Interpretable Features for Distinguishing Machine Generated News Articles.

by

Ravi Teja Karumuri

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved April 2022 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Steven Corman
Hasan Davalcu

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Social media has become a primary means of communication and a prominent source of information about day-to-day happenings in the contemporary world. The rise in the popularity of social media platforms in recent decades has empowered people with an unprecedented level of connectivity. Despite the benefits social media offers, it also comes with disadvantages. A significant downside to staying connected via social media is the susceptibility to falsified information or Fake News. Easy accessibility to social media and lack of truth verification tools favored the miscreants on online platforms to spread false propaganda at scale, ensuing chaos. The spread of misinformation on these platforms ultimately leads to mistrust and social unrest. Consequently, there is a need to counter the spread of misinformation which could otherwise have a detrimental impact on society. A notable example of such a case is the 2019 Covid pandemic misinformation spread, where coordinated misinformation campaigns misled the public on vaccination and health safety.

The advancements in Natural Language Processing gave rise to sophisticated language generation models that can generate realistic-looking texts. Although the current Fake News generation process is manual, it is just a matter of time before this process gets automated at scale and generates Neural Fake News using language generation models like the Bidirectional Encoder Representations from Transformers (BERT) and the third generation Generative Pre-trained Transformer (GPT-3). Moreover, given that the current state of fact verification is manual, it calls for an urgent need to develop reliable automated detection tools to counter Neural Fake News generated at scale.

Existing tools demonstrate state-of-the-art performance in detecting Neural Fake

News but exhibit a black box behavior. Incorporating explainability into the Neural Fake News classification task will build trust and acceptance amongst different communities and decision-makers. Therefore, the current study proposes a new set of interpretable discriminatory features. These features capture statistical and stylistic idiosyncrasies, achieving an accuracy of 82% on Neural Fake News classification. Furthermore, this research investigates essential dependency relations contributing to the classification process. Lastly, the study concludes by providing directions for future research in building explainable tools for Neural Fake News detection.

ACKNOWLEDGEMENTS

I am incredibly thankful to my advisor, Dr. Huan Liu, for allowing me to pursue my Master's Thesis under his tutelage. Furthermore, I would like to thank my committee members, Dr. Steven Corman and Dr. Hasan Davulcu, for their valuable suggestions and comments. Additionally, I would like to thank my DMML colleagues, especially Tharindu Kumarage, Amrita Bhattacharya, Mansooreh Karimi, Paras Sheth, Faisal Ahlatawi, David Mossallanezad, Anique Tahir, Ujun Jeong, Zeyad Alghamdi, Lu Cheng, Raha Moraffah, and other members of the lab for their critique and support. I would also like to thank all my friends for supporting me through the journey until the very end and making it memorable. Lastly, I would like to thank my family, especially my Mom, for believing in me and making this possible.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

There has been a massive increase in social media usage in recent times. According to the information provided by Statista, global social media users have increased from 2.86 Billion in 2017 to 3.96 Billion in 2022, accounting for a 40% increase in five years. Furthermore, the projected estimate for these numbers seems to climb up to 4.41 Billion in 2025, indicating a rising trend in social media usage. The active engagement of nearly 50% of the world population on social media platforms has made the Fake News problem increasingly relevant. Traditionally, the definition of Fake News includes any piece of fabricated information intentionally created to deceive people. Miscreants on social media platforms, also described as malicious actors by Zellers *et al.* (2020), spread false propaganda and misinformation, misleading people and influencing their opinions.

Pew Research Center, a non-partisan Think Tank, quantified the effects of Fake News spread on society. The studies conducted by Pew Research Center indicated that the number of users that often consumed news on social media increased from 18% in 2016 to 28% in 2019. Furthermore, they also reported that about 64% of U.S. adults were confused about basic facts about contemporary events. About one in four claimed to have shared made-up stories, and about 51% often encountered inaccurate news. In addition to Pew Research Center's detailed insights, the work done by Himelein-Wachowiak *et al.* (2021) highlights the repercussions of Fake News spread during the Covid-19 pandemic. Some of the real-world consequences include a shortage of drugs misinformed as potential safeguards against Covid-19. Additionally,

the propagation of conspiracy theories on online platforms misled people into disregarding WHO guidelines. Lastly, falsified information about vaccines' ineffectiveness led to campaigns against vaccines putting many lives in danger.

Another study conducted by Himelein-Wachowiak *et al.* (2021) demonstrates the repercussions of misinformation spread through online bots during the Covid-19 pandemic. Some of the real-world consequences of spreading fake news included the shortage of the Hydroxychloroquine drug. People believed that the drug would safeguard them against Covid-19 despite having no definitive evidence. Additionally, the belief in conspiracy theories has led people to disregard standard WHO guidelines during the Covid-19 pandemic. Lastly, spreading misinformation about vaccines has affected many people to lose faith in vaccination, thereby putting lives in danger.

Based on these trends, it is fair to claim that fake news has a detrimental impact on society. Therefore, it is crucial to develop robust methods to counter Fake News. Zellers *et al.* (2020) state that the current fact-checking and verification methods are primarily manual to keep the prediction process reliable and transparent. Unfortunately, manual verification processes are not scalable, especially when advancements in Natural Language Processing gave rise to models like BERT (Devlin *et al.* (2018)) and GPT-3 (Brown *et al.* (2020)) that can generate realistic text. The malicious actors can leverage the capabilities of these models to generate misinformation, also called Neural Fake News. The two dangers that Neural Fake News poses are: 1.) Neural Fake News is virtually indistinguishable from real news without external aid 2.) The generation of Neural Fake News can happen at scale.

Accordingly, recent studies in Fake News detection have focused on developing au-

tomated tools to identify and flag Neural Fake News. The progress in generated text detection has also benefitted the Neural Fake News detection problem. A commonality between the two problems is the underlying actor-critic model for generating text and distinguishing it from the original ones. For instance, the work done by Bakhtin *et al.* (2019) discussed the possibilities of utilizing Energy-Based Models (EBMs) for performing generated text detection. The authors used a generator-discriminator model to generate negative samples from a set of human-written samples. Finally, they used a discriminator model to classify text as machine-generated or human-written. They also summarized their findings on the performance of EBMs in multiple settings.

Additionally, Ippolito *et al.* (2019) studied the effects of sampling-based decoding strategies applied during the text generation process in language models. They identified certain statistical artifacts generated during the language generation process and used these artifacts to perform Neural Text classification. Furthermore, Zellers *et al.* (2020) borrow the idea of threat modeling from computer security and re-introduce the Neural News detection problem as a two-player adversarial game containing the adversary and the verifier. They developed the GROVER model containing a generator and discriminator model. The generator model shares architectural similarities with the Open-AI GPT-2 model Radford *et al.* (2019), and the discriminator model constitutes one of the following: BERT, variant of GPT-2, variant of GROVER, FastText. An essential feature of the GROVER generator model is to learn over multiple fields in a news document, such as Body, Author, Date, Headline, and Domain, and generate fabricated articles accordingly. The authors concluded the study by analyzing and comparing automated and human detection performances.

The work done by Schuster *et al.* (2020) took a slightly different approach deviating from the deep learning methods for Neural Fake News detection. Their work derived inspiration from the early deception detection methodologies and uses a stylometric approach for Neural Fake News detection. The authors also provided a benchmark for Neural Fake News detection and discussed the limitations of Stylometry in detecting machine-generated Fake News. They observed that the stylometric approach falls short when the source of both fake and real news is a machine. For instance, in the case of automated journalism, the real news is generated via Natural Language Generation tools. Such articles tend to be closer to Neural Fake News in style because they might use similar decoding strategies for text construction.

In addition to the deep learning and stylometric approaches, Gehrmann *et al.* (2019) proposed GLTR, a visualization tool that can aid humans in detecting generated text. The authors developed three statistical tests that measure the probability of a word, the absolute rank of the word, and the entropy of the predicted distribution given the context. The first two tests determine whether the generated word occurs from the head of the distribution. Furthermore, the last test verifies whether the previously generated context is well known to the prediction system such that it is sure of its next prediction. The authors demonstrated the validity of the statistical tests using a case study. Lastly, the authors conclude by presenting a study on human performance detecting machine-generated text. The authors claim an increase in the detection accuracy of humans from 52% to 72% with help from the GLTR tool.

Existing methods focus primarily on delivering accuracy or interpreting predictions. Moreover, applying discriminatory and explainable statistical features for performing Neural Fake News detection is a first. Accordingly, we propose a new set

of statistical features built from Dependency Parse Trees to perform Neural Fake News Classification. We first assumed that the Dependency Parse Tree structures capture stylistic information from the text. Based on the assumption, we hypothesized that the decoding strategies employed by Neural Language Generation models interfere with the Dependency Parse Tree structure of the text and, therefore, can help distinguish Fake News from Factual News. Accordingly, we developed two types of features: Bag-of-Relations (BoR) and Relation Frequency - Inverse Document Frequency (RFIDF), based on Bag-of-Words and Term Frequency - Inverse Document Frequency features.

Furthermore, we used the features to train Logistic Regression and Random Forest models and observed a maximum accuracy of 82% on the Random Forest model trained on the BoR features. We also identify the critical dependency relations that contribute to the prediction process by examining each feature's importance and SHAP values. Lastly, we presented our insights from the experiments and concluded by discussing future research directions.

