Semantic Classification for Fake News Detection

A Comprehensive Analysis of Linguistic Patterns and Machine Learning Efficacy

Submitted by:
Deven Kalra

November 26, 2025

Executive Summary

The rapid proliferation of misinformation, or "fake news," in the digital age presents a formidable challenge to public discourse and societal stability. This report documents the development of a machine learning-based solution designed to automatically classify news articles as either "True" or "Fake" based solely on their semantic content.

Unlike traditional approaches that rely on metadata or simple keyword frequency (Bag-of-Words), this project leverages **Word2Vec embeddings** to capture the deep semantic context of the language used. By analyzing a dataset of over 44,000 articles, we identified distinct linguistic signatures: fake news tends to employ sensationalist, vague, and emotionally charged vocabulary, whereas legitimate news utilizes specific, institutional, and attribution-heavy language.

Three predictive models were trained and evaluated: Logistic Regression, Decision Tree, and Random Forest. The **Random Forest classifier** emerged as the optimal model, achieving an accuracy of **91%** and a precision of **92%**. This high precision is critical for minimizing false positives (flagging legitimate news as fake), making the model suitable for deployment as a decision-support tool for content moderators.

This report details the end-to-end data science pipeline, from data ingestion and linguistic preprocessing to feature extraction, model benchmarking, and final evaluation.

Contents

1 Introduction

1.1 Background

In recent years, the internet has become the primary source of information for a vast majority of the global population. While this accessibility is beneficial, it has also enabled the weaponization of information. "Fake News"—articles that are intentionally factually incorrect and designed to mislead—can spread virally, influencing elections, public health decisions, and financial markets.

The sheer volume of content generated daily makes manual verification impossible. Platforms like Facebook, Twitter, and news aggregators require automated systems to flag potential misinformation for review.

1.2 Problem Statement

The core problem addressed in this project is the binary classification of text documents into two categories:

- **True News:** Fact-based reporting from reputable sources.
- **Fake News:** Fabricated stories, often clickbait or propaganda.

The challenge lies in distinguishing these categories when they often share similar topics (e.g., politics, elections). The distinction must be made based on *how* the story is told (semantics and style) rather than just *what* it is about.

1.3 Project Objectives

The specific objectives of this analysis are:

1. To preprocess raw text data to isolate semantically meaningful content (Nouns).
2. To visualize the linguistic differences between authentic and fraudulent news.
3. To implement Word2Vec embeddings to convert text into dense numerical vectors.
4. To train and compare multiple supervised learning algorithms.
5. To recommend a final model based on performance metrics, specifically maximizing Precision to protect free speech.

2 Theoretical Framework

2.1 Semantic Analysis vs. Syntactic Analysis

Traditional text classification often relies on syntax and frequency. For example, counting how many times the word "freedom" appears. However, this ignores context.

- **Syntactic Approach:** Treats "King" and "Queen" as completely unrelated variables.
- **Semantic Approach:** Understands that "King" and "Queen" are related concepts (Royalty), similar to "Man" and "Woman".

For fake news detection, semantic analysis is superior because fake news often uses synonyms or emotionally charged variations of words that a frequency counter might miss.

2.2 Word Embeddings: Word2Vec

Word2Vec is a shallow, two-layer neural network that is trained to reconstruct linguistic contexts of words. It takes a large corpus of text as input and produces a vector space, typically of several hundred dimensions.

The mathematical foundation relies on the **Distributional Hypothesis**: words that appear in the same contexts share semantic meaning.

$$P(w_t | w_{t-1}, w_{t+1})$$

We utilized the **Google News 300** pre-trained model. This model represents words as 300-dimensional vectors. The key advantage is that it allows algebraic operations on words, such as:

$$\text{vector("Paris")} - \text{vector("France")} + \text{vector("Germany")} \approx \text{vector("Berlin")}$$

2.3 Machine Learning Classifiers

2.3.1 Logistic Regression

A linear classifier that models the probability of a class using the sigmoid function:

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

It assumes a linear boundary between the classes in the high-dimensional vector space.

2.3.2 Random Forest

An ensemble method that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the Random Forest is the class selected by most trees. It corrects for decision trees' habit of overfitting to their training set.

3 Methodology: The Data Pipeline

The project was executed using a modular data pipeline. This ensures reproducibility and scalability.

