For example, the detection rate is the highest for the *advocate.com* dataset (i.e., 66.9%) and the lowest for the *politicops.com* dataset (i.e., 4.4%). Second, over

65% news in *advocate.com* are correctly detected by the first-level filter, meaning that a large portion of news posted in *advocate.com* have fake topics(detected by Filter-1). Third, the fake news detection rates by Filter-1 for *naturalnews.com* and *greenvillegazette.com* are medium low (i.e., 21.1% and 31.4%), indicating that a majority of news published on *naturalnews.com* and *greenvillegazette.com* have legitimate topics. Fourth, for the news found on *Politicot.com*, only 4.4% are detected as fake news by Filter-1, suggesting that almost of all the Politicot.com news contain trustworthy topics. Results suggest that first filtering layer provides a good mechanism for detecting the reliability of a news source. The output of the first layer is then fed to the second layer filter.

The third observation drawn from Fig. 6.8 is that Filter-1's performance is independent of threshold. Such an independence is expected, since the procedure used by Filter-1 for detecting fake topics is irrelevant to the threshold. Unlike Filter-1, Filter-2's performance is optimized by adjusting the threshold. In a set of news articles with legitimate topics, Filter-2 detects fake news items that comprise of fake events depending on the threshold value. Intuitively, Filter-2's detection performance is enhanced when the threshold value increases 6.8.

The fourth observation is that the second layer (i.e., filter- 2) is sensitive to threshold with respect to the percentage of detected fake news; in contrast, threshold has no impact on the first layer (i.e., filter-1) in the proposed system. Thus, an increasing number of fake news is detected by the second layer when we raise the threshold value. For example, when we push the threshold from 0.5 up to 0.6, filter-2's improvement in fake-news detection rate is 2%, 3%, 2%, and 4% for the *Advocate*, *naturalNews*, *politicot*, and *greenvillegazette* datasets, respectively. If the threshold increases from 0.6 to 0.7, such an improvement becomes 3%, 3%, 4%, and 2%, respectively. Similarly, if the threshold varies from 0.7 to 0.8, the improvement in filter-2 will be 1%, 2%, 2%, and 2%, respectively. This performance trend implies that there is no way to optimize filter-1's detection performance through the

threshold configuration; on the other end of the spectrum, one may boost filter-2's detection accuracy by adjusting the threshold.

Fig. 6.8 unravels the proposed model's accuracy performance when the threshold is in the range between 0.5 and 0.8. In case of the threshold that is smaller than 0.5 or larger than 0.8, we can derive the credibility distribution of the four datasets from the results plotted in Fig. 6.8 . For example, in the *Advocate*, *naturalNews*, *politicot*, and *greenvillegazette* datasets, the percentage of news items with credibility lower than 0.5 is around 24%, 40%, 61%, and 44%, respectively. Given the same four datasets, the percentage of news articles with credibility larger than 0.8 becomes 1.35%, 6.49% 4.46%, and 2.92%, respectively.

### 6.2.3 Scalability

The goal of this group of experiments is to measure the scalability of the proposed model from the perspectives of main memory size and the number of computing nodes based on the *New York Times* training dataset. It is worth noting that a computing cluster can be scaled up by either expanding main memory capacity or increasing the number of nodes. To focus on the scalability test, we disable the memory management technique. Nevertheless, the analysis on memory management is detailed in Section 4.3.3. We gauge the impact of the number of computing nodes on the processing time of our model's data preprocessing stage. More specifically, we vary the number of slave nodes from two to eight with an increment of two nodes. In addition to varying the number of nodes, we test two main-memory capacities (i.e., 2GB vs. 4GB) in the computing nodes. These two configurations shed some lights on the impact of main memory on the model's efficiency performance.

Fig. 6.9 demonstrates the processing-time trends with an increasing number of data points. Each sub-figure plots three curves representing three cases, namely, a standalone system (i.e., serial computing), 2-GB-node system, and 4-GB-node system. In the second and third cases, the main memory in computing nodes is set to 2 GB and 4GB, respectively. In what follows, we refer to these three cases as $Standalone$, $2-GBnodes$, and $4-GBnodes$.

Fig. 6.9 further demonstrates that regardless of the number of nodes, the performance gaps among *Standalone*, $2-GBnodes$ and $4-GBnodes$ are widened as input data size rises. For example, Fig. 6.9(a) unravels that the 2-GB- node and 4-GB-node systems shorten the running time of the standalone system by 701 and 1,850 seconds (i.e., from 6306 to 5605 and 4456 seconds), respectively for 17,656 news articles.

Comparing Figs. 6.9(a) and 6.9(b) using our proposed algorithms, we observe that the performance of our model significantly improves by increasing the number of computing nodes. For example, in the $2-GBnodes$ case, augmenting two extra nodes into the existing 2-node cluster allow us to cut back the processing time by 617 seconds (i.e., from 5605 to 4988 seconds). In the $4-GBnodes$ case, such processing-time reduction becomes 1,193 seconds (i.e., from 4456 to 3263 seconds). We draw a similar conclusion when we compare Figs. 6.9(b) with 6.9(c). For instance, increasing the number of nodes from four to eight reduces the processing time by 1,331 seconds (i.e., from 4988 to 3657 seconds) in the $2-GBnodes$ case. Such reduction is measured as 953 seconds (i.e., from 3263 to 2310 seconds) when it comes to the $4-GBnodes$ case.

The $4-GBnodes$ case in a 2-node cluster and the $2-GBnodes$ case in a 4-node cluster share the same total main memory capacity. Surprisingly, the 2-node cluster outperforms the 4-node cluster for such tests where input dataset varies from 3,000 to 17,656. For example, comparing in the $4-GBnodes$ case in Fig. 6.9(a) with the $2-GBnodes$ case in Fig. 6.9(b), we observe that the 2-node cluster is faster than the 4-node cluster by 532 seconds. Similarly, after we compare the $4-GBnodes$ case in Fig. 6.9(b) with $2-GBnodes$ case in Fig. 6.9(c), we notice that the 4-node cluster outperforms the 8-node cluster by 394 seconds. We conclude that keeping the overall main memory capacity as a constant, small-sized clusters for handling real time fake news are superior to large-sized counterparts due to reduced communication overhead.

The third observation is that when the system employs eight 2GB nodes, the system improves the performance by over 42.01% (i.e., the processing time is reduced from 6,306

to 3,657 seconds). If the system embraces eight 4GB nodes, the performance improvement becomes 63.37% (i.e., the processing time is reduced from 6,306 to 2,310 seconds) The reason for such noticeable improvements is two-fold. First, the valid usable main memory space of computing nodes is less than their entire memory space. Second, the system's overhead of assigning tasks to computing nodes is increased when the number of nodes is surging.

### 6.2.4 Memory Management

The key purpose of the last group of experiments is to measure the impact of the novel memory management technique that we proposed (see Section 4.3) to improve the processing time of the data pre-processing stage in the proposed model. Since the model's main goal is to perform real time analytics, an appropriate computing environment is required for its effective functioning. In this section, we evaluate and compare the efficiency of the memory-management-enabled model (Fig. 6.10(c) with the same system where no memory management (see Fig. 6.9(c)) is involved.

The model's performance in this group of experiments is better than its performance in earlier experiments shown in Section 6.2.3. For example, Fig. 6.10(a) illustrates that the processing time is reduced by 1,101 seconds (i.e., from 6306 to 5205) after adding two 2GB nodes into the standalone system; such reduction becomes 2,454 seconds (i.e., from 6306 to 3852) with two 4GB nodes. If we include eight 2GB nodes in the standalone system, the processing time decreases by 3,329 seconds (i.e., from 6306 to 2977); if the eight newly added nodes have 4GB main memory, the processing time is slashed by 4,465 seconds (i.e., from 6306 to 1841). Such reductions are substantial for performing a real time analytics task such as fake news detection.

On one hand, by comparing the baseline system processing time perspective, our system promotes computing efficiency by 52.79% if eight 2GB slave nodes expanded into the baseline system; the enhancement becomes 70.81% if eight 4GB nodes are deployed. On the other hand, the system optimizes the efficiency by 18.59% and 20.3% after applying

the memory management technique to an eight-2GB-node system and an eight-4GB-node system, respectively.

In a nutshell, we have demonstrated that the efficiency of the proposed model can be improved by both the Spark-based memory management (see Section 6.2.4) and the novel batch-size optimization procedure that we have proposed in this study (see Section 4.3.3). Such considerations are critical for understanding the how to effectively implement the proposed model for use in different organizational environment have varied computing infrastructure.

## 6.3 Experimental Results of OSS

In this section, we conduct three experiment groups by varying the three parameters, namely, (1) the view points of subjects (See Algorithm 3); (2) the view points of subject-related objects (See Algorithm 4); and (3) the subject-object-related tense (See Algorithm 5). In the first experiment group, we manipulate these three parameters to construct three two-elemental variable sets (i.e., ($p_s$, $p_o$),($p_s$, $q$), and ($p_o$, $q$)) to distinguish objective and subjective sentences. The second experiment group aims to extract objective and subjective patterns from the three-element variable set ($p_s$, $q$, $q_o$). Finally, we evaluate the performance of the extracted patterns depending on the given threshold list using the cross validation method.

### 6.3.1 Datasets

We adopt an existing labeled dataset [86] that encompasses (1) 5,000 movie review snippets from Rottentomatoes[1] as a subjective data collection and (2) 5,000 objective data items from the plot summaries available from the Internet Movie Database[2] to evaluate our method. All the tested sentences or snippets, each of which consists of at least ten words,

[1] www.rottentomatoes.com
[2] www.imdb.com

are originated from either movie reviews or plot summaries. As such, no redundant data exists in the dataset.