Chapter 2

RELATED WORK

The current study bases itself on observations and inferences obtained from various research approaches to solve fake news detection. Based on our literature survey and relevance to our research, we classified past work into three categories: Style, Deep Learning, and Assistance-Based approaches. Each category represents the methodology employed to perform fake news detection. We explore these categories in detail because they provide a foundation for the core idea of our research.

2.1 Style Based

In recent studies, the usage of stylistic differences to identify fake news derives inspiration from the stylometric analysis conducted to detect deception. Some of the early attempts in deception detection relied on using style and language characteristics as features. The work done by Hancock *et al.* (2007) and Vrij *et al.* (2007) used shallow lexico-syntactic cues such as dictionary-based word counting using *Linguistic Inquiry and Word Count* (Pennebaker (1993)) lexicon to identify duplicitous text. Subsequent work done by Mihalcea and Strapparava (2009) , Ott *et al.* (2011) etc., used lexico-syntactic patterns like n-grams, part-of-speech (POS) tags to perform deception detection.

The work done by Feng *et al.* (2012) investigated syntactic stylometry for deception detection. Their research described the feature engineering process using Words, Shallow Syntax, and Deep Syntax. They built the Word features using text's Uni-

gram and Bigram word tokens. Furthermore, they built the Shallow Syntax features using shallow syntactic information like Parts-Of-Speech (POS) tags combined with unigram features. Lastly, the Deep Syntax features encoded the production rules of Probabilistic Context-Free Grammar (PCFG). One of the key observations made by Feng *et al.* (2012) is that the features built out of Context-Free Grammar parse trees consistently improved the detection performance over several baseline methods that were based on shallow lexico-syntactic features. They also demonstrated an improvement on the best-published results on hotel review data of Ott *et al.* (2011) reaching 91.2% accuracy with a 14% error reduction. Additionally, they achieved accuracy up to 85% over the essay data of Mihalcea and Strapparava (2009).

The work done by Schuster *et al.* (2020) discusses the limitations of stylometry for detecting machine-generated fake news. According to the authors, stylometry is typically used for two purposes: (1) to detect the source of text to prevent impersonation or (2) to detect misinformation in the text due to deception. Case (1) focuses on identifying language features that correlate with a specific person or group, and case (2) relies on idiosyncracies of false information to classify misinformation. Additionally, the authors also built a benchmark model for detecting fake news produced by language models based on the truthfulness of the content. They focused on automatic false modifications of truthful news stories to keep them close to the actual content. Lastly, the authors observed that the malicious text generated by a Language Model might be harder to detect than hand-crafted malicious text.

2.2 Deep Learning Based

The ability of Neural Language Generation models to generate realistic text has enabled the development of sophisticated models to study and prevent machine-generated fake news. Recent studies in this domain utilized the Generator-Discriminator architectures to identify fake news. Researchers used current State-of-the-Art language generation models like BERT, GPT-2, and GPT-3 as generators to generate false content. They also used Discriminator models to distinguish between authentic and machine-generated content.

One of the most comprehensive studies conducted in this direction is the work pursued by Zellers *et al.* (2020). The authors defined fake news's scope and described the current state of fact-checking methods in their research. Additionally, the authors developed an adversarial framework for Neural Fake News generation and detection. In the adversarial setting, the authors built Generator and Discriminator models that played the roles of adversary and verifier, respectively. The objective behind creating GROVER was to develop a robust tool to safeguard against impending misinformation threats in the future.

The authors train the Generator model to jointly learn over different fields of Neural Fake News, such as Domain, Date, Authors, Headline, and Body, by modeling the conditional generation of Neural Fake News. For instance, the model generates the body of an article when any of the other fields like Domain, Date, Authors, or Headlines are given as inputs. The authors developed the GROVER Generator model similar to the GPT-2 model in terms of architecture. Furthermore, the authors developed GPT-2, BERT, FastText, and a variant of GROVER, for the Discriminator

to perform the classification task.

Based on the experimental results, the authors observed that a combination of Grover Generator and Grover Discriminator performed best on both unpaired and paired accuracies reaching 99.8% and 100%, respectively. Lastly, the authors draw helpful insights into exposure bias and variance reduction algorithms and their contribution to detecting machine-generated text.

The work done by Ippolito *et al.* (2019) emphasizes the sampling-based decoding strategies that neural language models use in constructing coherent and cogent text. They focus on three sampling strategies: top-K sampling, nucleus sampling, and untruncated random sampling that aid in generating sensible text but introduce statistical oddities that are difficult for a human to notice but easy for automatic detection tools. The main contributions Ippolito *et al.* (2019) made through this study are to firstly provide a comprehensive study of generated text detection systems' sensitivity to model structure, decoding strategy, and excerpt length and, secondly, analyze human rater's ability to detect machine-generated text. Furthermore, they used the web-text dataset and generated the corresponding negative samples using a combination of the GPT-2 output and each decoding strategy for their experiments.

As per the research contributions mentioned earlier, the authors provided insights into automatic detection and human detection. For automated detection, the authors used the BERT model fine-tuned on each dataset variation described previously and observed a maximum accuracy of 88% on the top-k sampling decoding strategy truncated at 40 words. Furthermore, they also observed that the discriminators trained on one decoding strategy did not generalize to the samples obtained from a different

decoding strategy. For the human detection task, the authors observed an overall human performance of about 71% across all sampling methods. However, they noted the best rating accuracy at about 85%, implying that humans had room for improvement to detect machine-generated text better.

The work pursued by Bakhtin *et al.* (2019) explored the idea of using Energy-Based Models (EBMs) to perform generated text detection. Since the EBMs cannot directly mine negative samples using gradient-based methods for text data, the authors resolved this issue by utilizing pre-trained language models to generate negative samples for a given set of human-written text. Furthermore, they trained the EBMs with a Binary Cross-Entropy loss function to output low scores for human-written text and high scores for machine-generated text. In addition to training the EBMs, the authors extensively evaluated the performance of multiple combinations of EBM architectures and corpora in in-domain, cross-architecture, cross-corpus, and unseen settings.

2.3 Assistance Based

In the current section, we explore the research work aimed at enhancing humans' capabilities in detecting machine-generated text. These studies achieved this objective by developing tools that provide valuable insights to humans.

For instance, the work done by Gehrmann *et al.* (2019) in developing a **G**iant **L**anguage model **T**esting Room abbreviated as **GLTR**. The GLTR framework is a visualization tool that displays the statistical artifacts of the machine-generated text. In this research, the authors hypothesized that the contemporary language

models generate excessively from a constricted subset of the true distribution of the natural language in order to keep the generated text coherent and cogent. For instance, sampling strategies like the max sampling, k-max sampling, beam search, and temperature-modulated sampling sampled the next word in the sequence from the head of the distribution. They pick the most probable word to construct the sequence. Consequently, these methods induce sampling bias into the generated text, which the authors exploited to detect and attribute the source of text as human or machine.

The authors presented a case study highlighting the differences between human-written text and machine-generated text by visualizing the rank of the words in the distribution. The authors built an interface to color-code words based on the occurrence in the top-K likely words given the context. Accordingly, they observed higher occurrences of less like words (rarely used words) in human-written text than in machine-generated text. Secondly, the authors also conducted an empirical analysis to validate the features and observed better performance with the GLTR features than with the baseline Bag-of-Words features. Lastly, the authors experimented with a small population of student volunteers and observed an improvement in the detection of generated text from 54% to 72% with no prior training. Accordingly, they argued that an assistive visualization tool with simple statistical tests significantly enhanced machine-generated text's human detection.

Chapter 3

PROBLEM DEFINITION AND RESEARCH FOCI

One of the significant conundrums with Fake News detection is the inconsistent definition of fake news. What constitutes fake news is highly contextual and sensitive to the purpose. According to Zellers *et al.* (2020), fake news exists in multiple forms ranging from Satire to Propaganda, with two broad goals in mind: Monetization and Promotion. Therefore, it is essential to accurately define the constituents of fake news within the scope of our research.

Conventionally, Fake News is any false information aimed to create confusion and mistrust amongst people. However, this definition seems too general and ambitious considering the scope of our research. Therefore, we limit ourselves to neural fake news as described in **Definition 3.0.1**.

Definition 3.0.1 *Any news article generated by a large Natural Language Generation model is defined as* ***Neural Fake News****.*

In the current study, we used a binary supervised classification approach to deal with Neural Fake News classification because we only identify whether a given news article is machine-generated or not. Accordingly, we used datasets containing news articles and the information of their source (human-written or machine-generated) to conduct our experiments. The subsequent sections discuss the hypothesis we developed based on our observations from Chapter 2. Furthermore, we use the hypothesis to build interpretable features for our analysis. Lastly, we note down the challenges

that occur as a consequence of our hypothesis.

3.1 Hypotheses

Circling back to our findings from Chapter 2, we observed that the Exposure Bias and Decoding Strategy are significant factors causing peculiarities in the machine-generated text. Consequently, we first introduce the definition of Exposure Bias and Decoding Strategy below to build our hypotheses:

Definition 3.1.1 ***Exposure Bias*** *arises when an autoregressive generative model uses only ground-truth contexts at training time but generated ones at test time.*

Definition 3.1.2 ***Sampling-based Decoding strategy*** *in a Neural Text Generation model is an algorithm that effectively samples next word in the sequence to generate cohesive text.*

Furthermore, we utilize the following observations about decoding strategies and exposure bias:

- The next word in the sequence is generated from the head of the distribution.
- A combination of Exposure Bias and the quirks of Decoding strategies introduce statistical artifacts that can be used to distinguish human and neural text.