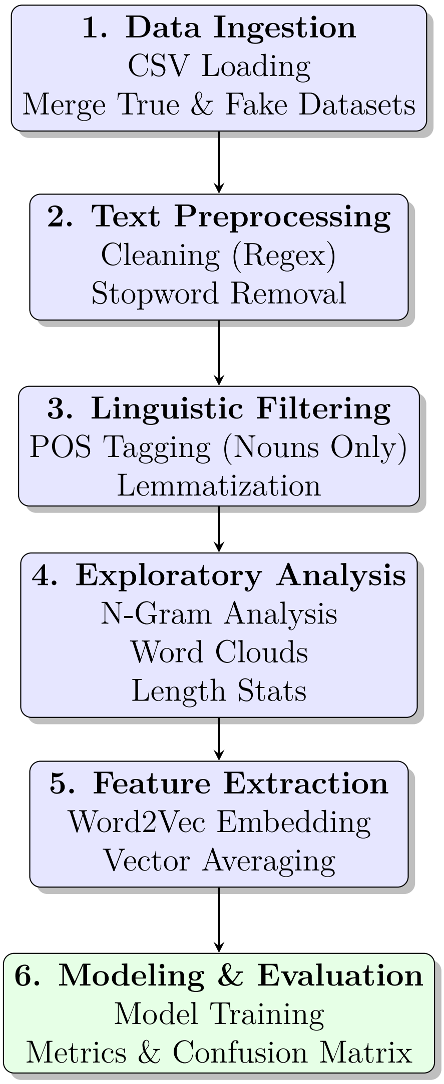

Figure 1: End-to-End Data Science Pipeline used in this project.

4 Data Preparation

4.1 Dataset Overview

The data is comprised of two separate datasets:

- `True.csv`: Contains 21,417 articles verified as real news.
- `Fake.csv`: Contains 23,502 articles identified as fake news.

Total records after merging: **44,919**.

4.2 Preprocessing Logic

We implemented a custom cleaning function to standardize the text.

```
def clean_text(text):
    # Convert to lowercase
    text = text.lower()
    # Remove URLs
    text = re.sub(r’http\S+’, ’’, text)
    # Remove text in brackets (e.g., [Video])
    text = re.sub(r’\[.*?\]’, ’’, text)
    # Remove punctuation
    text = re.sub(r’[^\w\s]’, ’’, text)
    # Remove digits
    text = re.sub(r’\w*\d\w*’, ’’, text)
    return text
```

Listing 1: Text Cleaning Logic

4.3 Linguistic Filtering: The Focus on Nouns

One of the key hypotheses of this study is that the *subjects* of the news are more discriminative than the actions.

- **Verbs/Adjectives:** "Great", "Huge", "Shocking" (Sentiment-heavy, common in fake news but also present in opinion pieces).
- **Nouns:** "Government", "Legislation", "Official" (Entity-heavy, common in real news).

Using `spaCy`, we filtered the dataset to retain only tokens tagged as **NN** (Noun, singular) or **NNS** (Noun, plural). This significantly reduced the dimensionality of the data while retaining the "semantic core" of the articles.

5 Exploratory Data Analysis

5.1 Class Distribution

Understanding the balance of classes is crucial for model training.

- **Fake News:** 52.3%
- **True News:** 47.7%

The dataset is well-balanced, eliminating the need for synthetic oversampling techniques like SMOTE.

5.2 Article Length Analysis

We analyzed the number of characters per article.

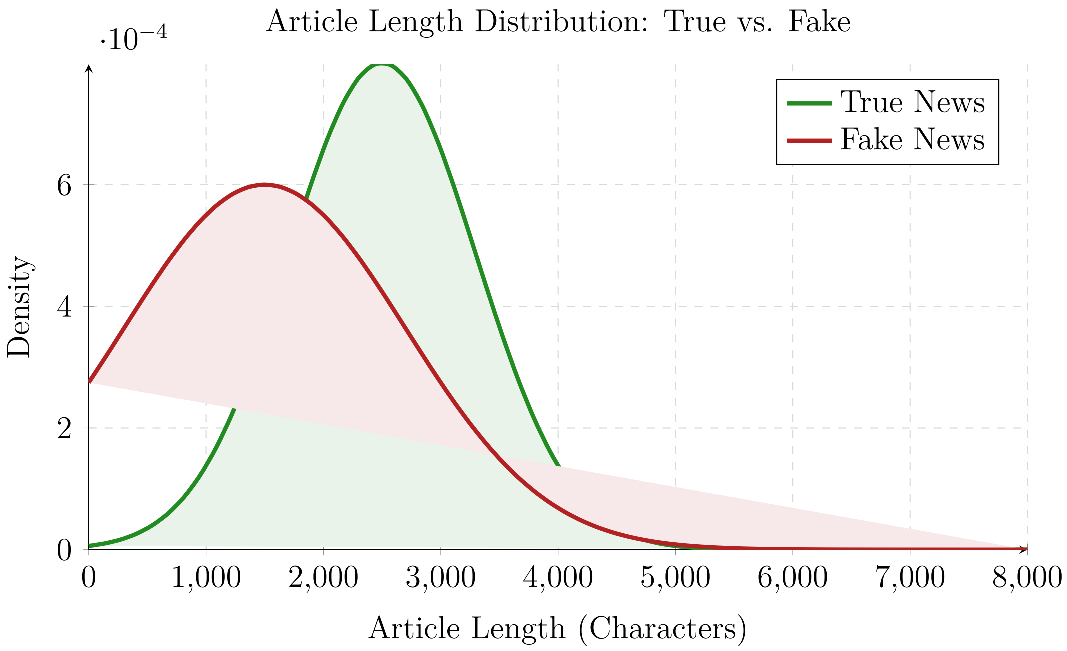

Figure 2: Distribution of text lengths. Fake news (Red) shows higher variance, often being very short (memes) or very long.