### 6.3.2 Experiment Group 1: Two-Elemental Variable

Table 6.8: All subject-tense pairs exclusively occurred in subjective or objective datasets.

| Subjective ($p_i^u$, $q_i$) Pairs | Objective ($p_i^u$, $q_i$) Pairs |
|---|---|
| ((3, 11), 1), ((0, 2), 1), ((4, 4), 2), ((0, 3), 1), ((1, 12), 11), ((1, 11), 1), ((1, 10), 3), ((1, 7), 8) | (3, 10), 4), ((2, 11), 1)) |

In this group of experiments, we gauge the frequencies of two-element pairs (($p_i^u, q_i$), ($p_i^u, p_i^o$), and ($q_i, p_i^o$)) to compare their differences in objective and subjective datasets. Two types of pairs are demonstrated for each two-element pairs: (1) the pairs uniquely occur in subjective or objective datasets (e.g., pair (1, 7) appears 8 and 0 times in the objective and subjective datasets, respectively); (2) the pairs are highly unbalanced among the two datasets (e.g., pair (3, 6) appears 18 and 113 times in the objective and subjective datasets, respectively).

**Two-Element Pair ($p_i^u, q_i$)**: First, Table 6.8 lists all the extracted ($p_i^u, q_i$) pairs that are unique in the subjective or objective datasets. In this table, there are eight unique ($p_i^u, q_i$) pairs in the subjective dataset. For example, $((1, 12), 11)$ represents that pair $(1, 12)$ appears 11 times in the subjective dataset only. Similarly, there are two pure objective ($p_i^u, q_i$) pairs obtained from the objective dataset; the frequencies of pairs $(3, 10)$ and $(2, 11)$ are 4 and 1, respectively.

Next, we list all highly unbalanced ($p_i^u, q_i$) pairs in Table 6.9. The table is divided as two components that the first seven pairs (from (1, 6) to (2, 6)) have significantly higher frequencies in the subjective dataset than the objective one. For example, pair $(1, 1)$ appears 16 times in the subjective dataset, but the pair occurs only two times in the objective dataset; the probability of being a subjective statement is $\frac{16}{16+2} = 88.89\%$. In contract, the last ten

Table 6.9: All ($p_i^u$, $q_i$) pairs comparisons.

| ($p_i^u$, $q_i$) Pairs | Frequency in Dataset S | Frequency in Dataset O | Probability (in %) |
|---|---|---|---|
| (1, 6) | 7 | 3 | 70.00 |
| (2, 12) | 5 | 2 | 71.43 |
| (1, 4) | 8 | 3 | 72.73 |
| (2, 8) | 57 | 13 | 81.43 |
| (2, 7) | 8 | 1 | 88.89 |
| (1, 1) | 16 | 2 | 88.89 |
| (2, 6) | 9 | 1 | 90 |
| (3, 8) | 321 | 803 | 28.56 |
| (4, 12) | 2 | 5 | 28.57 |
| (0, 6) | 128 | 327 | 28.13 |
| (4, 8) | 20 | 59 | 25.32 |
| (3, 7) | 19 | 89 | 17.59 |
| (4, 6) | 2 | 8 | 20.00 |
| (0, 10) | 6 | 24 | 20.00 |
| (4, 7) | 1 | 4 | 20.00 |
| (0, 11) | 1 | 5 | 16.67 |
| (3, 6) | 18 | 113 | 13.74 |

pairs (from (3, 8) to (3, 6)) have remarkably lower frequencies in the subjective dataset than the objective dataset. The pair (3, 6) has the lowest probability of being subjective patterns ($\frac{18}{113+18} = 13.74\%$), because this pair appears 113 and 18 times in the objective and subjective datasets.

**Two-Elemental Pair ($p_i^u, p_i^o$)**: Table 6.10 displays all the unique $(p_i^u, p_i^o)$ pairs from the two datasets. The table shows that the numbers of unique $(p_i^u, p_i^o)$ pairs in two datasets are 3 and 0, respectively. For example, pair $(2, 4)$ occurs twice in the subjective dataset only.

Table 6.10: All subject-object pairs exclusively occur in subjective or objective datasets.

| Subjective ($p_i^u$,$p_i^o$) Pairs | Objective ($p_i^u$,$p_i^o$) Pairs |
|---|---|
| ((4, 2), 1), ((2, 1), 1), ((2, 4), 2) | Null |

Next, we reveal the frequency discrepancies of subject-object pairs ($p_i^u$, $p_i^o$) in the two datasets (See Table 6.11). Table 6.11 consists of two partitions that the first 7 ($p_i^u$, $p_i^o$) pairs (from (0, 2) to (2, 2)) have significantly higher frequencies in the subjective dataset than in

the objective one. For example, the pair (2, 0) appears 65 times in the subjective dataset but only 15 times in the objective dataset. The second partition of the table contains 5 pairs ranging from (0, 3) to (3, 4). All of these five pairs have very low percentages, because such pairs appear much more times in the objective dataset than in the subjective dataset. Pair (3, 4) has the lowest percentage among all the five pairs, meaning that this pair has the best possibility of being used in objective statements.

Table 6.11: All ($p_i^u$, $p_i^o$) pairs comparisons.

| ($p_i^u$, $p_i^o$) Pairs | Frequency in Dataset S | Frequency in Dataset O | Probability (in %) |
|---|---|---|---|
| (0, 2) | 20 | 8 | 71.43 |
| (0, 1) | 31 | 11 | 73.81 |
| (1, 4) | 3 | 1 | 75.00 |
| (3, 1) | 9 | 3 | 85.00 |
| (2, 3) | 13 | 3 | 81.25 |
| (2, 0) | 65 | 15 | 81.25 |
| (2, 2) | 14 | 2 | 87.50 |
| (0, 3) | 199 | 489 | 28.92 |
| (4, 0) | 23 | 66 | 25.84 |
| (4, 4) | 1 | 3 | 25.00 |
| (4, 3) | 2 | 7 | 22.22 |
| (3, 4) | 8 | 33 | 19.51 |

**Two-Element Pair ($q_i, p_i^o$)**: We are positioned to form pairs ($q_i, p_i^o$) to investigate any distinct differences between the objective and subjective datasets from perspectives of pair frequencies. Table 6.13 enumerates pair frequencies in the two datasets coupled with subjective probabilities.

Table 6.12: All tense-object pairs are exclusively occurred in the subjective or objective datasets.

| Subjective ($q_i, p_i^o$ ) Pairs | Objective ($q_i, p_i^o$) Pairs |
|---|---|
| ((3, 0), 1), ((4, 1), 1), ((2, 0), 1), ((12, 2), 1) | ((10, 4), 3), ((12, 4), 8), ((11, 4), 1), ((6, 1), 1), ((11, 3), 2), ((10, 3), 2) |

Tables 6.10 and 6.8 shows all the subject-object pairs and subjective-tense pairs that are exclusively discovered in the subjective or objective datasets. Similarly, Table 6.12

illustrates all the distinctive $(q_i, p_i^o)$ pairs that are collected into the subjective-only pair set and objective-only pair set. In Table 6.12, four subjective $(q_i, p_i^o)$ pairs are exclusive to the objective dataset. However, the frequencies of all these four pairs are equal to 1. Objective $(q_i, p_i^o)$ pair set consists of six pairs: (10, 4), (12, 4), (11, 4), (6, 1), (11, 3), and (10, 3). Pair (11, 3) represents that a $(q_i, p_i^o)$ pair has past continuous tense and its object's is a third person pronoun (view point $p_i^o = 3$). Particularly, pair (12, 4) occurs eight times only in the objective dataset.

Table 6.13: All $(q_i, p_i^o)$ pairs comparisons.

| $(q_i, p_i^o)$ Pairs | Frequency in Dataset S | Frequency in Dataset O | Probability (in %) |
|---|---|---|---|
| (8, 1) | 42 | 16 | 72.41 |
| (7, 2) | 3 | 1 | 75.00 |
| (8, 2) | 31 | 10 | 75.61 |
| (4, 2) | 8 | 2 | 80.00 |
| (10, 0) | 9 | 23 | 28.13 |
| (7, 4) | 3 | 7 | 30.00 |
| (6, 0) | 142 | 376 | 27.41 |
| (8, 3) | 182 | 562 | 24.46 |
| (8, 4) | 31 | 97 | 24.22 |
| (12, 3) | 10 | 30 | 25.00 |
| (6, 3) | 18 | 58 | 23.68 |
| (6, 4) | 4 | 17 | 19.05 |
| (7, 3) | 16 | 78 | 17.02 |

Table 6.13 embraces two partitions including 13 $(q_i, p_o^u)$ pairs in total. In this table, four pairs are classified in the first partition, and nine pairs are folded in the second partition of the table.

Tables 6.9 6.11 list all the subject-object pairs and subjective-tense pairs that are highly imbalanced between the objective and subjective datasets. Similarly, Table 6.13 collects all the $(q_i, p_o^u)$ pairs that are unbalanced in the two datasets. All such pairs can be divided into two partitions. The first partition contains four pairs that have the subjective probabilities larger than 70%. In contrast, the second partition has the pairs whose probabilities are lower than 30%. Therefore, the pairs in the first partition are more likely to appear in the

subjective dataset, and the other pairs in the table prefer to be used in the objective dataset. A remarkable observation drawn from Table 6.13 is that the scope of objects' view points $p_i^o \in \{1, 2, 3, 4, 0\}$ can be classified as $p_i^o u \in \{1, 2\}$, and $p_i^o o \in \{3, 4, 0\}$, where $p_i^o u$ represents all objects' view points in the subjective dataset, and $p_i^o o$ denotes all objects' view points in the objective dataset. The results plotted in Table 6.12 and Table 6.13 confirm this classification trend.