Accordingly, we base our research on the following hypotheses:

- Dependency parse tree structure captures stylistic information.
- Decoding Strategies affect dependency parse tree structures of generated text.
- The statistical features derived from dependency parse tree structures are interpretable.

Chapter 4

METHOD

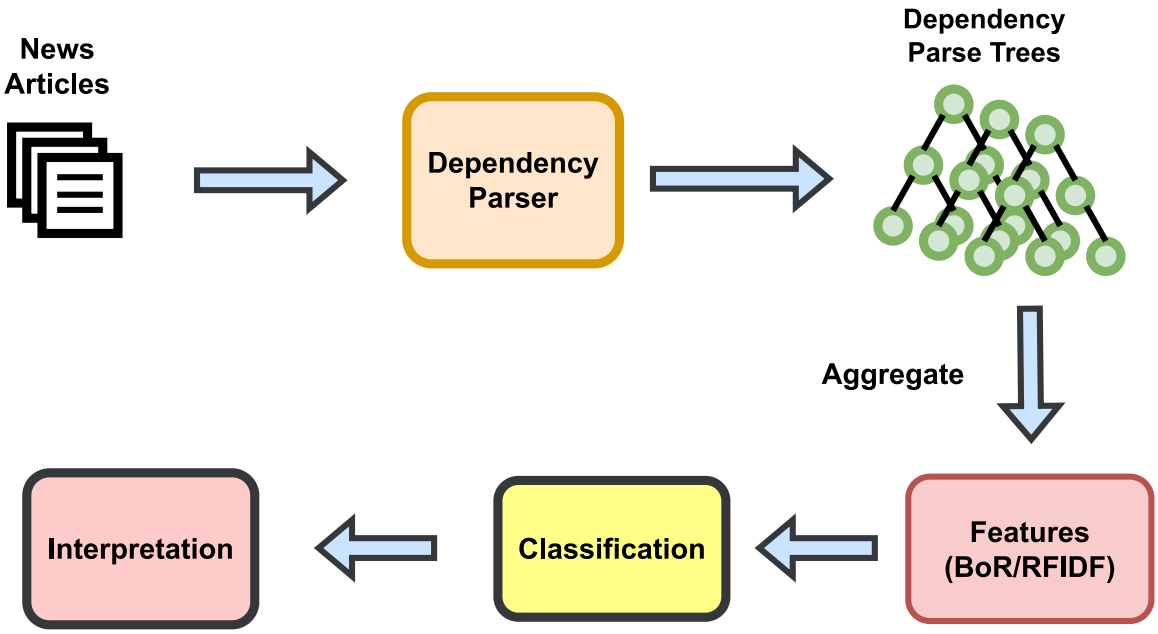


Figure 4.1: Thesis Research Overview.

This chapter discusses the methodology that leverages significant findings from Chapter 2 and Hypotheses from Chapter 3 and addresses the fundamental issue of interpretability. Based on the image shown in Figure 4.1 that briefly summarizes the overview of the current study, we first preprocess the news articles and obtain the dependency parse tree structures. Second, we aggregate these dependency parse trees and construct desired statistical features. Third, we classify the news samples as authentic/fake using the features we built in the second step. Lastly, we use the trained classification models to deduce essential features contributing to the classification process.

In the Feature Engineering section, we first present the motivation and challenge of creating the desired features. Furthermore, we dive deep into the fundamentals of Dependency Parsing and explore the metrics for evaluating the dependency parsing models. We also elaborate on our assumptions while choosing the dependency parsing model. Next, we describe our feature construction process in section 4.1.2, especially the two novel features we developed using the dependency parse trees in the sections 4.1.3 and 4.1.4. Lastly, we look at the classification models we used to classify the news articles in the sections 4.2 and 4.3. We also discuss our approaches to interpreting the critical dependency relationships that contributed to the segregation of fake and factual news articles in sections 4.2, 4.3 and 4.4.

4.1 Feature Engineering

Building explainable discriminatory features for the news articles can be challenging, requiring extensive feature engineering. One of the significant challenges we face while building vector representations for text data is constructing fixed-length feature vectors. Consequently, building feature representations for news articles is no different and can be tricky in our case for three reasons.

Firstly, the number of nodes in a dependency parse tree, a Directed Acyclic Graph (DAG) constituting the words and the relationships between them, is arbitrary. Secondly, the dependency parser works at a sentence level, which means we might get multiple dependency parse trees for a news article. Lastly, the aggregation of the dependency features across all the sentences in an article does not have a conventional representation and requires an appropriate definition.

To address these problems, we developed feature construction algorithms for the two set of features that processes raw news articles and generates meaningful representation for each news article. Section 4.1.2 provides a detailed explanation pertinent to the features and the corresponding procedures for their construction.

4.1.1 Dependency Parser

Since our objective is to build features that are not just discriminatory but also comprehensible, we adopted the idea of using a dependency parser to incorporate the stylistic information into our features. In simple words, a dependency parser is any model that, given a sentence, generates a directed acyclic graph where each node represents a word in the sentence and the edges represent the dependency relationships between the words. We leveraged the dependency relationships between the words to derive statistical features and used them to explain our results. For our research, we followed the dependency relationships standard provided by Choi and Palmer (2012). Before delving into further details, let us first formally define a dependency parsing model as described in Kübler *et al.* (2009):

Definition 4.1.1 *A dependency parsing model consists of a set of constraints* Γ *that define the space of permissible dependency structures for a given sentence, a set of parameters* λ *(possibly null), and fixed parsing algorithm* h*. A model is denoted by* $M = (\Gamma, \lambda, h)$*.*

We also introduce the formal definitions to a dependency graph and dependency parse tree:

Definition 4.1.2 *A dependency graph* $G = (V, A)$ *is a labeled directed graph (digraph) in the standard graph-theoretic sense and consists of nodes,* V *, and arcs,* A*,*

such that for sentence $S = w_0 w_1 ... w_n$ *and label set* R *the following holds:*

1. $V \subseteq \in w_0, w_1, ..., w_n$
2. $A \subseteq V \times R \times V$
3. *if* $(w_i, r, w_j) \in A$ *then* $(w_i, r', w_j) \in A$ *for all* $r' \neq r$

Definition 4.1.3 *A well-formed dependency graph* $G = (V, A)$ *for an input sentence* S *and dependency relation set* R *is any dependency graph that is a directed tree originating out of node* w_0 *and has the spanning node set* $V = V_S$ *. We call such dependency graphs dependency trees.*

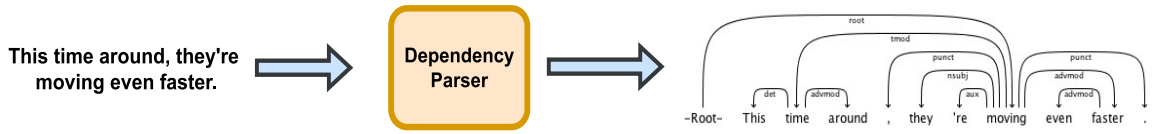


Figure 4.2: Illustration of Dependency Parsing Model.

The literature on the Dependency Parsing domain not only contains a rich set of algorithms and techniques to generate dependency parse trees but also specialized metrics to evaluate dependency parsing. Traditionally, the performance of dependency parsing models has been evaluated based on two metrics, namely the Unlabelled Attachment Score (UAS) and Labelled Attachment Score (LAS). These scores consider the head node (the node from which the relation edge originates) and its dependent node (the node at which the relation edge terminates) and evaluate the relationship's correctness.

According to Nivre and Fang (2017), the LAS and UAS scores measure the percentage of correct head predictions with/without label respectively. The current

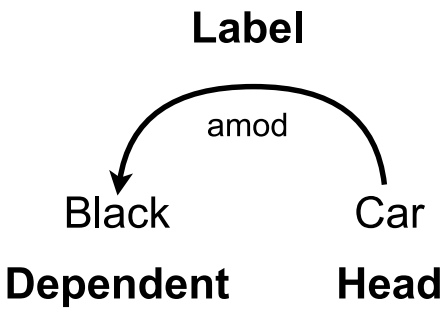


Figure 4.3: Illustration Demonstrating Head and Dependent.

state-of-the-art dependency parsing models use deep learning techniques and score high on the LAS and UAS metrics. Accordingly, we chose the Neural Dependency Parser model built by Dozat and Manning (2016) for our research as it performs on par with the state-of-the-art models.

$$LAS = \frac{\#\ of\ correct\ head\ and\ label\ predictions}{Total\ \#\ of\ predictions} * 100$$

(a) Definition of LAS Metric.

$$UAS = \frac{\#\ of\ correct\ head\ predictions}{Total\ \#\ of\ predictions} * 100$$

(b) Definition of UAS Metric.

Figure 4.4: LAS and UAS Metrics.

Also, we would like to point out that we do not make any assumptions regarding the dependency parsing model we used for our research. Therefore, any model that performs reasonably well on the metrics mentioned above can replace the current dependency model.

4.1.2 Feature Construction

We derived inspiration from the conventional count-based statistical features to build our version of Bag-of-Words (BoW) and Term Frequence-Inverse Document Frequency (TFIDF) features using the dependency relationships between the words.