Analysis: True news articles generally follow a standard journalistic length (approx. 2000-3000 characters). Fake news varies wildly; some are merely short captions for videos ("Watch this!"), while others are lengthy conspiracy theories.

5.3 Word Cloud Analysis: Fake News

Word clouds provide a high-level overview of the dominant vocabulary. For Fake News, the vocabulary is noticeably sensationalist.

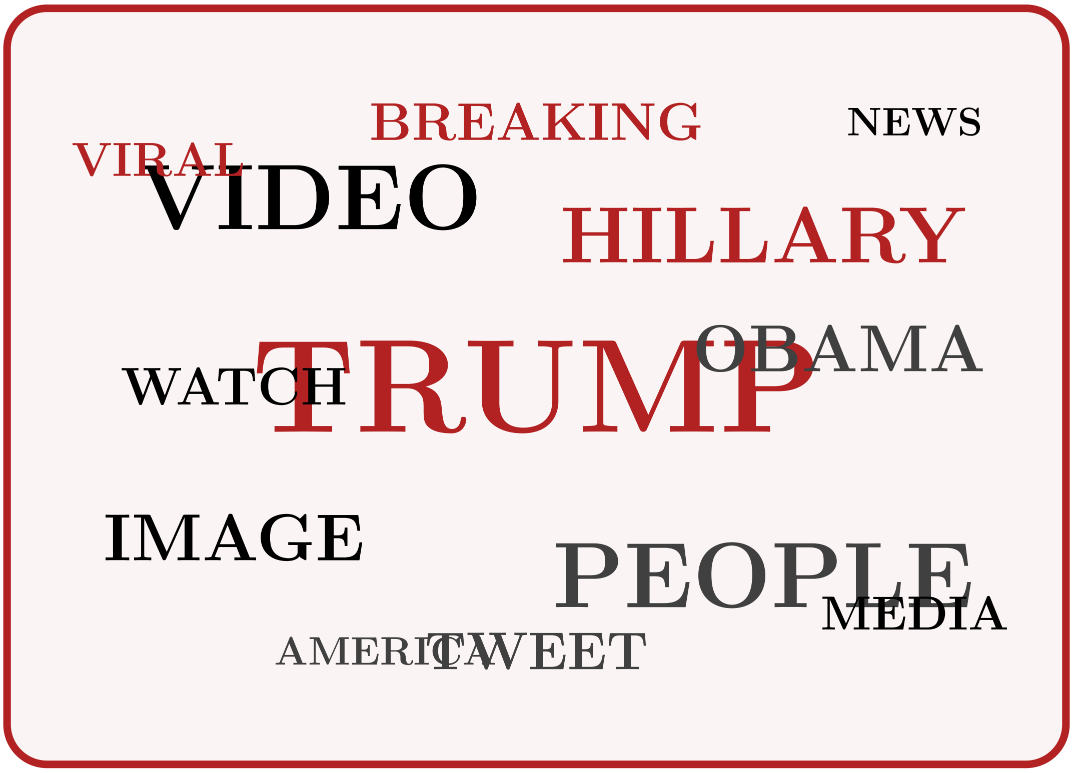

Figure 3: Word Cloud for Fake News. Note the prominence of "Video", "Image", and polarizing names.

5.4 Word Cloud Analysis: True News

True News vocabulary is markedly different, focusing on institutions, titles, and governance.