Table 6.14: Unique objective and subjective triple Patterns.

| **Unique Subjective Patterns** | **Unique Objective Patterns** |
|---|---|
| ((0, 4, 1), 1), ((2, 7, 3), 1), ((2, 6, 3), 1), ((1, 12, 3), 3), ((1, 12, 0), 8), ((4, 4, 0), 2), ((0, 3, 0), 1), ((0, 12, 1), 2), ((0, 12, 2), 1), ((3, 11, 0), 1), ((0, 2, 0), 1), ((2, 4, 3), 2), ((2, 12, 3), 2), ((4, 7, 2), 1), ((1, 10, 0), 3), ((1, 7, 3), 1), ((2, 6, 4), 1), ((1, 11, 0), 1), ((2, 8, 1), 1), ((2, 4, 2), 2), ((2, 8, 4), 1), ((1, 7, 0), 7), | ((0, 10, 4), 2), ((4, 12, 4), 1), ((0, 11, 4), 1), ((0, 6, 1), 3), ((4, 6, 3), 1), ((0, 11, 3), 2), ((4, 7, 0), 3), ((3, 12, 4), 3), ((3, 12, 3), 10), ((3, 12, 1), 1), ((3, 10, 0), 3), ((0, 12, 4), 4), ((3, 10, 4), 2), ((2, 11, 0), 1), ((3, 7, 4), 3), ((3, 7, 3), 22), ((0, 10, 3), 2), ((4, 7, 4), 1), |

### 6.3.3 Experiment Group 2: Three-Element Patterns

In this group of experiments, we dive into the investigation of three-element patterns $(p_i^u, q_i, p_i^o)$. More specifically, we calculate all $(p_i^u, q_i, p_i^o)$ triples' possibility of being subjective statements or objective ones. Table 6.14 enumerates unique subjective or objective $(p_i^u, q_i, p_i^o)$ patterns accompanied by frequencies. For example, Triple (1, 10, 3) appears three times in the subjective dataset only. Table 6.15 includes $(p_i^u, q_i, p_i^o)$ triples with high-likelihood of being subjective or objective statements.

Table 6.14 reveals that there are 23 subjective patterns and 10 objective patterns. Each triple in the table appears either in the subjective or objective datasets. We draw three

observations from Table 6.14. First, the view point of subject $p_i^u = 3$ only shows up once; $p_i^u = 4$ just occurs three times. Thus, we have ((3, 11, 0), 1), ((4, 4, 0), 2), and ((4, 7, 2), 1) in the unique subjective pattern list. In contrast, objective patterns prefer to use $p_i^u = 3or4$. Second, the view point of subject $p_i^u = 1or2$ just appears once (i.e., ((2, 11, 0), 1)) in the unique objective pattern list, but $p_i^u = 1or2$ have higher frequencies in the subjective dataset. Sum of above, the unique subjective patterns prefer to use $p_i^u \in \{0, 1, 2\}$ as their subjects' view points, but the unique objective patterns favor the usage of $p_i^u \in \{0, 3, 4\}$ as their subjects' view points. Second, the future tense ($q_i <= 4$) only occurs in the unique subjective patterns rather than the objective ones. Last, the view point of objects $p_i^o = 2$ appears in the subjective dataset only, indicating that unique subjective patterns more likely to use $p_i^o = 2$ than unique objective patterns.

Table 6.15 lists all the objective and subjective patterns whose possibilities fall into a window between 70% and 100%. There are 13 subjective patterns in Table 6.15. The possibilities of the first six subjective patterns are in the range between 70% and 80%. The possibilities of all the 13 subjective patterns are lower than 90%. Three triples' possibilities of being objective patterns are higher than 90%; these triples are (3, 4, 3), (3, 8, 3), and (3, 6, 3). The objective pattern (3, 8, 3) not only has a high probability, but also its frequency is significantly larger than those of the other objective patterns with high probability (e.g., ¿ 80%).

Recall the three observations we obtained from Table 6.14. First, $p_i^u \in \{0, 1, 2\}$ has much higher frequencies in the subjective patterns than in the objective ones, and $p_i^u \in \{0, 3, 4\}$ is more likely to be used in the unique objective patterns. Surprisingly, we observe that $p_i^u \in \{1, 2\}$ never pop up in the objective patterns, but they are massively applied in the subjective patterns. On the other hand, $p_i^u = 3or4$ only occurs four or zero times in the subjective dataset, but view points of all objective patterns' subjects have $p_i^u \in 0, 3, 4$. Therefore, we conclude that a triple with a view point of subject $p_i^u \in \{1, 2\}$ is more likely to be a subjective pattern, and the triples with view points of subject $p_i^u \in \{3, 4\}$ tend to

be an objective pattern. Second, the number of subjective patterns that have future tenses ($q_i <= 4$) is three, but only one objective pattern applies to future tense. Last, the view point of objects $p_i^o = 2$ only appears in the subjective patterns. In other word, a pattern that have an object matching $p_i^o = 2$ is more likely to be a subjective one.

### 6.3.4 Experiment Group 3: Cross Validation

In this group of experiments, we dive into the effectiveness evaluation of our proposed approach through the 10-fold cross validation. The training dataset constitutes nine folds of the divided datasets, whereas the remaining one fold is utilized as a testing dataset. We assess our system's capability of detecting objective and subjective statements by extracting subjective patterns and objective patterns from the datasets in the context of subjective- and objective-sentence testings. Recall that in Table 6.15, each pattern has a chance of being a subjective pattern or an objective one. Thus, we configure $\varphi$ as a threshold to elect objective and subjective patterns. Given a training dataset, varying threshold $\varphi$ may output different pattern-sets. More specifically, the increasing value of threshold $\varphi$ curtails the number of subjective patterns and objective patterns yielded from the model. According to the massive numbers of training and testing processes on the applied datasets, we set up such a threshold as $\varphi \in \{70.00\%, 72.50\%, 75.00\%, 77.50\%, 80.00\%, 82.50\%, 85.00\%,$
$87.50\%, 90.00\%\}$ in our empirical study. Table 6.15 lists all the extracted objective and subjective patterns when the threshold is larger than 70.00%.

To analyze the stability of the our method, we implement and execute Algorithm 6 for 100 times. In each run, the model takes randomly sorted objective and subjective datasets as an input. Ten validations are reported for each run; thus, the total time of executions is 1,000. For example, the threshold of 0.7 (see Table 6.16) leads to the subjective- pattern precision of 77.37%, which is calculated by two steps: 1) calculating the average results over the 10 validations and 2) executing step 1 for 100 times followed by the mean value across the 100 runs. Table 6.16 summarizes the performance of the model using the four metrics,

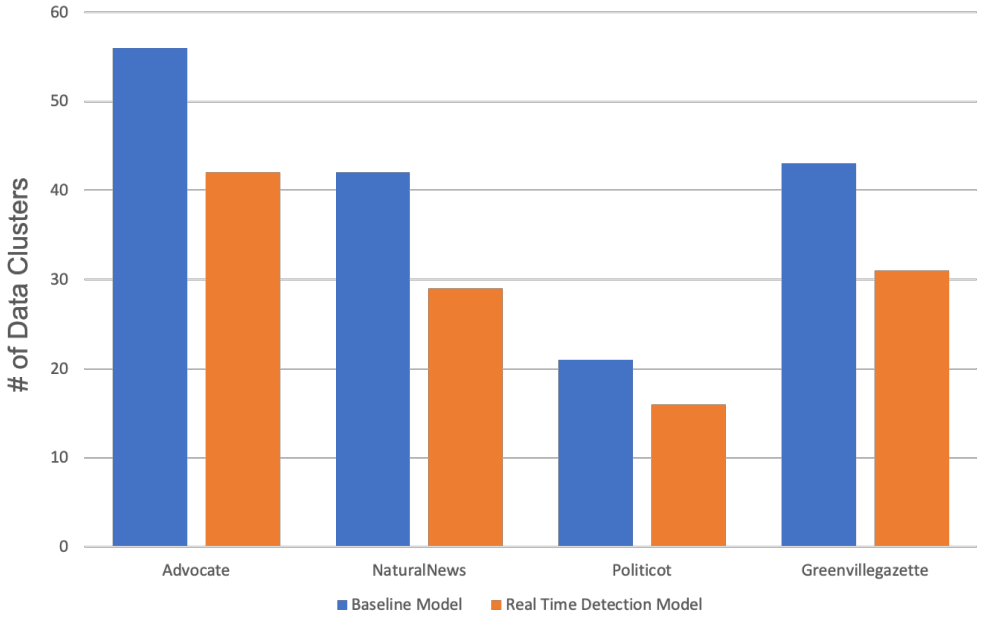
namely, accuracy, precision, recall, and F-measure. We also apply Eq. 6.1 to assess the variance (i.e., $\sigma^2$) of the four metrics upon different thresholds in the table.

$$\sigma^2(X) = \frac{\sum_{i=1}^{100}(x_i - \mu(X))^2}{100}, \tag{6.1}$$
$$X \in \{P_s, R_s, F_s, P_o, R_o, F_o, A\}$$