Adapting to the traditional definitions of Bag-of-Words and TFIDF, we replaced the vocabulary of words with the vocabulary of relationships that can occur in a dependency parse tree. Consequently, we built fixed-length feature vectors from the dependency parse trees as the number of dependency relationships is constant. We define these features as Bag-of-Relationships (BoR) and Relation Frequency-Inverse Document Frequency (RFIDF) and explore them in the sections 4.1.3 and 4.1.4. Reiterating and elaborating further on our Hypotheses mentioned in Chapter 3, we see that Gehrmann *et al.* (2019) demonstrated in their paper that the sampling methods used during neural text generation led the algorithm to sample words frequently from the head of the distribution in order to generate coherent text. Sampling from the head of the distribution limits the vocabulary that the model can choose from and thereby creates statistical inconsistencies. Based on these findings, we hypothesized that the statistical inconsistencies in the text would induce structural differences in a dependency parse tree generated for human written and machine-generated text. We hypothesized that the count-based features defined above would accommodate these differences and aid in the classification task. It is important to note here that we only include the relationships in a dependency parse tree and exclude any kind of word-level information to build our features. The reason for this is that the vocabulary for news articles can be vast, and the absence of a word in the vocabulary can potentially disrupt the feature-building process. Therefore, we decided to consider only the relationships between the words for this research.

4.1.3 *Bag-of-Relations (BoR)*

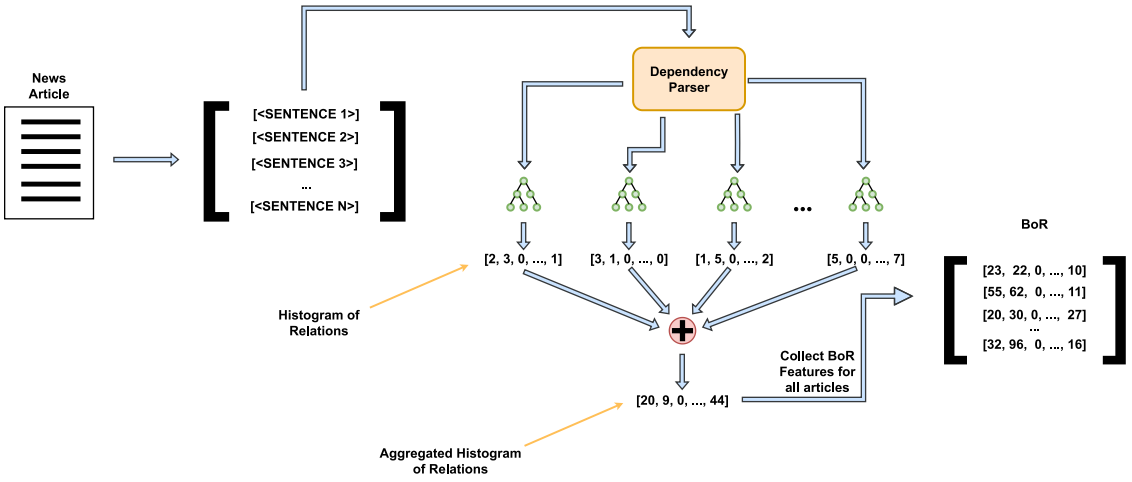


Figure 4.5: Algorithm for Bag-of-Relations Features

To better understand algorithm 1, we first define the vocabulary V, which is the vocabulary of all the relations occurring in a dependency parse tree and $|V|$ refers to the size of the vocabulary. Furthermore, the function *zeros_like* accepts the vocabulary size as an argument and generates a vector of zeros with the vector size equal to the vocabulary size. The $SentenceTokenizer$ function takes the textual content of a news article and splits the text into sentences. Additionally, the $DependencyParsing$ function abstracts the dependency parsing model that generates a dependency parse tree given a sentence. Lastly, the $Count$ function first generates a vector of zeros with the vector size equal to the vocabulary size and updates the count of the relation i in V at the i^{th} index of the vector. The Procedure shown in algorithm 1 is repeated for every news article in the dataset to obtain the BoR matrix shown in Figure 4.5 and algorithm 2.

Algorithm 1 ALGORITHM FOR BAG-OF-RELATIONS FEATURES

procedure BAGOFRELATIONS($news\_article\_text$)
    $input$: $news\_article\_text$
    $output$: BoR_vector
    $BoR\_vector \leftarrow zeros\_like(|V|)$
    $sentences \leftarrow SentenceTokenizer(news\_article\_text)$
    for each $sentence$ **in** $sentences$ **do**
        $relations \leftarrow DependencyParsing(sentence)$
        $relations\_counts \leftarrow Count(relations, |V|)$
        $BoR\_vector \leftarrow BoR\_vector + relations\_counts$
    end for
    return BoR_vector
end procedure

Algorithm 2 ALGORITHM FOR BAG-OF-RELATIONS MATRIX

procedure GETBAGOFRELATIONSMATRIX($news\_articles\_texts$)
    $input$: $news\_articles\_texts$
    $output$: BoR_Matrix
    $BoR\_Matrix \leftarrow [\,]$
    for each $news\_article\_text$ **in** $news\_article\_text$ **do**
        $BoR\_Matrix.append(BagOfRelations(sentence))$
    end for
    return BoR_Matrix
end procedure

4.1.4 *Relation Frequency Inverse Document Frequency (RFIDF)*

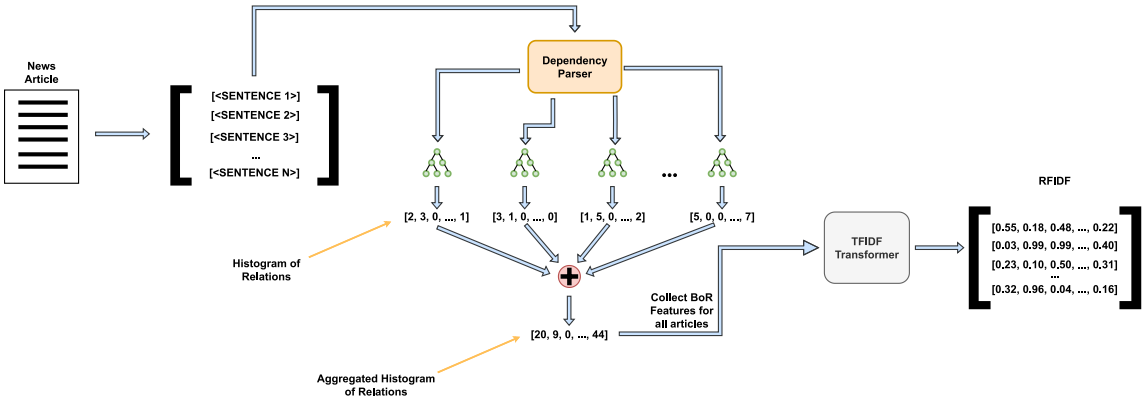


Figure 4.6: Algorithm for Relation-Frequency Inverse-Document-Frequency Features

Algorithm 3 builds on top of algorithm 2 with an extra step that transforms count features into TFIDF features. We define the $TFIDF\_Transformer$ function as a method to compute the TFIDF values given the counts of items in a vocabulary. In this case, we compute the TFIDF values for the relations using the count matrix obtained from algorithm 2 and return an RFIDF matrix.

Algorithm 3 ALGORITHM FOR RFIDF MATRIX

procedure RFIDF($news\_articles\_texts$)
    $input : news\_articles\_texts$
    $output : RFIDF\_Matrix$
    $BoR\_Matrix \leftarrow GetBagOfRelationsMatrix(news\_articles\_texts)$
    $RFIDF\_Matrix \leftarrow TFIDF\_Transformer(BoR\_Matrix)$
    return $RFIDF\_Matrix$
end procedure

4.2 Logistic Regression

A Logistic Regression classifier models linear relationships between the independent variables and the log-odds ratio of the probabilities positive class and negative class. Eq 4.2 displays the mathematical definition of a Logistic Regression model.

$$y = \sigma(\beta * x_1 + \beta * x_2 + \beta * x_3 + ... + \beta * x_n)$$

where the function $\sigma(.)$ is the sigmoid function defined as :

$$\sigma(k) = \frac{1}{1 + e^{-k}}$$

The coefficients of the Logistic Regression model represented by β_i quantify the weights assigned to each feature while classifying data samples. We use these weights to identify essential features contributing to the classification by sorting the weights and picking the top-k corresponding features.

4.3 Random Forests

Restating the definition of Random Forest as described in Breiman (2001), A Random Forest is a classifier consisting of a collection of tree-structured classifiers $h(x, \theta_k), k = 1, ..., N$ where the θ_k are independent identically distributed random vectors, and each tree casts a unit vote for the most popular class at input x.

In a Random Forest model, the Hypothesis function represented as $h(x, \theta_k)$ is typically a Decision Tree classifier. The Random Forest models are trained using the Bagging strategy, which contains two phases: 1.) Bootstrapping the data 2.) Aggregating the results. Accordingly, each decision tree model is independently trained on

bootstrapped data which contains a subset of the original data. The Bagging strategy reduces the chances of overfitting the Decision Tree Classifiers to the data, thereby enhancing the generalization capabilities of the model. To predict the output of a given data sample, the Random Forest model obtains the outputs across all Decision Tree classifiers and aggregates the result to provide the output.

The current study uses impurity-based feature importance scores to extract essential features. The model tracks the drop in the criterion such as Gini Impurity, Entropy, etc., caused by a feature and aggregates and normalizes these drops to derive the feature importance value for a feature.

4.4 SHAP

The SHapley Additive exPlanantions, also known as SHAP, is a model interpretability framework that uses a game-theoretic approach to calculate the feature importance values for a particular prediction. The SHAP framework uses a simpler explanation model $g(z')$, a linear function of binary variables.