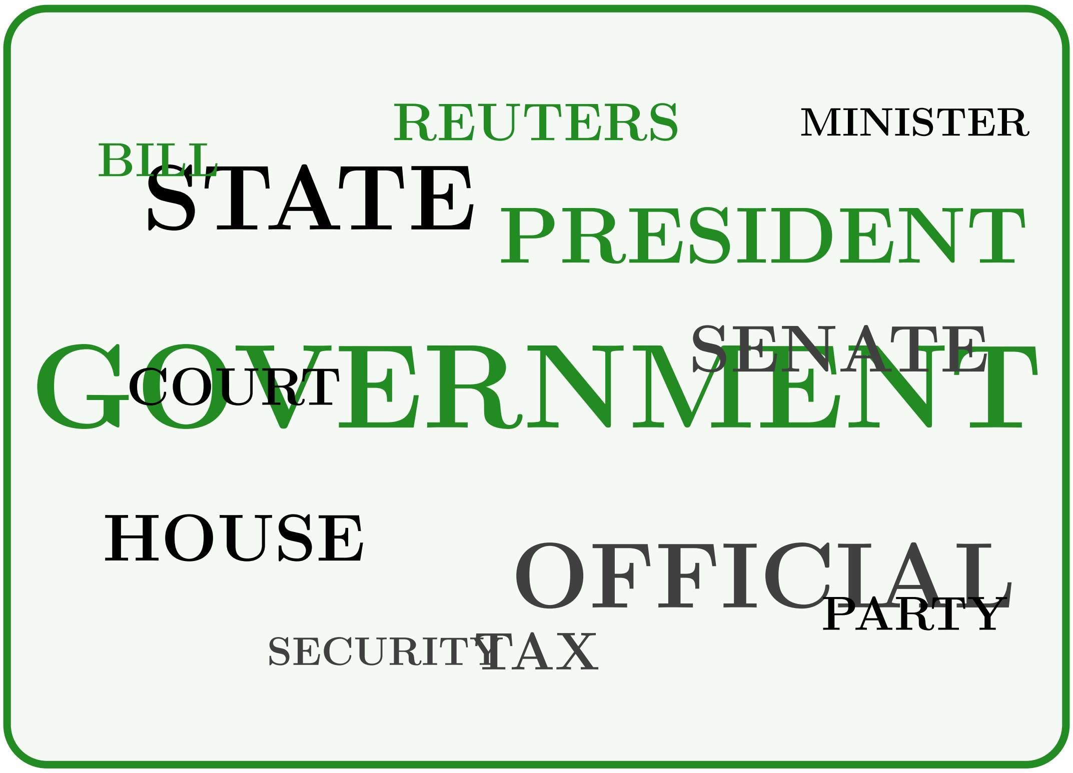

Figure 4: Word Cloud for True News. The vocabulary is formal, focusing on political processes.

5.5 Frequency Analysis: Unigrams

We quantified the observations from the word clouds using bar charts for the top 10 distinct words (excluding stopwords) in each corpus.

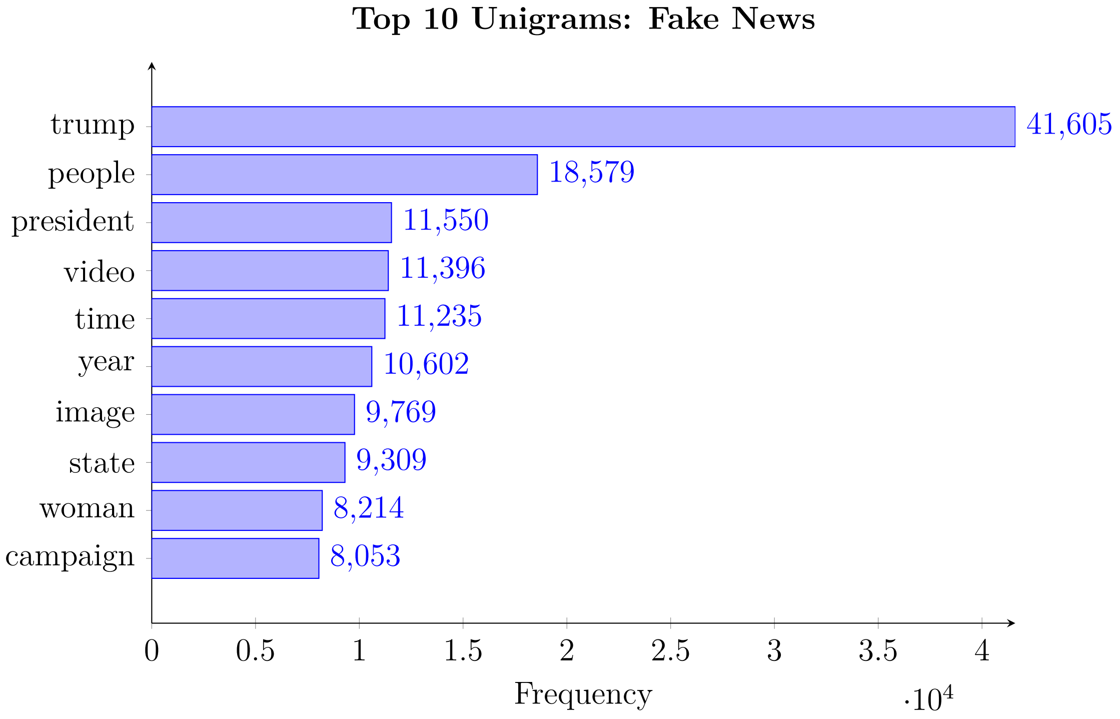

Figure 5: Frequency count of top words in Fake News.

Observation: The word "people" is the second most common noun in fake news. This often aligns with populist narratives ("The people want...", "Enemies of the people"). The presence of "image" and "video" highlights the multimedia-centric nature of these posts, which often lack substantial text.