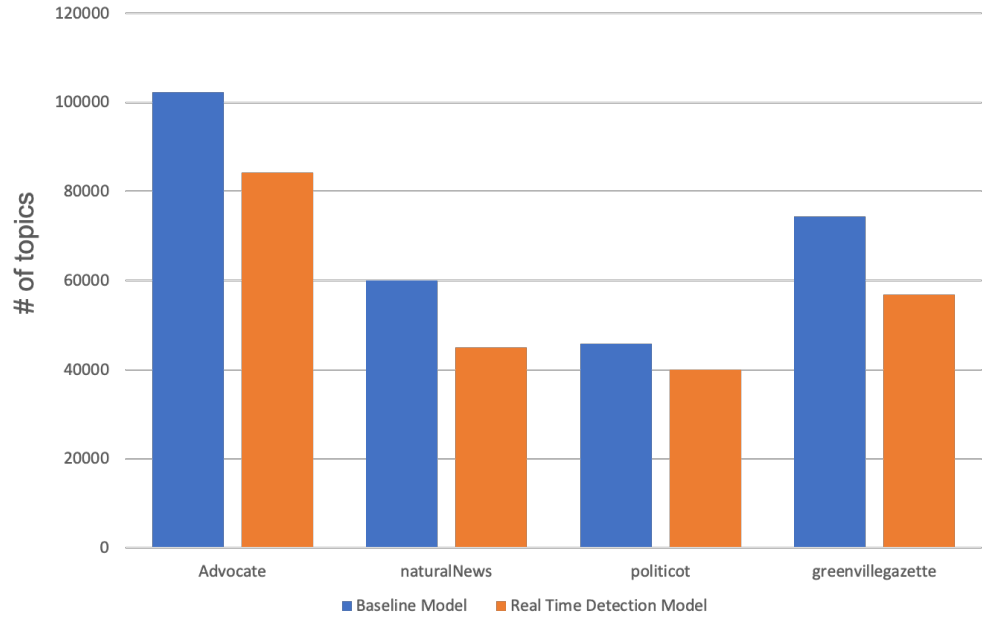
where $\sigma^2(X)$ represents the variance of $X$. $\mu(X)$ indicates the mean value of $X$. $P_s, R_s, F_s$ denote the precision, recall, and F-score measures of extracted subjective patterns. $P_o, R_o, F_o$ indicate precision, recall, and F-score of extracted objective patterns. $A$ is an accuracy over all the extracted patterns.

There are five intriguing observations obtained from Table 6.16. First, when the threshold is lifted from 0.70 to 0.80, all the ten measures share an ascending trend except for *recall* $R_o$, which stays at the high end (i.e., $R_o \geq 95.17\%$) within the threshold range $(0.7, 0.8)$. Second, when the threshold surges from 0.80, all the ten measures are decreasing except for the precision $P_o$ and the F-score $F_o$. Precision $P_o$ and F-score $F_o$ climb to the peak at 91.72% and 92.63% when the threshold is set to 0.9. In contrast, precision $P_s$ drops from 76.51% to 57.30%, because there are only unique-subjective patterns (Table 6.14) left in the training dataset when the threshold is larger than 87.5% (Table 6.15). Meanwhile, scarce pairs of unique-subjective patterns cause an unbalance between training and testing datasets. Third, the model delivers the best performance when the threshold is set to 0.8. In this case, the average precision, recall, and F-score reach to the high levels of 81.83%, 71.38%, and 73.48%, respectively. The subjective-pattern precision is the highest across all the threshold values. Fourth, our model performs very well on detecting objective statements all the time that delivers $\min(P_o) = 76.4\%$, $\min(R_o) = 93.9\%$, and $\min(F_o) = 85.44\%$. Finally, the variances of objective-related measures (i.e., $\sigma^2(P_o)$, $\sigma^2(R_o)$, and $\sigma^2(F_o)$) keep on a relatively low level. However, the variances of subjective-related measures (i.e., $\sigma^2(P_s)$, $\sigma^2(R_s)$, and $\sigma^2(F_s)$) are

increasing to high levels, because the numbers of objective and subjective patterns are decreasing with the increasing threshold value. Similar to the second observation, this trend triggers unbalanced data distributions between the training and testing datasets.

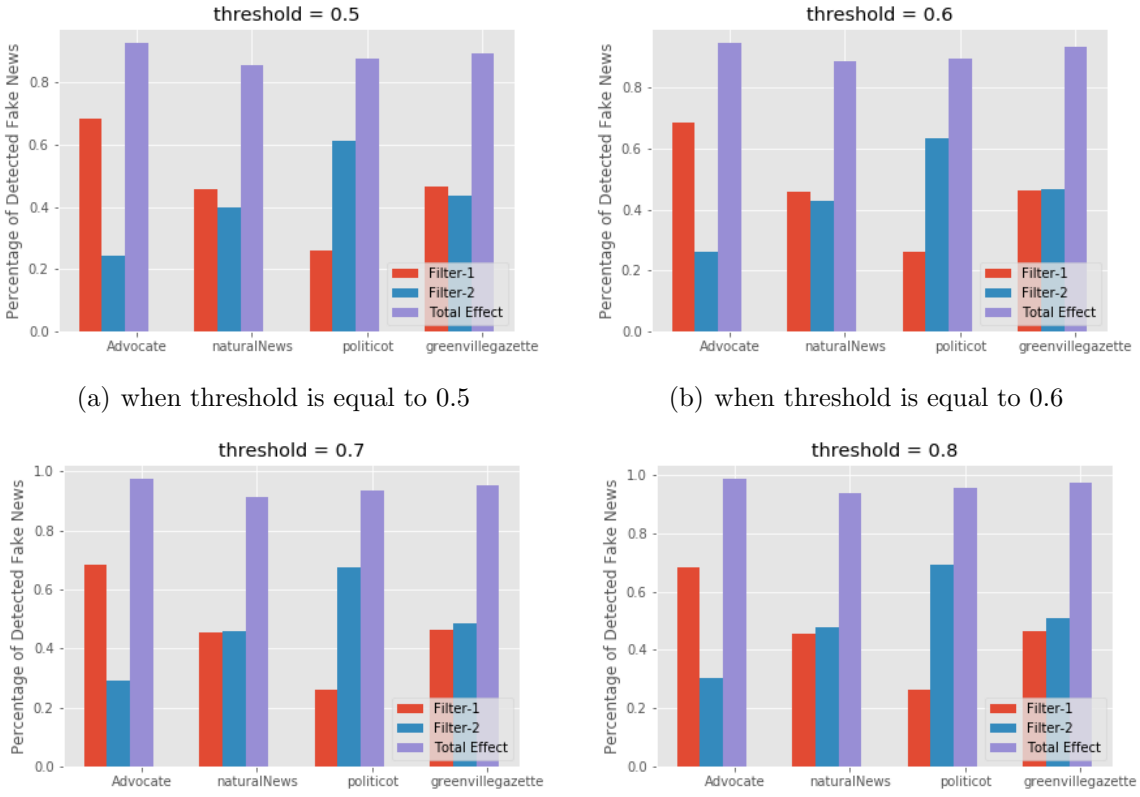



(a) the number of topics




(b) the number of clusters

Figure 6.7: Comparisons of the number of topics and clusters generated by FEND and the proposed approach.



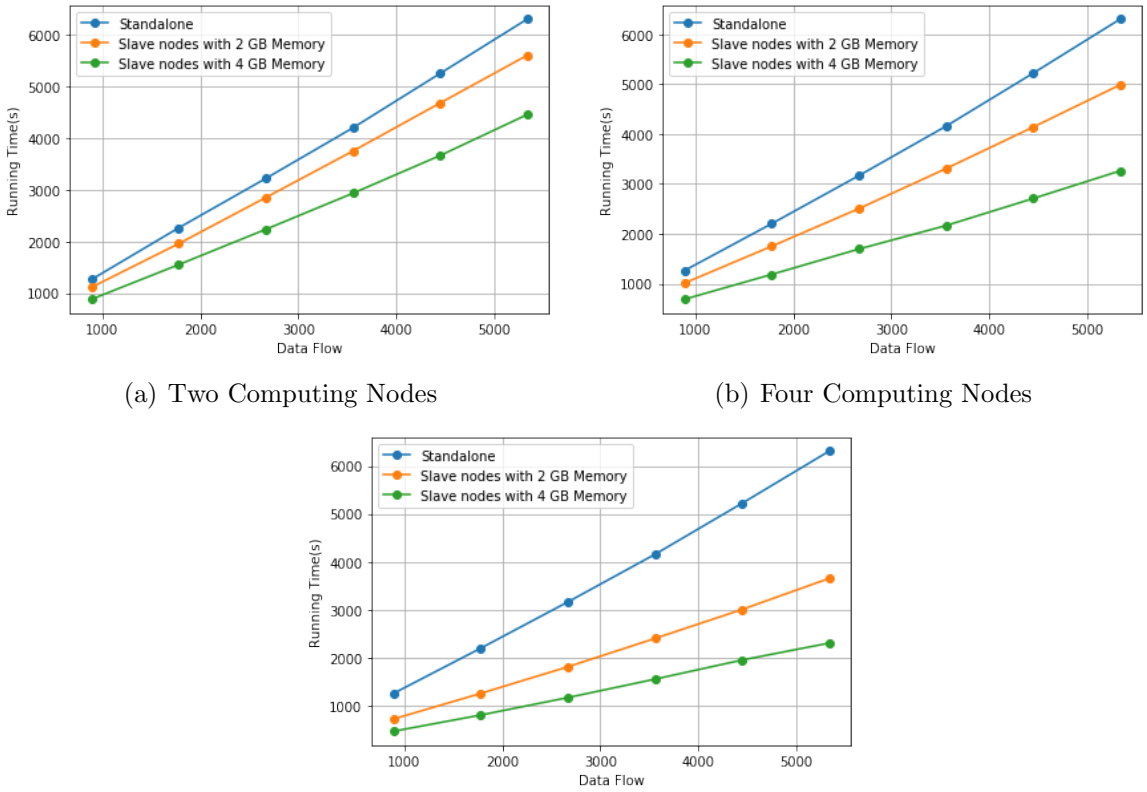

(a) when threshold is equal to 0.5

(b) when threshold is equal to 0.6

(c) when threshold is equal to 0.7

(d) when threshold is equal to 0.8

Figure 6.8: Percentage of detected fake news in layer 1 (i.e., filter-1), layer 2 (i.e., filter-2), and the combined layers of filter-1 and filter 2. The threshold in the model is set to 0.5, 0.6, 0.7, and 0.8, respectively.