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{4.1}$$

where $z' \in \{0,1\}^M$, M is the number of simplified input features, and $\phi_i \in \mathbb{R}$. The simplified input feature space maps to the original feature space using a mapping function h_x.

$$h_x(x') = x \tag{4.2}$$

Furthermore, the Shapley regression values, represented by ϕ_i, correspond to feature importances for a linear model in the presence of multi-collinearity. To compute

the effect a feature i produces on the prediction, two models $f_{S\cup\{i\}}$ and f_S are trained. The former model includes feature i and the latter does not, and the predictions from the two models are compared on the current input through the following equation.

$$diff = f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S) \tag{4.3}$$

Lastly, this difference is computed and averaged across all possible subsets of features through the following equation:

$$\phi_i = \sum_{S\subset F\setminus\{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!}[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \tag{4.4}$$

An interesting feature exhibited by the additive feature attribution methods is a single solution with three desirable properties: Local Accuracy, Missingness, and Consistency. The following describes the definitions for these properties:

- **Local Accuracy:** The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{4.5}$$

- **Missingness:** Missingness constrains features where $x'_i = 0$ to have no attributed impact.

$$x'_i = 0 \Longrightarrow \phi_i = 0 \tag{4.6}$$

- **Consistency:** Let $f_x(z') = f(h(z'))$ and $z' \setminus i$ denote the setting $z'_i = 0$. For any two models f and f', if

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \tag{4.7}$$

for all inputs $z' \in \{0,1\}^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$

Chapter 5

DATA

For the current study, we confine the scope of our problem to just the textual data, specifically data that contains news articles (and their fake counterparts) from online sources. Furthermore, we approach the task of identifying human-written and machine-generated news articles using supervised learning techniques. Therefore, we use Articles and NeuralNews datasets that contain the news articles and the corresponding labels indicating the authenticity of the articles. Table 5.1 contains information about the distribution of classes in the datasets.

Dataset	**# human-written samples**	**# machine-generated samples**	**Total**
Articles	15439	15434	30873
NeuralNews	32000	32000	64000

Table 5.1: Table denoting the number of real and fake news articles in the Articles and NeuralNews dataset.

5.1 The NeuralNews Dataset

The NeuralNews dataset was created by Tan *et al.* (2020) as a benchmarking dataset for the machine-generated news articles detection task. It constitutes a collection of news articles sourced from the GoodNews dataset, which in turn is created from the New York Times articles. Since the GoodNews dataset was initially intended for the image-captioning task, the authors have refactored the contents and developed

the NeuralNews dataset to suit the misinformation detection problem. As a part of remodeling the dataset, the authors first defined four categories as listed below:

- Real Articles and Real Captions
- Real Articles and Generated Captions
- Generated Articles and Real Captions
- Generated Articles and Generated Captions

Second, the authors collected about 32K samples for the real category and generated their fake counterparts using the GROVER model built by Zellers *et al.* (2020). They used the original titles and the article's contents as seed context to generate pertinent fake articles. Third, the authors used real captions from the GoodNews dataset and generated the fake captions using the entity-aware image captioning model built by Biten *et al.* (2019). Lastly, the authors combined the real/fake news articles with real/counterfeit captions. They generated four categories with 32K samples in each leading to 128K samples in total.

To suit our research requirements, we took the NeuralNews dataset and modified it accordingly. Firstly, we omitted the news captions from the dataset as we are only interested in the contents of the news articles. This modification resulted in a dataset containing 64K samples of the real and fake news articles from the NeuralNews dataset. We split the data into train, validation, and test sets, each containing 70%, 10%, and 20% data. Table 5.2 displays information about the number of samples in the real and fake categories in the NeuralNews dataset.

Class	#Train	#Validation	#Test	Total
Fake	22309	2885	6806	32000
Real	22491	2875	6634	32000
Total	44800	5760	13440	64000

Table 5.2: Distribution of Fake and Real samples in NeuralNews dataset.

5.2 The Articles Dataset

The articles dataset was developed on-premise at the Data Mining and Machine Learning lab to serve as a benchmark for the machine-generated text detection task. It was built to extend the NeuralNews dataset and includes news articles broadly classified into topics like Climate Change, Military Ground Vehicles, and Covid-19. Additionally, the news articles can further be sub-categorized into more nuanced issues like Floods, Fires, Military Capabilities, Death tolls, etc. Table 5.3 consists of all the categories and the sub-categories of the topics that constitute the Articles dataset.

Topics	Sub Topics
Climate Change	Floods, Hurricanes, Fires, Agriculture
COVID-19	Death Tolls, Vaccination, Weakened Forces
Military Ground Vehicles	Military Capability, Military Parades

Table 5.3: List of topics and sub topics in the Articles dataset

One of the primary motivations for creating the Articles dataset is to build upon the current datasets for misinformation detection and include content based on recent events in the world, especially significant events like the Covid-19 pandemic in 2019. As a result, we collected the news articles from multiple reputed news outlets

like the BBC, Al-Jazeera, Canadian Dimension, etc., and generated fake counterparts to these articles using the Grover generator model and created a dataset containing about 14K samples of real and fake articles each. Table 5.4 displays information about the number of samples in the real and fake categories in the Articles dataset.

Class	**#Train**	**#Validation**	**#Test**	**Total**
Fake	10106	1013	4320	15439
Real	10090	1007	4337	15434
Total	20196	2020	8657	30873

Table 5.4: Distribution of Fake and Real samples in Articles dataset.

Chapter 6

EXPERIMENTS

We validate our methodology with a series of experiments that objectively analyze the features built from the datasets and their capabilities in explaining the differences between machine-generated and human-written texts. In Experiment 1, we assess the validity of the proposed features by visualizing and analyzing the topology of the data. In Experiment 2, we train the Logistic Regression and Random Forest classification algorithms and classify the news articles as factual or fake. Experiment 3 identifies the crucial features for each classifier using the SHAP framework and the feature importance scores. Lastly, we used the following packages for performing our experiments:

- Supar library for dependency parsing.
- Scikit-Learn for Classification, Clustering.
- Pandas, Numpy, NLTK for data preprocessing.
- Matplotlib, Seaborn for data visualization.

6.1 Evaluation Metrics

Actual/Predicted	Real	Fake
Real	True Positives (TP)	False Negatives (FN)
Fake	False Positives (FP)	True Negatives (TN)

Table 6.1: Confusion matrix for Binary classification

We use the following metrics to evaluate and compare the performance of our models:

- **Precision:**

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:**

$$Recall = \frac{TP}{TP + FN}$$

- **F1:**

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

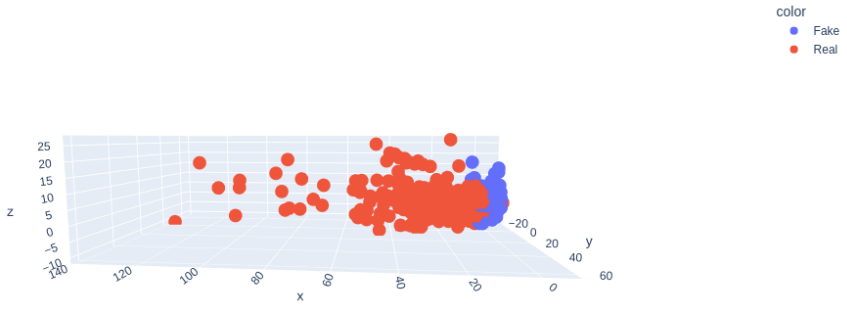
6.2 Experiment 1

This experiment verifies the differences between the Fake and Real data samples from the features we built. We first segregated the data into fake and real samples based on the labels to achieve this objective. After separating the fake and real samples, we obtained the corresponding Bag-of-Words features for the articles and performed Principal Component Analysis. We then obtained the top-3 principal components from the first ten principal components for visualizing the data. Figure 6.1 displays the PCA plots for both NeuralNews and the Articles dataset. We also looked at the explained variances to identify the amount of variance captured by each principal component in Figure 6.2 and noted down the values in Table 6.2.

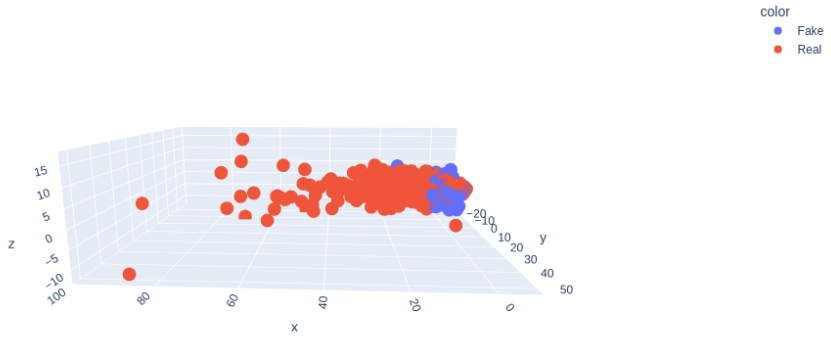
Data	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7	PC_8	PC_9	PC_{10}
NeuralNews	21.14	2.47	1.63	1.40	1.06	1.02	1.01	0.89	0.83	0.82
Articles	20.64	2.32	1.53	1.42	1.14	1.07	0.98	0.92	0.86	0.85

Table 6.2: Top 10 Principal Component of BoR and RFIDF Features

Based on the figure 6.1 that shows the PCA plots of NeuralNews and Articles datasets, we observe that the variance in the real news samples differs from the variance observed in the fake news samples. These plots also demonstrate the discriminatory behavior of the features that can be exploited to classify the news articles.