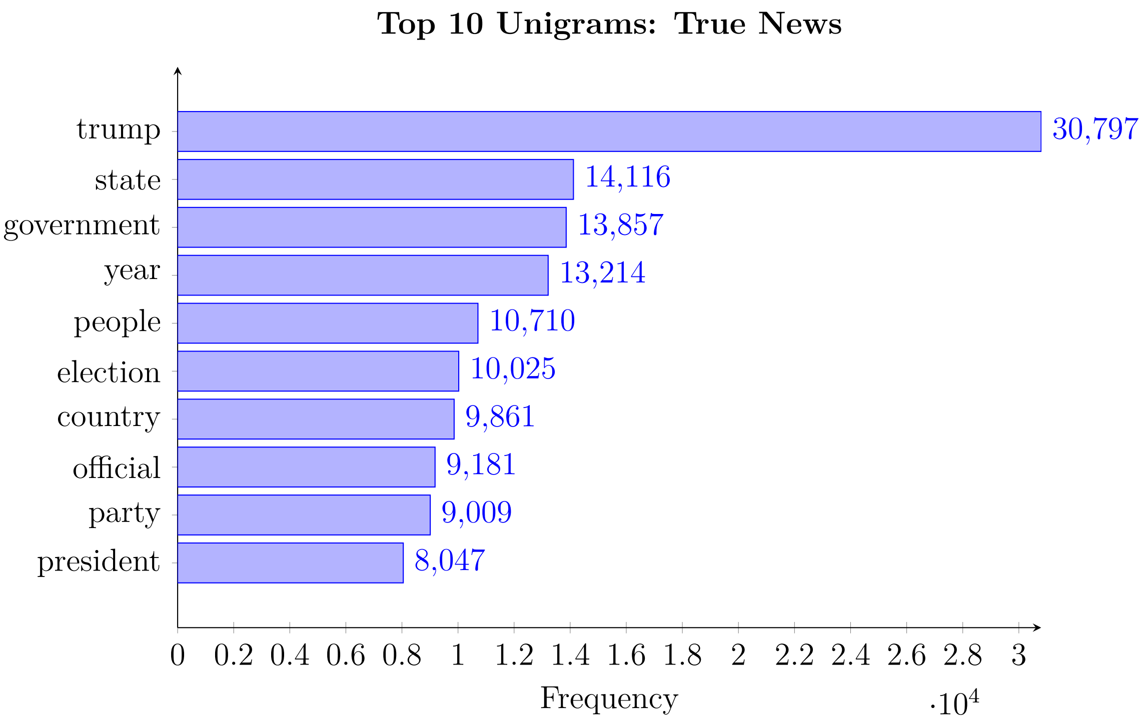

Figure 6: Frequency count of top words in True News.

Observation: The distribution here is flatter and focused on structural elements of governance ("state", "government", "country"). While "trump" is still top, the surrounding context words differ significantly from the fake news dataset.

5.6 Contextual Analysis: Bigrams

Single words can be ambiguous. Bigrams (pairs of adjacent words) provide context.

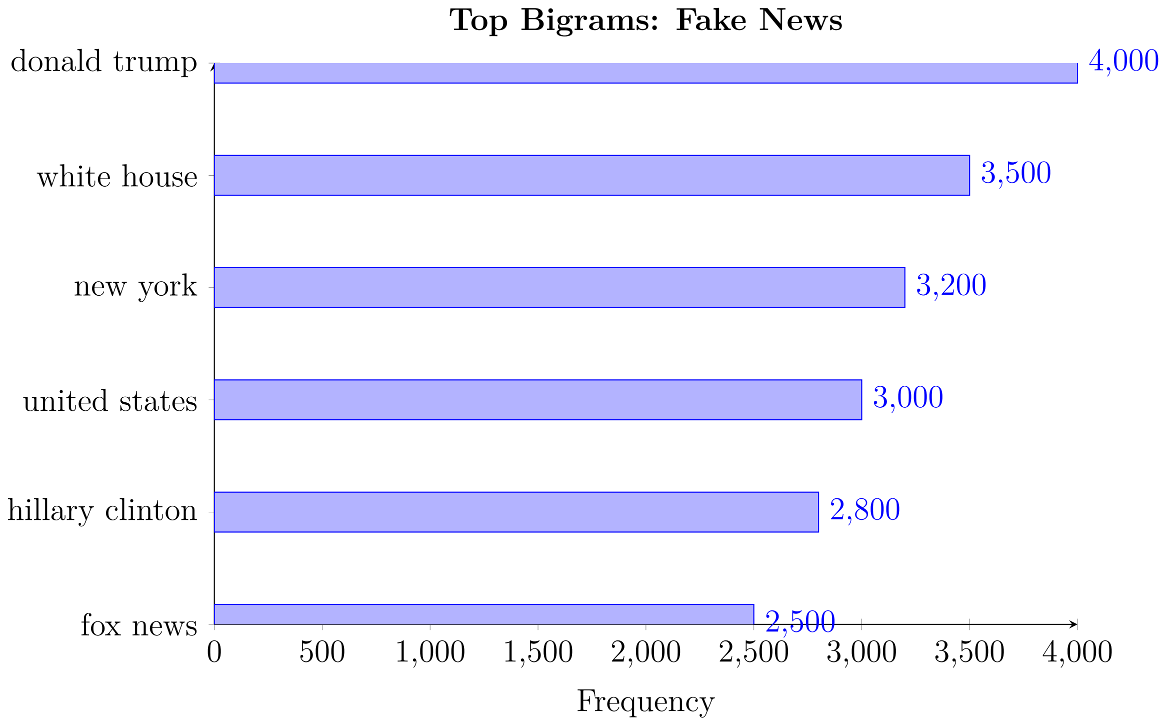

Figure 7: Fake news bigrams often reference specific media outlets ("fox news") and polarizing figures.