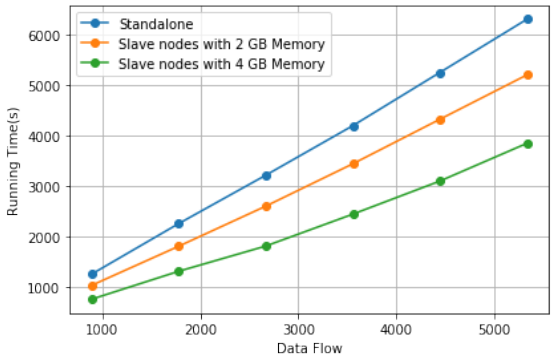



(a) Two Computing Nodes

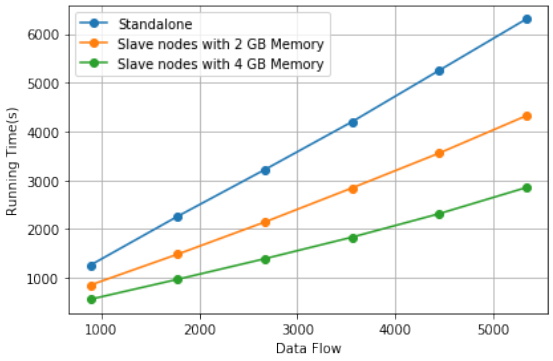
(b) Four Computing Nodes

(c) Eight Computing Nodes

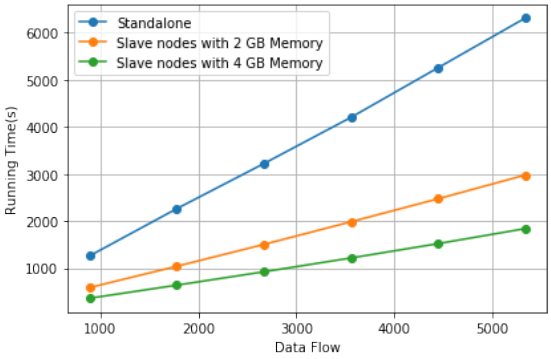
Figure 6.9: Impacts of the number of computing nodes on the processing time of the data pre-processing stage in proposed model.




(a) Two Computing Nodes




(b) Four Computing Nodes




(c) Eight Computing Nodes

Figure 6.10: Impacts of the number of computing nodes on the processing time of the data pre-processing stage after applying memory optimization to the model.

Table 6.15: Lists of extracted significant triples coupled with frequencies and possibilities of being subjective patterns or objective patterns.

| Subjective Patterns | | | | Objective Patterns | | | |
|---|---|---|---|---|---|---|---|
| Pattern | (Freq in S) | (Freq in O) | Possibility | Pattern | (Freq in S) | (Freq in O) | Possibility |
| (1, 6, 0) | 7 | 3 | 70.00 | (0, 6, 0) | 110 | 282 | 71.94 |
| (2, 8, 3) | 7 | 3 | 70.00 | (0, 8, 3) | 152 | 364 | 70.54 |
| (0, 8, 1) | 28 | 10 | 73.68 | (3, 7, 0) | 22 | 64 | 74.42 |
| (1, 8, 4) | 3 | 1 | 75.00 | (3, 8, 4) | 6 | 17 | 73.91 |
| (1, 4, 0) | 7 | 2 | 77.78 | (4, 8, 3) | 2 | 6 | 75.00 |
| (3, 8, 2) | 7 | 2 | 77.78 | (4, 8, 0) | 17 | 52 | 75.36 |
| (3, 8, 1) | 9 | 2 | 81.82 | (0, 10, 0) | 6 | 20 | 76.92 |
| (2, 8, 0) | 36 | 8 | 81.82 | (4, 6, 0) | 2 | 7 | 77.78 |
| (0, 4, 2) | 5 | 1 | 83.33 | (0, 8, 4) | 20 | 78 | 79.59 |
| (2, 4, 0) | 12 | 2 | 85.71 | (0, 6, 4) | 2 | 9 | 81.82 |
| (2, 8, 2) | 12 | 2 | 85.71 | (3, 12, 0) | 9 | 40 | 81.63 |
| (2, 7, 0) | 7 | 1 | 87.50 | (0, 7, 3) | 14 | 56 | 80.00 |
| (2, 6, 0) | 7 | 1 | 87.5 | (0, 12, 3) | 5 | 20 | 80.00 |
| | | | | (3, 6, 0) | 16 | 83 | 83.84 |
| | | | | (3, 6, 4) | 1 | 8 | 88.89 |
| | | | | (3, 4, 3) | 1 | 10 | 90.91 |
| | | | | (3, 8, 3) | 18 | 184 | 91.09 |
| | | | | (3, 6, 3) | 1 | 22 | 95.65 |

Table 6.16: Performance evaluations on extracted subjective patterns and objective patterns using the four metrics under various thresholds.

| **Threshold** | **Subjective (in %)** | | | **Objective (in %)** | | | **Average (in %)** | | | **Accuracy (in %)** |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F-Score** | **Precision** | **Recall** | **F-Score** | **Precision** | **Recall** | **F-Score** | |
| | $\sigma^2(P_s)$ | $\sigma^2(R_s)$ | $\sigma^2(F_s)$ | $\sigma^2(P_o)$ | $\sigma^2(R_o)$ | $\sigma^2(F_o)$ | $\sigma^2(P_a)$ | $\sigma^2(R_a)$ | $\sigma^2(F_a)$ | $\sigma^2(A)$ |
| **0.7** | 77.37 | 25.33 | 37.67 | 76.40 | 97.00 | 85.44 | 76.89 | 61.16 | 61.55 | 76.46 |
| | 0.99 | 0.52 | 0.71 | 0.08 | 0.02 | 0.03 | 0.53 | 0.27 | 0.37 | 0.08 |
| **0.725** | 76.39 | 40.51 | 51.88 | 80.74 | 95.17 | 87.27 | 78.56 | 67.84 | 69.58 | 80.00 |
| | 1.11 | 1.43 | 1.31 | 0.20 | 0.07 | 0.08 | 0.66 | 0.75 | 0.69 | 0.17 |
| **0.75** | 77.35 | 42.84 | 54.33 | 82.01 | 95.18 | 88.04 | 79.68 | 69.01 | 71.19 | 81.9 |
| | 1.27 | 1.06 | 0.93 | 0.15 | 0.08 | 0.07 | 0.71 | 0.57 | 0.50 | 0.15 |
| **0.775** | 79.91 | 44.77 | 56.46 | 82.87 | 95.85 | 88.80 | 81.39 | 70.31 | 72.63 | 82.34 |
| | 1.37 | 1.35 | 1.14 | 0.21 | 0.08 | 0.08 | 0.79 | 0.71 | 0.61 | 0.18 |
| **0.8** | 80.76 | 47.26 | 58.31 | 82.90 | 95.50 | 88.65 | 81.83 | 71.38 | 73.48 | 82.41 |
| | 1.57 | 1.87 | 1.47 | 0.25 | 0.11 | 0.10 | 0.91 | 0.99 | 0.79 | 0.21 |
| **0.825** | 76.51 | 45.67 | 54.54 | 85.43 | 94.97 | 89.80 | 80.97 | 70.32 | 72.17 | 83.70 |
| | 2.70 | 3.14 | 2.24 | 0.32 | 0.18 | 0.14 | 1.51 | 1.67 | 1.19 | 0.28 |
| **0.85** | 67.05 | 45.11 | 49.32 | 89.00 | 94.60 | 91.54 | 78.02 | 69.86 | 70.43 | 85.81 |
| | 1.3 | 3.3 | 2 | 0.7 | 0.2 | 0.3 | 1 | 1.8 | 1.1 | 0.5 |
| **0.875** | 51.54 | 48.40 | 47.34 | 91.03 | 93.17 | 91.98 | 71.28 | 70.78 | 69.66 | 86.32 |
| | 1.81 | 5.34 | 2.29 | 0.33 | 0.14 | 0.14 | 1.07 | 2.76 | 1.22 | 0.34 |
| **0.9** | 57.30 | 52.98 | 52.76 | 91.72 | 93.90 | 92.63 | 74.51 | 73.44 | 72.69 | 87.57 |
| | 1.32 | 5.68 | 2.34 | 0.31 | 0.22 | 0.17 | 0.82 | 2.95 | 1.26 | 0.30 |

# Chapter 7

# Conclusions

Research on fake news is in nascent stages. As fake news becomes more permeated and difficult to detect, increasingly sophisticated approaches are needed to detect fake news. The misinformation spread by fake news poses serious risk for its consumers and target, which could be individuals as well as enterprises. While an individual consuming the fake news develops distorted or misinterpreted perception of reality, which influences their beliefs and decision making, enterprises suffer from fake news due to loss of competitive advantage or damaging impact on their brand.