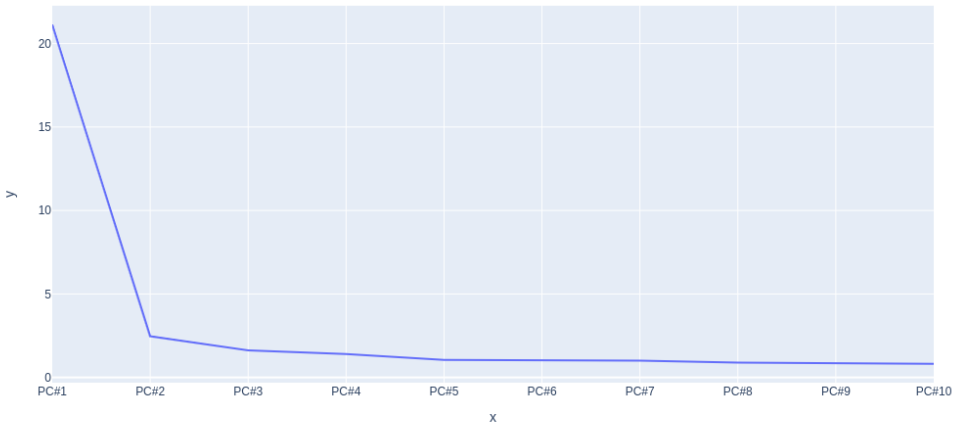


(a) PCA Plot of NeuralNews Dataset with Top-3 Principal Components

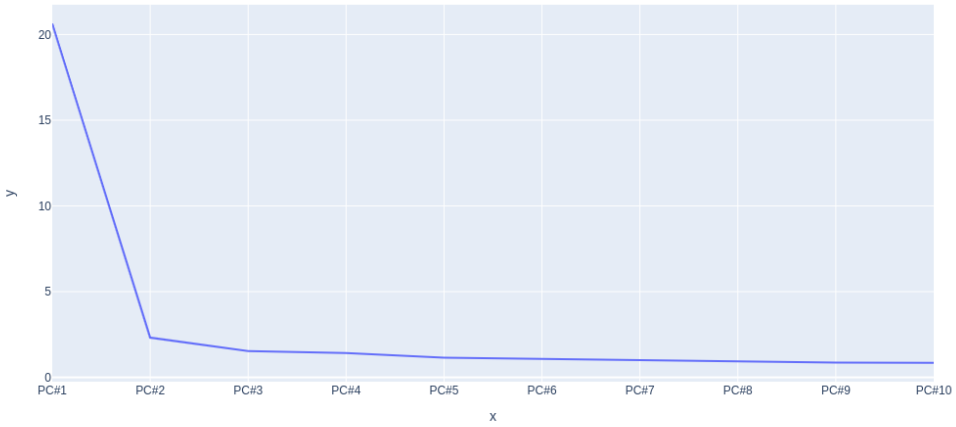


(b) PCA Plot of Articles Dataset with Top-3 Principal Components

Figure 6.1: PCA Plots for NeuralNews and Articles Datasets.



(a) Explained Variance of Principal Components for NeuralNews Dataset



(b) Explained Variance of Principal Components for Articles Dataset

Figure 6.2: Explained Variance Plots for NeuralNews and Articles datasets.

6.3 Experiment 2

In this experiment, we examine the predictive capabilities of the classification models trained on our features to identify fake and authentic news. We first train each combination of the (feature, classifier, dataset) where feature, classifier, and dataset can be one of BoR, RFIDF, Logistic Regression, Random Forest, NeuralNews, Articles respectively. Additionally, we performed a grid search for each feature, classifier, and model combination to arrive at the optimum set of hyperparameters to train our models for the best performance. We kept the default parameters available in the Scikit-Learn package constant during grid search and altered only a limited set of parameters mentioned in the Tables 6.3 and 6.4.

Parameter	**Values**
penalty	none, l1, l2, elasticnet
solver	newton-cg, lbfgs, liblinear
C	1e-5, 1e-4, 1e-3, 1e-2, 1

Table 6.3: Hyperparamters for training Logistic Regression Model.

Parameter	**Values**
n_estimators	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
criterion	Gini, Entropy
max_depth	2, 5, 10, 20, 50
min_samples_leaf	1, 5, 10

Table 6.4: Hyperparameters for training Random Forest Model.

After obtaining the eight combinations from the previous step, we compare the

performances of these models with the baseline model using the F1 metric. We selected a combination of the Logistic Regression model trained on the Trigram Bag-of-Words features on both datasets for the baseline model. Table 6.5 displays the performances of each combination of models we trained along with the baseline model. Lastly, we identified the essential features contributing to the classification process via the feature importance scores/model coefficients. We sorted these scores and picked the features corresponding to the top ten scores. Tables 6.6 and 6.7 show the top ten essential features sorted in descending fashion, appearing left to right in the table.

Data/model	**LR-BoR**	**LR-RFIDF**	**RF-BoR**	**RF-RFIDF**	**Baseline**
NeuralNews	0.79	0.78	0.79	0.78	0.92
Articles	0.77	0.77	0.81	0.80	0.79

Table 6.5: F-1 scores of all the models.

6.4 Experiment 3

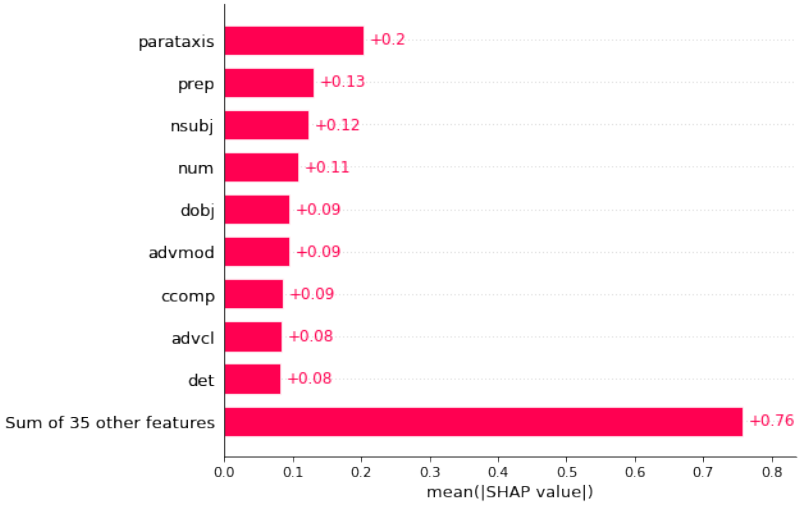
This experiment primarily focuses on the interpretability aspect of the prediction results. Firstly, we examined the feature importance scores/model coefficients of the Random Forest and Logistic Regression models, respectively. We sorted these scores in descending manner and extracted the features corresponding to the top ten scores. Tables show the top ten essential features sorted in descending fashion, appearing left to right. Furthermore, we also used the SHAP framework to identify important features by computing the Shapley values. Figures 6.3, 6.4, 6.5 and 6.6 demonstrate the mean Shapley values of top contributing features.

Model	**Top-10 features**
LR-BoR	conj, pobj, nsubj, dep, ccomp, nn, det, poss, advcl, cop
RF-BoR	punct, nn, prep, pobj, nsubj, det, dobj, appos, ccomp
LR-RFIDF	conj, dep, discourse, advcl, ccomp, cop, acomp, poss, expl, parataxis
RF-RFIDF	punct, appos, nn, ccomp, conj, dep, nsubj, prep, xcomp, aux

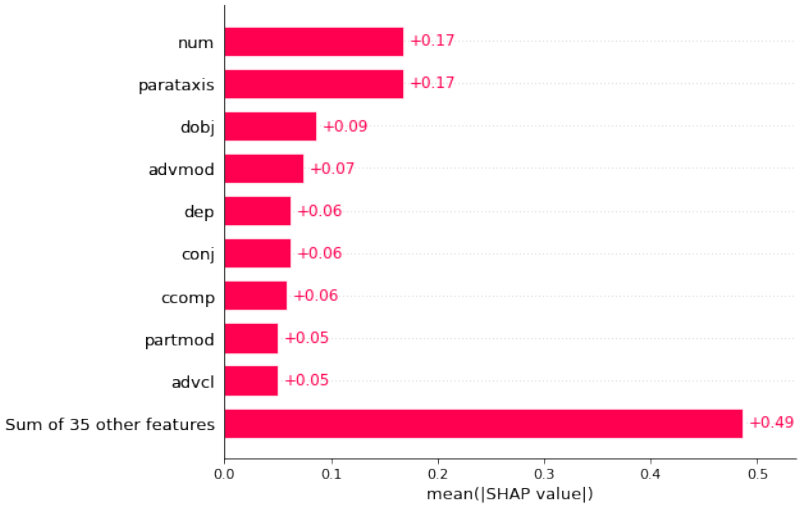
Table 6.6: Top-10 features for each model based on the NeuralNews dataset.

Model	**Top-10 features**
LR-BoR	prep, nsubj, det, punct, aux, nsubjpass, cop, poss, csubjpass, nn
RF-BoR	parataxis, num, prep, det, punct, pobj, dobj, dep, nsubj, nn
LR-RFIDF	appos, cop, neg, expl, poss, cc, det, aux, nsubjpass, number
RF-RFIDF	parataxis, num, dep, advcl, partmod, conj, dobj, advmod, mark, det

Table 6.7: Top-10 features for each model based on the Articles dataset.