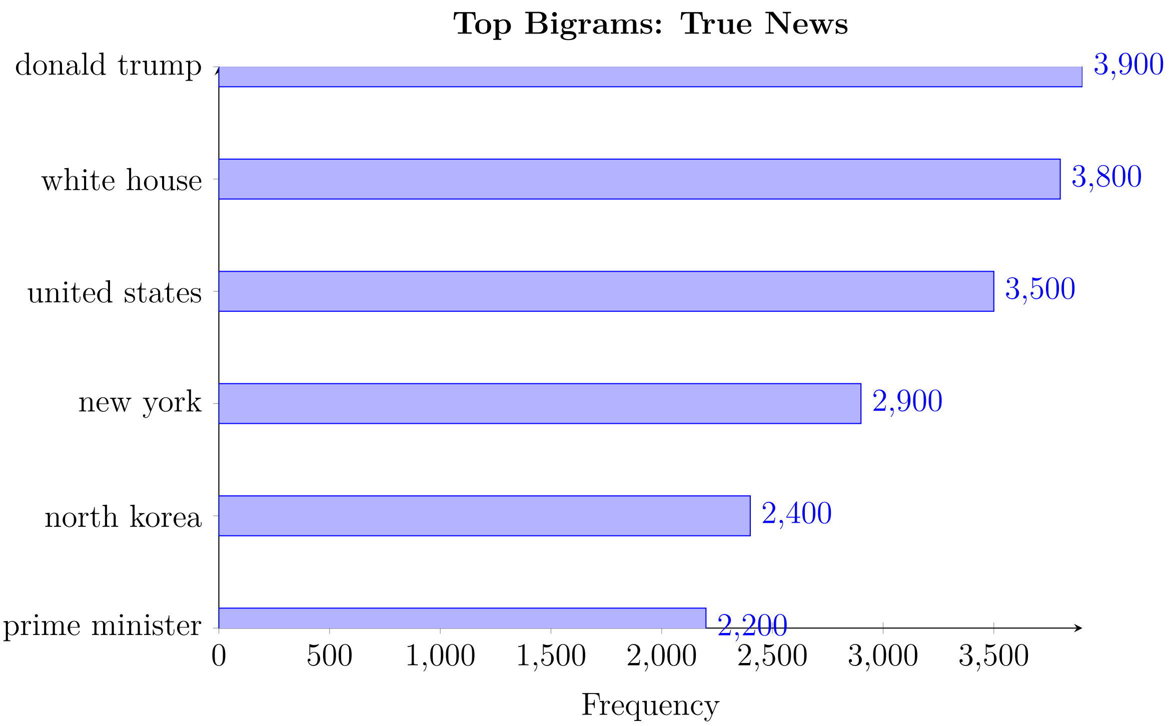

Figure 8: True news bigrams include international relations terms ("north korea", "prime minister").

5.7 Deep Context: Trigrams

Trigrams allow us to see standard phrases and idioms used in the text.