## 7.1 Summary of FEND

In the study of FEND, we propose a novel analytics-driven framework for detecting fake news. We then describe the *FEND* system, which implements the proposed framework for fake news detection and provides its validation. This study also required the development a comprehensive repository of real and fake news which may be utilized for developing future work in this important area of research. This framework utilizes a double-layered approach for classification. The first layer performs fake topic detection and the second layer performs fake event detection, leading to an overall average accuracy of 91.9%. Our approach is novel in the sense that each news article is translated into events, which departs from the traditional approaches of fake news detection that are merely based on syntax rules or sentiments. Our main objective in this study is to develop models that can deal with fake news detection, which is a challenging problem and poses risk for wide sector of population and organizations.

The study has several limitations. There is a focused stream of research on distinguishing fact versus opinion articles (e.g., [117], [107], [78], [96]). We also note that there are other

types of news categories, like satire, which are outside the scope of this study, but can be an area of future research based on recent studies(e.g., [48], [103]). Future studies could specifically focus on models trained on opinions and perspectives. Validation of such a dataset for training purposes may require additional steps to reliably classify them. This study uses clustering approaches to segregate fake news from the real news. Future work on fake news detection can focus on using advanced approaches such as deep learning. The validation of the *FEND* system that implements the proposed framework is done using the fake news repository developed from four different web sources. Future work ought to utilize more data from wide sources of fake news outlets. Such data could also be supplemented from social media sources such as Facebook, Twitter, Reddit, etc. While this study was conducted on a single node graphical processing unit (GPU), development of real time analytics approaches that utilize streaming data from social media sources and the capabilities of GPU enabled computing clusters could help with real-time identification of fake news. This study is an important attempt towards curbing the risks imposed by fake news through timely identification using analytics approaches. Additionally, we believe that allowing users to specify the importance of each event by assigning a weight based on their knowledge of the event could further improve our proposed model. A large weight value indicates an event is more important than the other event. This enables an individual's knowledge to be factored into the model.

A majority of news articles are time sensitive, implying that better fake-news detection maybe handled in real-time. To deal with real-time news, models need to incorporate a pre-processing module to process real-time properties of news articles. The development of such real-time pre-processing module is beyond the current scope of this study but is an interesting topic for future research. Such a model will have the capabilities to also handle new and emerging fake news topics.

## 7.2 Summary of RT-FEND

Fake news issue becomes increasingly challenging to tackle due to the prevalence of real-time news data and fast consumption of news by the society. Fake news acts as a negative catalyst in the social media circulating articles, video, and comments. There is a pressing demand to assist people to identify misinformation from massive amount of news data in a timely manner. Ad-hoc models of detecting fake news fall short of providing the promise of curbing fake news spread in a timely manner. To fulfil such a demanding need, we have developed an NLP-based fake news detection system that can be set up in high-performance cluster computing environment of any organization. More specifically, Our proposed system - a real-time fake-news detector runns on Spark computing clusters. Our system offers high efficiency, scalability, and availability during the course of discovering fake news articles.

The proposed system goes beyond the analytic capabilities of other existing Fake news detection models that are predominantly non- real-time in nature. Our proposed approach embraces multifaceted new features. First, the system governs the topic reduction technique - referred to as topic merging - to significantly cutback the valid dimensionality in high-dimensional news datasets. Second, the system's framework includes the embedded functionalities and requirements of a high-performance distributed computing cluster, thereby shortening the knowledge-base construction process. Third, the system seamlessly integrates the memory management techniques, including our novel batch-size optimization that we proposed and test in this study with the Spark-based memory management. With the optimization schemes in place, the system judiciously speeds up the process of detecting fake news articles. Finally, as demonstrated, our system delivers high performance from the aspects of fake-news detection time and accuracy. We quantitatively gauged the performance of the proposed approach driven by four real-world news datasets, namely, *Advocate*, *NaturalNews*, *Politicot*, and *Greenville Gazette*. The extensive empirical studies demonstrate that our approach significantly reduces the number of topics and the number of clusters by up to a

maximum of 25.03% and 30.95% with averages of 19.76% and 26.92%, respectively. Our system is conducive to unraveling the credibility distribution of the four different news datasets. For example, in the *Advocate*, *naturalNews*, *politicot*, and *greenvillegazette* datasets, the percentage of news items with credibility larger than 0.8 is around 1.35%, 6.49% 4.46%, and 2.92%, respectively. Running on a Spark cluster equipped with eight 4GB nodes, the system boosts the efficiency of the standalone system by up to 63.37% with an average of 23.45%. After integrating the novel memory management schemes, the system further improves the detection performance of baseline model by up to 20.3% with an average of 19.45%.

Our proposed model detects fake news in real time and leverages the memory management method for effective functioning of the model have some limitations, which provides directions for future research in detecting fake news. First, news articles with continuous events should be analysed by a fake-news detector, because the authenticity of events of preceding articles may be overturned due to previously covered events if there is a disagreement among articles on the same topic. The proposed model is unable to handle such ambiguity on its own. To tackle this problem, future models could incorporate time stamps of the news articles to track news articles that relate to a continuous event. Second, the model may treat few fake topics or events as real ones leading to classification error; this type of errors are more likely to occur when legitimate news data are conflicting with each other. Future work in this area could consider purging outdated news articles. Third, the other features excluded from the analysis in the model are sentiments, social behaviours, syntax, and sources. Future work could address this concern by designing a sentiment- based model followed by one that can detect social-behaviour. Last, but not least, the topic-merging mechanism implemented in our model is the first step toward detecting fake news from high-dimensional datasets. Our proposed topic-merging scheme is far from perfect. Future work could continue optimizing the model's performance from the perspective of feature reduction.

## 7.3 Summary of OSS

Our ability to identify objectivity within text documents plays a vital role in controlling the rapid propagation of misinformation as this approach could benefit various NLP-based applications such as fact checking, knowledge base construction, information retrieval, AI-driven question-answering systems, and deception detection on social media. This study made a rigorous inquiry into mechanisms that could identify objective and subjective patterns in terms of agents' (i.e, subjects or objects of the sentences) view points (first, second or third, person usage) and tenses from the annotated corpora by leveraging the classical OpenIE techniques.

Separating subjectivity from objectivity is a complex problem due to several challenges. For example, there is no comprehensive dictionary for subjective language. In addition, many subjective expressions have objective usages or vice-versa [127], so development of a comprehensive dictionary alone would not suffice. Recognizing these challenges and the limitation of traditional subjective clues (manifested in the form of words, collocations or phrases) identification approaches such a poor recognition rates, etc., we proposed to manipulate latent features extracted from relational triples (subjects, objects, and predicates) in conjunction with view points and tenses. Because complicated and diverse sentence compositions could generate multiple relational triples, we utilized triples having the largest span to represent sentences with the text documents. More specifically, a triple consists of an initial subject, a corresponding final object, and a subject-object-related predicate. To this end, we propose three novel algorithms to extract features based on sequential order of view point of the subject, an object, and the predicate tense. The forth algorithm evaluates these extracted features to detect objectivity and subjectivity in the input text.

We demonstrate the distributional discrepancies of the three pairs and the triples between the objective and subjective datasets by shuffling the three latent features. Findings suggest that subjects in the first and second view points are more likely to be associated with subjective patterns. On the flip side, subjects that are formed by third view point (e.g.,

pronouns or named-entities) tend to be objective patterns. The experimental results unravel that our approach exhibits significantly high precision, recall and F-score measures on objective patterns evaluation. Delivering high performance in evaluating objective patterns, our model is expected to benefit modern NLP applications, including misinformation detection in social media.

Using our proposed methodologies, future studies could focus on constructing ground-truth knowledge bases to wrestle with textual misinformation over the social media and develop improved models of objective pattern recognition for specific use contexts such as COVID-19 fake new detection, controversy analysis, etc. The proposed methodology has two limitations and subsequent studies could be executed to address these shortcomings. First, the existing OpenIE method fails to recognize individual sentence in certain complex cases. For example, triples are difficult to extract from the sentences where subjects have lengthy attributive expressions. Another example is that of inverted sentences that could not be represented in forms of relational triples. Second, there is the lack of annotated objective and subjective datasets and; therefore, pattern versatility verification becomes a grand challenge.

Bibliography


[1] A. S. Abrahams, W. Fan, G. A. Wang, Z. J. Zhang, and J. Jiao. An integrated text analytic framework for product defect discovery. *Production and Operations Management*, 24(6):975–990, 2015.

[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. J. Passonneau. Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)*, pages 30–38, 2011.

[3] A. K. Al Tamimi, M. Jaradat, N. Al-Jarrah, and S. Ghanem. Aari: automatic arabic readability index. *Int. Arab J. Inf. Technol.*, 11(4):370–378, 2014.

[4] H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research, 2017.

[5] AMPLab. Streaming k-means. `https://github.com/amplab/iolap/blob/master/docs/mllib-clustering.md`, 2016. [Online; accessed December 10, 2015].

[6] G. Angeli, M. J. J. Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 344–354, 2015.

[7] B. Anselment, A. Hasanli, A. Petrova, S. Poluetkov, M. Schneider, M. M. Beutter, A. Duplaa, and C.-M. T. Fuchs. Using nlp for adaptive fact extraction and text summarization. 2020.

[8] J. S. Armstrong. Unintelligible management research and academic prestige. *Interfaces*, 10(2):80–86, 1980.