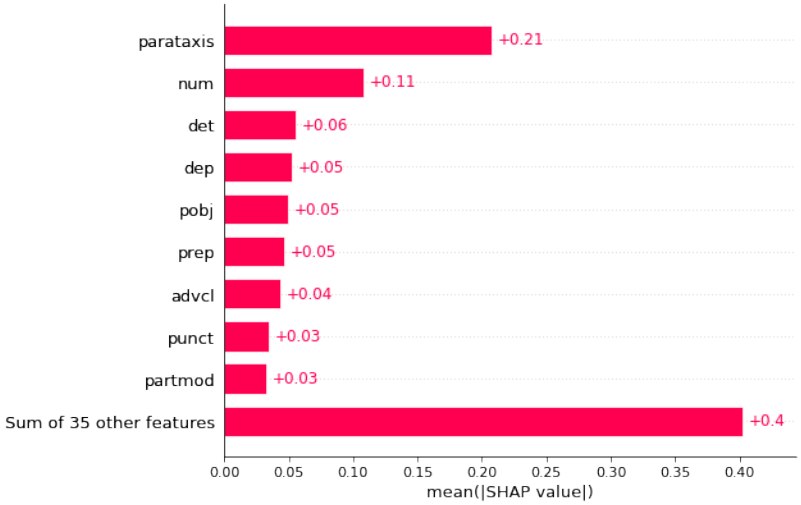


(a) SHAP Values for LR-BoR

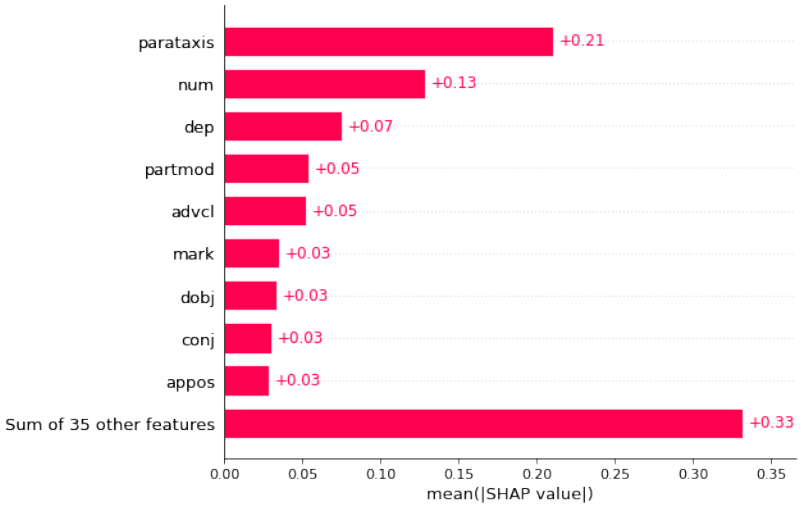


(b) SHAP Values for LR-RFIDF

Figure 6.3: SHAP Values Explaining Important Features With LR model for Articles Dataset.

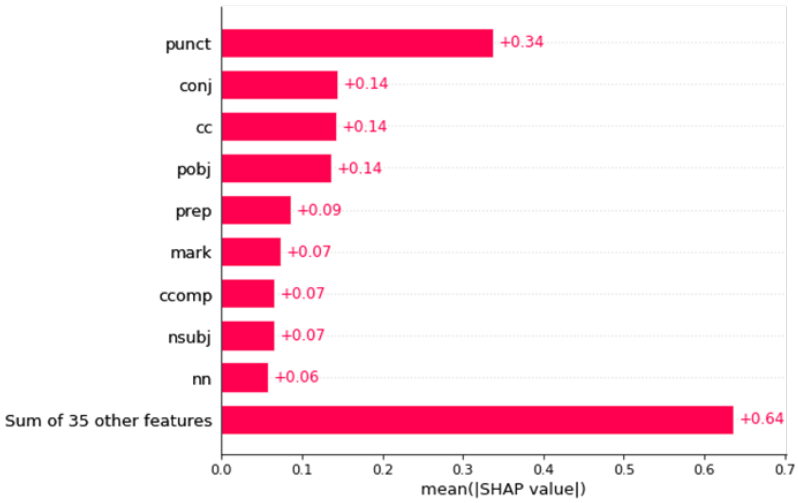


(a) SHAP Values for RF-BoR

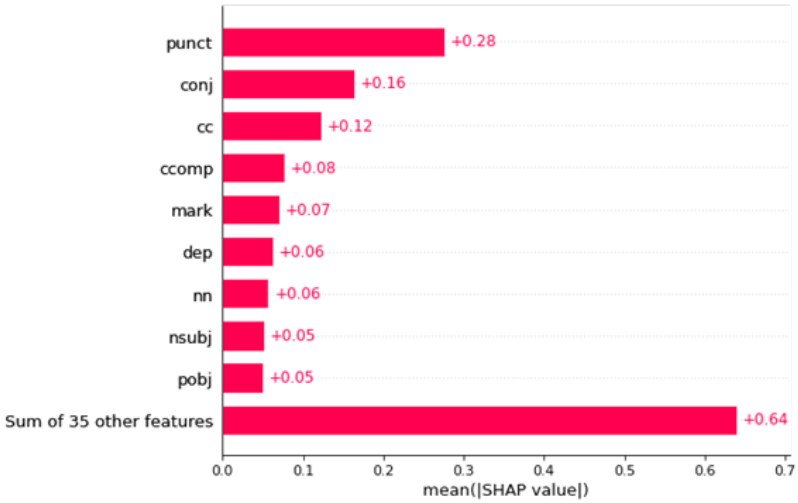


(b) SHAP Values for RF-RFIDF

Figure 6.4: SHAP Values Explaining Important Features With RF model for Articles Dataset.

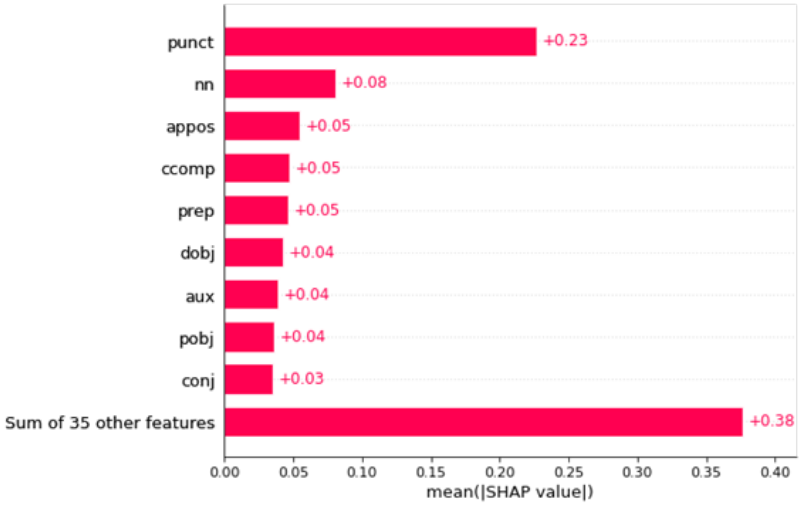


(a) SHAP Values for LR-BoR

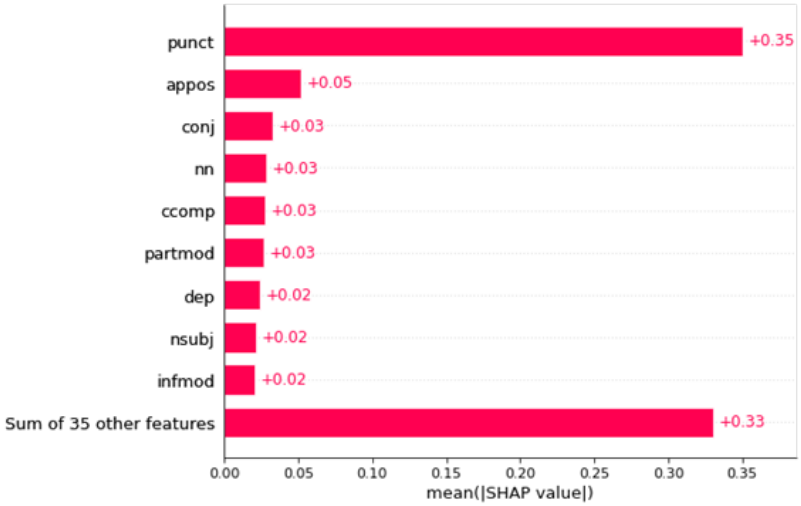


(b) SHAP Values for LR-RFIDF

Figure 6.5: SHAP Values Explaining Important Features With LR model for Neural-News Dataset.



(a) SHAP Values for RF-BoR



(b) SHAP Values for RF-RFIDF

Figure 6.6: SHAP Values Explaining Important Features With RF model for Neural-News Dataset.

Chapter 7

RESULTS

We observed from Experiment 1 that the Random Forest model trained on Bag-of-Relations performed the best amongst all the models on both NeuralNews and Articles datasets. Furthermore, the Logistic Regression and Random Forest models trained on both BoR and RFIDF performed better on the Articles dataset than the baseline models. On the contrary, our models do not beat the baseline models for the Neural News dataset. We investigated this issue by plotting the first three principal components of the BoW and TFIDF features in Figures 7.1 and 7.2 and observed minimal variance for the Real samples group. We theorized the cause for compact packing of Real samples to be a similar writing style, which was true because all the real samples in NeuralNews articles were collected using the NYTimes API.