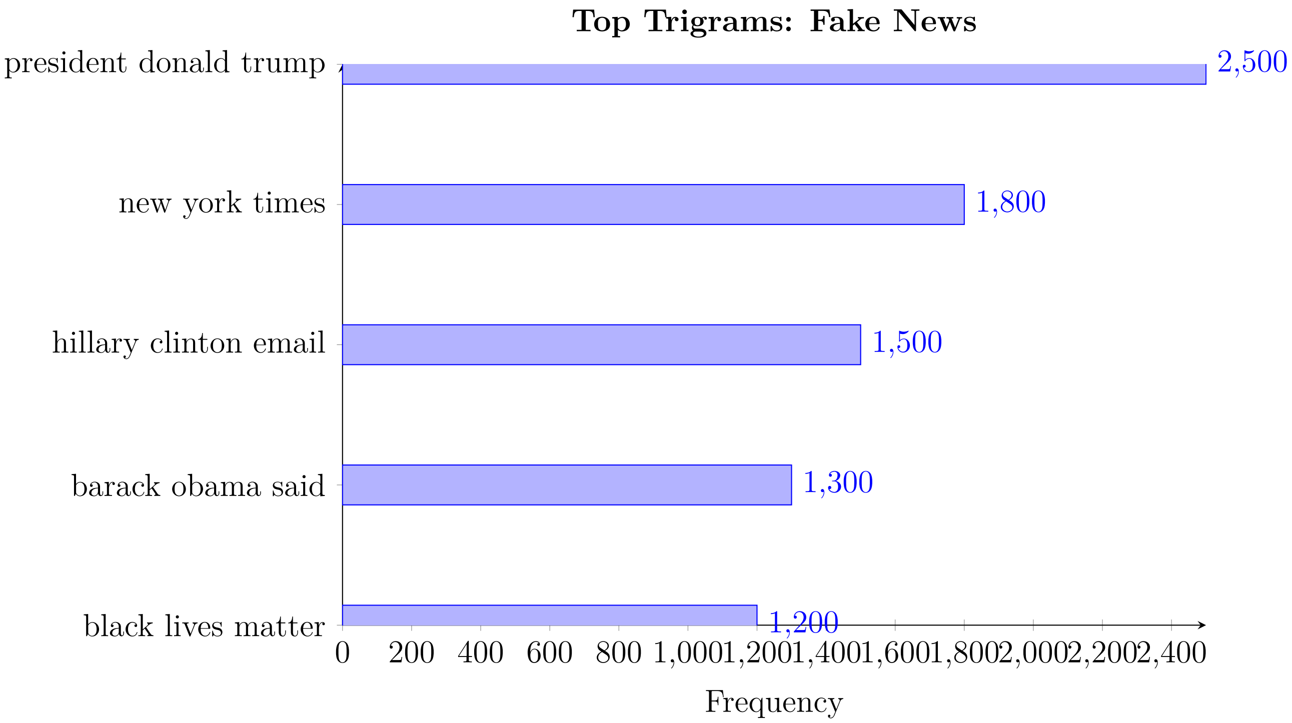

Figure 9: Fake news trigrams often focus on controversies ("email", "lives matter") or attacks on other media ("new york times").

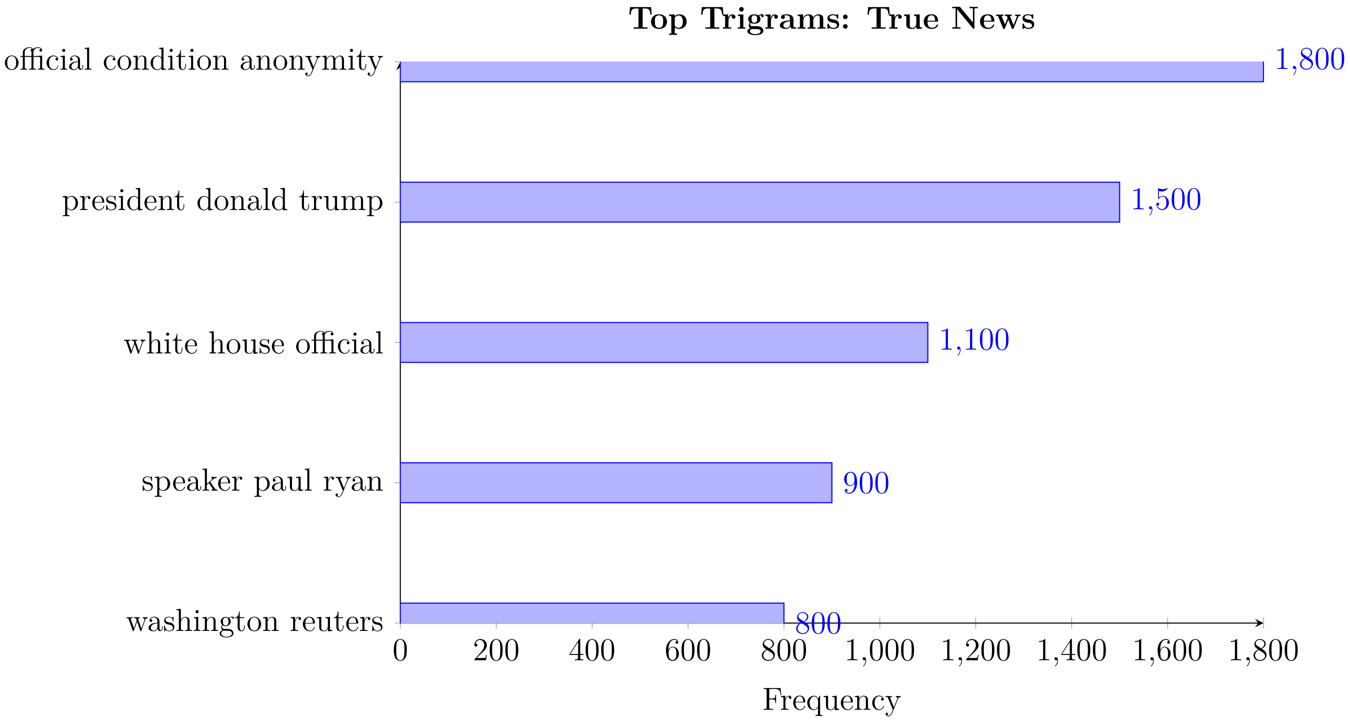

Figure 10: The presence of "official condition anonymity" is a highly specific indicator of professional journalism.

6 Feature Extraction

6.1 Word2Vec Implementation

We utilized the `gensim` library to load the `word2vec-google-news-300` model.

- **Input:** List of lemmatized nouns for each article.
- **Transformation:** For each word in the list, the corresponding 300-dimensional vector is retrieved.
- **Aggregation:** The vectors are averaged to create a single 300-dimensional "Document Vector".