[9] N. Asher, F. Benamara, and Y. Mathieu. Distilling opinion in discourse: A preliminary study. In *Coling 2008: Companion volume: Posters*, pages 7–10, 2008.

[10] N. Asher, F. Benamara, and Y. Y. Mathieu. Appraisal of opinion expressions in discourse. *Lingvisticæ Investigationes*, 32(2):279–292, 2009.

[11] G. Attardi, V. Basile, C. Bosco, T. Caselli, F. Dell'Orletta, S. Montemagni, V. Patti, M. Simi, and R. Sprugnoli. State of the art language technologies for italian: The evalita 2014 perspective. *Intelligenza Artificiale*, 9(1):43–61, 2015.

[12] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[13] P. Barnaghi, P. Ghaffari, and J. G. Breslin. Opinion mining and sentiment polarity on twitter and correlation between events and sentiment. In *2016 IEEE second international conference on big data computing service and applications (BigDataService)*, pages 52–57. IEEE, 2016.

[14] H. Bast and E. Haussmann. Open information extraction via contextual sentence decomposition. In *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, pages 154–159. IEEE, 2013.

[15] C. Berkeley. Evaluating resources: Scholarly & Popular Sources. `http://guides.lib.berkeley.edu/c.php?g=83917\&p=3747680`, 2018. [Online; accessed 20-Feb-2018].

[16] N. Bhutani, A. Traylor, C. Chen, X. Wang, B. Golshan, and W.-C. Tan. Sampo: Unsupervised knowledge base construction for opinions and implications. *Automatic Knowledge Base Construction (AKBC)*, 2020.

[17] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[18] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.

[19] M. Bonzanini, M. Martinez-Alvarez, and T. Roelleke. Opinion summarisation through sentence extraction: An investigation with movie reviews. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1121–1122, 2012.

[20] C. Bosco, V. Patti, and A. Bolioli. Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE Intelligent Systems*, 28(2):55–63, 2013.

[21] N. Braun, M. Goudbeek, and E. Krahmer. Affective words and the company they keep: Studying the accuracy of affective word lists in determining sentence and word valence in a domain-specific corpus. *IEEE transactions on affective computing*, 2020.

[22] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent systems*, 28(2):15–21, 2013.

[23] I. Chaturvedi, E. Cambria, R. E. Welsch, and F. Herrera. Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. *Information Fusion*, 44:65–77, 2018.

[24] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, and E. Cambria. Bayesian network based extreme learning machine for subjectivity detection. *Journal of The Franklin Institute*, 355(4):1780–1797, 2018.

[25] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min. Statistical features-based real-time detection of drifted twitter spam. *IEEE Transactions on Information Forensics and Security*, 12(4):914–925, 2017.

[26] C. Chen, Z. Wang, and W. Li. Tracking dynamics of opinion behaviors with a content-based sequential opinion influence model. *IEEE Transactions on Affective Computing*, 2018.

[27] J. M. Chenlo and D. E. Losada. An empirical study of sentence features for subjectivity and polarity classification. *Information Sciences*, 280:275–288, 2014.

[28] E. Choi, H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang, and L. Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

[29] G. Cinque. The semantic classification of adjectives: A view from syntax. *Studies in Chinese Linguistics*, 35(1):1–30, 2014.

[30] N. J. Conroy, V. L. Rubin, and Y. Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015.

[31] D. M. Cook, B. Waugh, M. Abdipanah, O. Hashemi, and S. A. Rahman. Twitter deception and influence: Issues of identity, slacktivism, and puppetry. *Journal of Information Warfare*, 13(1):58–71, 2014.

[32] L. Del Corro and R. Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366. ACM, 2013.

[33] A. Deoras, K. Yao, X. He, L. Deng, G. G. Zweig, R. Sarikaya, D. Yu, M.-Y. Hwang, and G. Mesnil. Assignment of semantic labels to a sequence of words using neural network architectures, Sept. 2 2013. US Patent App. 14/016,186.

[34] L. Dormehl. A 19-year-old Stanford student has created a 'fake news detector AI'. `https://www.digitaltrends.com/cool-tech/fake-news-detector-ai/`, 2017. [Online; accessed January 20, 2017].

[35] D. Dueck and B. J. Frey. Non-metric affinity propagation for unsupervised image categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[36] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer, 2006.

[37] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[38] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[39] D. H. Farias and P. Rosso. Irony, sarcasm, and sentiment analysis. In *Sentiment Analysis in Social Networks*, pages 113–128. Elsevier, 2017.

[40] C. Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.

[41] S. Feng, R. Banerjee, and Y. Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 171–175. Association for Computational Linguistics, 2012.

[42] M. Fisher. Who cares if it's true? Modern-day newsrooms reconsider their values. *Columbia Journalism Review*, 52(6):14, 2014.

[43] D. H. Fusilier, M. Montes-y Gómez, P. Rosso, and R. G. Cabrera. Detecting positive and negative deceptive opinions using pu-learning. *Information processing & management*, 51(4):433–443, 2015.

[44] M. Gahirwal, S. Moghe, T. Kulkarni, D. Khakhar, and J. Bhatia. Fake news detection. *International Journal of Advance Research, Ideas and Innovations in Technology*, 4(1):817–819, 2018.

[45] A. Gelbukh. Sentiment analysis and opinion mining: Keynote address. In *Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2017 6th International Conference on*, pages 41–47. IEEE, 2017.

[46] A. Ghose and P. G. Ipeirotis. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE transactions on knowledge and data engineering*, 23(10):1498–1512, 2010.

[47] S. Ghosh and K. Ghosh. Overview of the fire 2016 microblog track: Information extraction from microblogs posted during disasters. In *FIRE (Working Notes)*, pages 56–61, 2016.

[48] J. Golbeck, M. Mauriello, B. Auxier, K. H. Bhanushali, C. Bonk, M. A. Bouzaghrane, C. Buntain, R. Chanduka, P. Cheakalos, J. B. Everett, et al. Fake news vs satire: A dataset and analysis. In *Proceedings of the 10th ACM Conference on Web Science*, pages 17–21. ACM, 2018.

[49] M. Gross. The dangers of a post-truth world, 2017.

[50] A. Guess, B. Nyhan, and J. Reifler. Selective exposure to misinformation: evidence from the consumption of fake news during the 2016 U.S. presidential campaign. Technical report, Dartmouth College, 2018.

[51] J. T. Hancock, M. T. Woodworth, and S. Porter. Hungry like the wolf: A word-pattern analysis of the language of psychopaths. *Legal and criminological psychology*, 18(1):102–114, 2013.

[52] hongbing Yue. Shuanghuanglian - can be used to curb the novel coronavirus. https://web.archive.org/web/20200224082720/http://scitech.people.com.cn/n1/2020/0131/c1007-31566098.html, 2020. [Online; accessed January 31, 2020].

[53] M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1), 2017.

[54] B. D. Horne and S. 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. *arXiv preprint arXiv:1703.09398*, 2017.

[55] J. Howard and S. Ruder. Fine-tuned language models for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[56] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou. Understand short texts by harvesting and analyzing semantic knowledge. *IEEE transactions on Knowledge and data Engineering*, 29(3):499–512, 2017.

[57] A. Iyengar, G. Kalpana, S. Kalyankumar, and S. GunaNandhini. Integrated spam detection for multilingual emails. In *Information Communication and Embedded Systems (ICICES), 2017 International Conference on*, pages 1–4. IEEE, 2017.

[58] S. M. Jang, T. Geng, J.-Y. Q. Li, R. Xia, C.-T. Huang, H. Kim, and J. Tang. A computational approach for examining the roots and spreading patterns of fake news: Evolution tree analysis. *Computers in Human Behavior*, 84:103–113, 2018.

[59] Z. Jin, J. Cao, Y.-G. Jiang, and Y. Zhang. News credibility evaluation on microblog with a hierarchical propagation model. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 230–239. IEEE, 2014.

[60] F. H. Khan, U. Qamar, and S. Bashir. esap: A decision support framework for enhanced sentiment analysis and polarity classification. *Information Sciences*, 367:862–873, 2016.

[61] D. O. Klein and J. R. Wueller. Fake news: A legal perspective. *Journal of Internet Law*, 20(10):1, 6–13, 2017.

[62] R. Y. K. Lau, W. Zhang, and W. Xu. Parallel aspect-oriented sentiment analysis for sales forecasting with big data. *Production and Operations Management*, 27(10):1775–1794, 2018.

[63] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, et al. The science of fake news. *Science*, 359(6380):1094–1096, 2018.

[64] H. Li, A. Gupta, J. Zhang, and N. Flor. Who will use augmented reality? an integrated approach based on text analytics and field survey. *European Journal of Operational Research*, 2018.

[65] Y.-S. Lin, J.-Y. Jiang, and S.-J. Lee. A similarity measure for text classification and clustering. *IEEE transactions on knowledge and data engineering*, 26(7):1575–1590, 2014.

[66] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[67] G. Lopez. Recognizing objective and subjective language. `http://content.nroc.org/DevelopmentalEnglish/unit05/Foundations/recognizing-objective-and-subjective-language.html`, 2020.

[68] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[69] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[70] B. Marr. How much data do we create every day? the mind-blowing stats everyone should read, 2018.

[71] A. Mbaziira and J. Jones. A text-based deception detection model for cybercrime. In *Int. Conf. Technol. Manag*, 2016.

[72] M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282, 2012.

[73] G. R. McMenamin. *Forensic stylistics*. Elsevier Science Ltd, 1993.

[74] O. Michalon, C. Ribeyre, M. Candito, and A. Nasr. Deeper syntax for better semantic parsing. In *Coling 2016*, 2016.