On the other hand, we observed that the real samples are more spread out in the TFIDF feature space resulting in lower accuracy scores. Explaining this phenomenon requires further analysis and will be conducted in future research. On the other hand, we compensate for this shortcoming by adding the interpretability aspect to the features. For instance, the Trigram feature set is far from providing the kind of insights obtained from Experiment 3. Therefore, we decided to proceed with the proposed features despite the discrepancy.

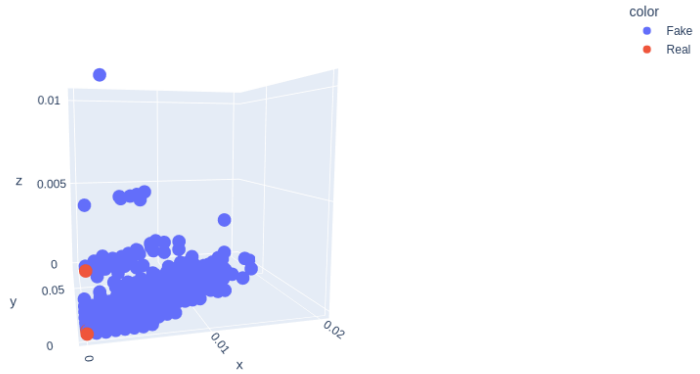


Figure 7.1: First Three Principal Components of Bag-of-Words Features for NeuralNews Dataset.

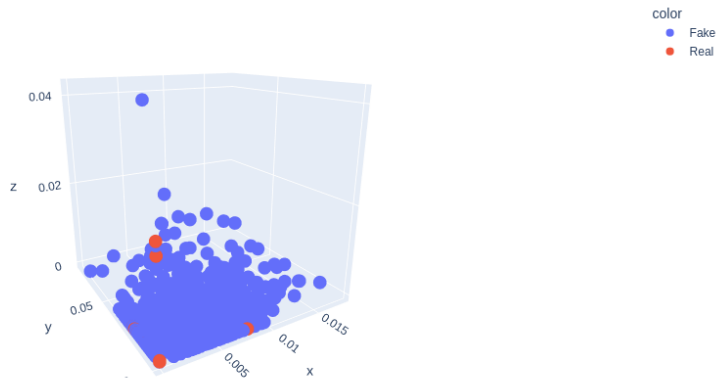


Figure 7.2: First Three Principal Components of TFIDF features for NeuralNews Dataset.

Chapter 8

CONCLUSION AND FUTURE WORK

We conclude our research by reiterating our contributions. Firstly, we demonstrated the feasibility of utilizing dependency parse trees to build statistical features and use them to classify neural fake news articles. Secondly, we provided insights into essential features by analyzing the crucial relations in dependency parse trees that contributed to the classification process.

Despite fulfilling the core objectives of our research, we believe we can improve the current study to address its shortcomings and better understand the neural fake news classification process. For instance, current research recognizes the relationships from the dependency parse tree as individual entities. Instead, we can try incorporating the structural information from the dependency parse tree into the feature construction process. Including the structural information will provide valuable insights, such as the distinctive sub-structures in dependency parse trees that can help distinguish neural fake news from actual news better. For instance, using sophisticated representation learning methods like Graph Neural Networks to learn structural information is a desirable future direction.

In addition to learning better features, the current set of experiments can be expanded to new datasets and modern news articles to verify the robustness of the features. Seeing how stylistic features can fall short while segregating fake and factual news content generated solely by machines, we can expand the current research scope and develop a better set of features that can tackle this variant of fake news detection.

We can also extend the current study to include Deep Neural Network models and alternate interpretable methods like Saliency maps to interpret the predictions. Using BERT to classify fake news can help improve the accuracy but might hinder the interpretability aspect as the context of BERT is limited to 512 tokens. The datasets used in the current set of experiments have multiple news articles spanning beyond the context size of BERT. We could instead use the Longformer model that can encode long documents. Lastly, we can create customized datasets using current news articles and analyze the model performances on the set of features described in this study.

REFERENCES

Adelani, D. I., H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi and I. Echizen, "Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection", arXiv preprint arXiv:1907.09177 (2019).

Bakhtin, A., S. Gross, M. Ott, Y. Deng, M. Ranzato and A. Szlam, "Real or fake? learning to discriminate machine from human generated text", arXiv preprint arXiv:1906.03351 (2019).

Barthel, M., A. Mitchell and J. Holcomb, "Many americans believe fake news is sowing confusion", (2016).

Biten, A. F., L. Gomez, M. Rusinol and D. Karatzas, "Good news, everyone! context driven entity-aware captioning for news images", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 12466–12475 (2019).

Bouygues, H. L., "What happens when robots make fake news?", Forbes. https://www.forbes.com/sites/helenleebouygues/2021/07/15/what-happens-when-robots-make-fake-news/?sh=47e228557453 (2021).

Breiman, L., "Random forests", Machine learning **45**, 1, 5–32 (2001).

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, "Language models are few-shot learners", in "Advances in Neural Information Processing Systems", edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin, vol. 33, pp. 1877–1901 (Curran Associates, Inc., 2020), URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Choi, J. D. and M. Palmer, "Guidelines for the clear style constituent to dependency conversion", Center for Computational Language and Education Research, University of Colorado Boulder, Institute of Cognitive Science, Technical Report **1** (2012).

Dathathri, S., A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski and R. Liu, "Plug and play language models: A simple approach to controlled text generation", arXiv preprint arXiv:1912.02164 (2019).

Department, S. R., "Number of social network users worldwide from 2017 to 2025", Statista. https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/ (2022).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805 (2018).

Dozat, T. and C. D. Manning, "Deep biaffine attention for neural dependency parsing", arXiv preprint arXiv:1611.01734 (2016).

Feng, S., R. Banerjee and Y. Choi, "Syntactic stylometry for deception detection", in "Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)", pp. 171–175 (2012).

Gehrmann, S., H. Strobelt and A. M. Rush, "Gltr: Statistical detection and visualization of generated text", arXiv preprint arXiv:1906.04043 (2019).

Hancock, J. T., L. E. Curry, S. Goorha and M. Woodworth, "On lying and being lied to: A linguistic analysis of deception in computer-mediated communication", Discourse Processes **45**, 1, 1–23 (2007).

Himelein-Wachowiak, M., S. Giorgi, A. Devoto, M. Rahman, L. Ungar, H. A. Schwartz, D. H. Epstein, L. Leggio and B. Curtis, "Bots and misinformation spread on social media: Implications for covid-19", J Med Internet Res **23**, 5, e26933, URL `https://www.jmir.org/2021/5/e26933` (2021).

Ippolito, D., D. Duckworth, C. Callison-Burch and D. Eck, "Automatic detection of generated text is easiest when humans are fooled", arXiv preprint arXiv:1911.00650 (2019).

Jaiswal, S., "Natural language processing — dependency parsing", URL `https://towardsdatascience.com/natural-language-processing-dependency-parsing-cf094bbbe3f7` (2021).

Karimi, H. and J. Tang, "Learning hierarchical discourse-level structure for fake news detection", arXiv preprint arXiv:1903.07389 (2019).

Kübler, S., R. McDonald and J. Nivre, "Dependency parsing", Synthesis lectures on human language technologies **1**, 1, 1–127 (2009).

Lerman, R., "Vaccine hoaxes are rampant on social media. here's how to spot them", Washington Post. https://www. washingtonpost. com/technology/2020/12/18/faq-coronavirus-vaccine-misinformation (2020).

McGill, A., "Have twitter bots infiltrated the 2016 election?", URL `https://www.theatlantic.com/politics/archive/2016/06/have-twitter-bots-infiltrated-the-2016-election/484964/` (2016).

Mihalcea, R. and C. Strapparava, "The lie detector: Explorations in the automatic recognition of deceptive language", in "Proceedings of the ACL-IJCNLP 2009 conference short papers", pp. 309–312 (2009).

Nivre, J. and C.-T. Fang, "Universal dependency evaluation", in "Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)", pp. 86–95 (2017).

Ott, M., Y. Choi, C. Cardie and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination", arXiv preprint arXiv:1107.4557 (2011).

Pennebaker, J. W., "Putting stress into words: Health, linguistic, and therapeutic implications", Behaviour research and therapy **31**, 6, 539–548 (1993).

Pennebaker, J. W., R. L. Boyd, K. Jordan and K. Blackburn, "The development and psychometric properties of liwc2015", Tech. rep. (2015).

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners", OpenAI blog **1**, 8, 9 (2019).

Reis, J. C., A. Correia, F. Murai, A. Veloso and F. Benevenuto, "Explainable machine learning for fake news detection", in "Proceedings of the 10th ACM conference on web science", pp. 17–26 (2019).

Schuster, T., R. Schuster, D. J. Shah and R. Barzilay, "The limitations of stylometry for detecting machine-generated fake news", Computational Linguistics **46**, 2, 499–510 (2020).

Shearer, E. and A. Mitchell, "News use across social media platforms in 2020", (2021).

Tan, R., B. A. Plummer and K. Saenko, "Detecting cross-modal inconsistency to defend against neural fake news", arXiv preprint arXiv:2009.07698 (2020).

Van der Maaten, L. and G. Hinton, "Visualizing data using t-sne.", Journal of machine learning research **9**, 11 (2008).

Vrij, A., S. Mann, S. Kristen and R. P. Fisher, "Cues to deception and ability to detect lies as a function of police interview styles", Law and human behavior **31**, 5, 499–518 (2007).

Zellers, R., A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner and Y. Choi, "Defending against neural fake news", Neurips (2020).

Zhou, X. and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities", ACM Computing Surveys (CSUR) **53**, 5, 1–40 (2020).