[75] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[76] A. Mitchell. Which news organization is the most trusted? The answer is complicated. `http://www.pewresearch.org/fact-tank/2014/10/30/which-news-organization-is-the-most-trusted-the-answer-is-complicated/`, 2014. [Online; accessed OCTOBER 30, 2014].

[77] A. Montejo-Ráez, E. Martínez-Cámara, M. T. Martín-Valdivia, and L. A. Ureña-López. Ranked wordnet graph for sentiment polarity classification in twitter. *Computer Speech & Language*, 28(1):93–107, 2014.

[78] A. Mullick, S. Maheshwari, S. C., P. Goyal, and N. Ganguly. A generic opinion-fact classifier with application in understanding opinionatedness in various news section. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 827–828, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[79] G. Murray and G. Carenini. Subjectivity detection in spoken and written conversations. *Natural Language Engineering*, 17(3):397–418, 2011.

[80] R. S. Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of General Psychology*, 2(2):175–220, 1998.

[81] B. Nyhan and J. Reifler. When corrections fail: The persistence of political misperceptions. *Political Behavior*, 32(2):303–330, 2010.

[82] OpenSources. OpenSources Professionally curated lists of online sources, available free for public use. `https://docs.google.com/document/d/10eA5-mCZLSS4MQY5QGb5ewC3VAL6pLkT53V_81ZyitM/preview`, 2017. [Online; accessed April 28, 2017].

[83] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics, 2011.

[84] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[85] B. Pang. Movie review data. `http://www.cs.cornell.edu/people/pabo/movie-review-data/otherexperiments.html`, 2012. [Online; accessed April, 2012].

[86] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[87] S. B. Parikh and P. K. Atrey. Media-rich fake news detection: A survey. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 436–441. IEEE, 2018.

[88] E. Pariser. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*. Penguin Books, 2012.

[89] M. C. Pavan, V. G. Dos Santos, A. G. Lan, J. Martins, W. R. Santos, C. Deutsch, P. B. Costa, F. C. Hsieh, and I. Paraboni. Morality classification in natural language text. *IEEE Transactions on Affective Computing*, 2020.

[90] G. Pennycook and D. G. Rand. Who falls for fake news? The roles of analytic thinking, motivated reasoning, political ideology, and bullshit receptivity. *SSRN Electronic Journal*, September, 2017.

[91] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[92] D.-A. Phan, Y. Matsumoto, and H. Shindo. Autoencoder for semisupervised multiple emotion detection of conversation transcripts. *IEEE Transactions on Affective Computing*, 2018.

[93] S. Poria, E. Cambria, and A. Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49, 2016.

[94] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. Association for Computational Linguistics, 2011.

[95] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937, 2017.

[96] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937. Association for Computational Linguistics, 2017.

[97] E. Ravenscraft. B.S. Detector Lets You Know When You're Reading a Fake News Source. `https://lifehacker.com/b-s-detector-lets-you-know-when-youre-reading-a-fake-n-1789084038`, 2016. [Online; accessed November 19, 2016].

[98] E. Riloff and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112, 2003.

[99] E. Riloff, J. Wiebe, and W. Phillips. Exploiting subjectivity classification to improve information extraction. In *AAAI*, pages 1106–1111, 2005.

[100] P. Rosso and L. C. Cagnina. Deception detection and opinion spam. In *A Practical Guide to Sentiment Analysis*, pages 155–171. Springer, 2017.

[101] A. Roy. Pants On Fire: PolitiFact Tries To Hide That It Rated 'True' in 2008 Obamacare's 'Keep Your Health Plan' Promise. `https://www.forbes.com/sites/theapothecary/2013/12/27/in-2008-politifacts-2013-lie-of-the-year-that-you-could-keep-your-\\health-plan-under-obamacare-it-rated-true`, 2013. [Online; accessed December 27, 2013].

[102] V. L. Rubin, Y. Chen, and N. J. Conroy. Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015.

[103] V. L. Rubin, Y. Chen, and N. J. Conroy. Deception detection for news: Three types of fakes. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, ASIST '15, pages 83:1–83:4, Silver Springs, MD, USA, 2015. American Society for Information Science.

[104] V. L. Rubin, N. J. Conroy, Y. Chen, and S. Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of NAACL-HLT*, pages 7–17, 2016.

[105] V. L. Rubin and T. Lukoianova. Truth and deception at the rhetorical structure level. *Journal of the Association for Information Science and Technology*, 66(5):905–917, 2015.

[106] S. Rustamov, E. Mustafayev, and M. Clements. Sentence-level subjectivity detection using neuro-fuzzy models. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 108–114, 2013.

[107] I. Sahu and D. Majumdar. Detecting factual and non-factual content in news articles. In *Proceedings of the Fourth ACM IKDD Conferences on Data Sciences*, CODS '17, pages 17:1–17:12, New York, NY, USA, 2017. ACM.

[108] J. Shin and K. Thorson. Partisan selective sharing: The biased diffusion of fact-checking messages on social media. *Journal of Communication*, 67(2):233–255, apr 2017.

[109] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[110] L. Si and J. Callan. A statistical model for scientific readability. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 574–576. ACM, 2001.

[111] M. Siering, J.-A. Koch, and A. V. Deokar. Detecting fraudulent behavior on crowdfunding platforms: The role of linguistic and content-based cues in static and dynamic contexts. *Journal of Management Information Systems*, 33(2):421–455, 2016.

[112] C. Silverman. Lies, damn lies, and viral content: How news websites spread (and debunk) online rumors, unverified claims, and misinformation. Technical report, Tow Center for Digital Journalism, Columbia Journalism School, Columbia University, New York, NY, 2015.

[113] V. Singh and S. K. Dubey. Opinion mining and analysis: A literature review. In *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)*, pages 232–239. IEEE, 2014.

[114] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[115] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics, 2011.

[116] Staff. List of how many homes each cable network is in as of July 2015. `https://tvbythenumbers.zap2it.com/reference/list-of-how-many-homes-each-cable-network-is-in-as-of-july-2015/`, 2015. [Online; accessed JULY 21, 2014].

[117] A. Stepinski and V. Mittal. A fact/opinion classifier for news articles. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 807–808, New York, NY, USA, 2007. ACM.

[118] J. Swartz. The world wide web's inventor warns it's in peril on 28th anniversary. *USA Today*, 2017.

[119] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*, 2017.

[120] R. Tang, L. Ouyang, C. Li, Y. He, M. Griffin, A. Taghian, B. Smith, A. Yala, R. Barzilay, and K. Hughes. Machine learning to parse breast pathology reports in chinese. *Breast cancer research and treatment*, pages 1–8, 2018.

[121] C. Toprak, N. Jakob, and I. Gurevych. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 575–584, 2010.

[122] M.-F. Tsai and C.-J. Wang. On the risk prediction and analysis of soft information in finance reports. *European Journal of Operational Research*, 257(1):243–250, 2017.

[123] S. Venkatesan, W. Han, and R. Sharman. A response quality model for online health communities. *Thirty Fifth International Conference on Information Systems*, page 28, 2014.

[124] Venkatesan, Srikanth and Han, Wencui and Kisekka, Victoria and Sharman, Raj and Kudumula, Vidyadhar and Jaswal, Hardeep Singh. Misinformation in Online Health Communities'. *WISP 2012 Proceedings*, page 28, 2013.

[125] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, 174:806–814, 2016.

[126] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao. A semantic approach for text clustering using wordnet and lexical chains. *Expert Systems with Applications*, 42(4):2264–2275, 2015.

[127] J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin. Learning subjective language. *Computational linguistics*, 30(3):277–308, 2004.

[128] Wikipedia. Shuanghuanglian - A traditional Chinese medicine. `https://zh.wikipedia.org/wiki/\%E5\%8F\%8C\%E9\%BB\%84\%E8\%BF\%9E\%E5\%8F\%A3\%E6\%9C\%8D\%E6\%B6\%B2`, 2020. [Online;].

[129] T. Wood and E. Porter. The Elusive Backfire Effect: Mass Attitudes' Steadfast Factual Adherence. *Political Behavior*, pages 1–29, 2018.

[130] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):13, 2008.

[131] C. C. Xavier and V. L. S. de Lima. Boosting open information extraction with noun-based relations. In *LREC*, pages 96–100, 2014.

[132] J. Xu and M. Taft. The effects of semantic transparency and base frequency on the recognition of english complex words. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 41(3):904, 2015.

[133] X.-F. Yang and W.-C. Siu. Vehicle detection under tough conditions using prioritized feature extraction with shadow recognition. In *Digital Signal Processing (DSP), 2017 22nd International Conference on*, pages 1–5. IEEE, 2017.

[134] A. Yessenalina, Y. Yue, and C. Cardie. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1046–1056, 2010.

[135] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica. Discretized streams: Fault-tolerant streaming computation at scale. In *Proceedings of the twenty-fourth ACM symposium on operating systems principles*, pages 423–438. ACM, 2013.

[136] C. Zhang, A. Gupta, C. Kauten, A. V. Deokar, and X. Qin. Detecting fake news for reducing misinformation risks using analytics approaches. *European Journal of Operational Research*, 2019.

[137] W. Zhang, C. Bu, T. Yoshida, and S. Zhang. Cospa: A co-training approach for spam review identification with support vector machine. *Information*, 7(1):12, 2016.

[138] Y. Zhao, X. Xu, and M. Wang. Predicting overall customer satisfaction: Big data evidence from hotel online textual reviews. *International Journal of Hospitality Management*, 76:111–121, 2019.