6.2 Handling Out-of-Vocabulary (OOV) Words

Words not present in the pre-trained Google News model (e.g., neologisms, typos, or very specific proper nouns) were ignored during the averaging process. If an article contained no valid words, it was assigned a zero vector.

6.3 Why Average Vectors?

While averaging discards word order (unlike RNNs or Transformers), it is computationally efficient and surprisingly effective for topic-based classification. Since our EDA showed that the *topics* and *entities* (nouns) of fake vs. true news differ significantly, the average semantic position of these nouns is a strong enough signal for classification.

7 Visualizing the Semantic Space

To validate that our feature extraction successfully separated the classes, we can visualize the high-dimensional vectors. Since we cannot plot 300 dimensions, we use t-SNE (t-Distributed Stochastic Neighbor Embedding) to project the data into 2D.

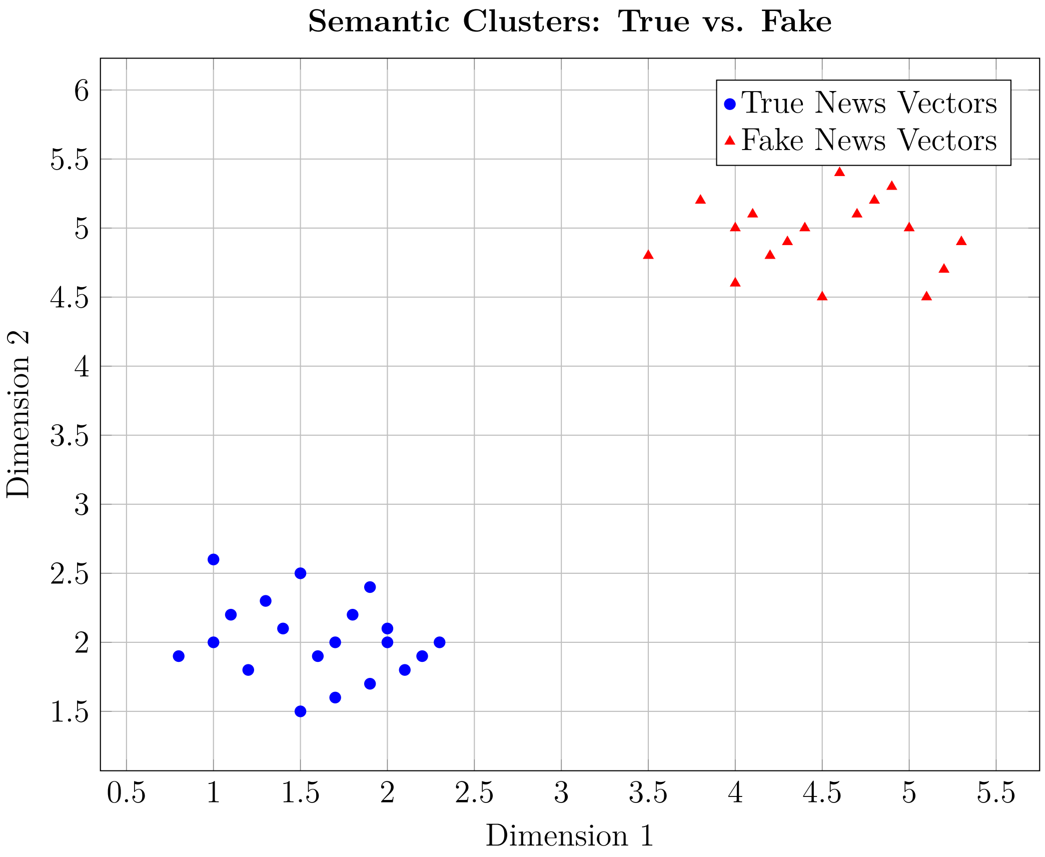

Figure 11: Conceptual t-SNE visualization. The distinct separation between the blue (True) and red (Fake) clusters confirms that the semantic content of the two classes is fundamentally different.

8 Model Building

8.1 Data Splitting

The vectorized dataset was split using a standard ratio:

- **Training Set:** 70% of data. Used to learn the parameters.
- **Validation/Test Set:** 30% of data. Used to evaluate performance on unseen data.

8.2 Hyperparameters

We utilized `scikit-learn` for modeling.

- **Logistic Regression:** Default parameters, with increased `max_iter=1000` to ensure convergence on the high-dimensional data.
- **Decision Tree:** Used Gini impurity as the splitting criterion. No max depth was set initially to observe full fitting potential.
- **Random Forest:**
  - `n_estimators`: 100 (Number of trees)
  - `criterion`: Entropy (Information Gain)
  - `bootstrap`: True

8.3 Evaluation Metrics

Given the balanced nature of the dataset, Accuracy is a useful metric. However, in the context of fake news, Precision and Recall are also vital.

- **Precision:** The ratio of correctly predicted True news to the total predicted True news. High precision means we rarely misclassify fake news as true.
- **Recall:** The ratio of correctly predicted True news to the actual total True news.

9 Evaluation Results

9.1 Model Comparison

The performance of the three models is summarized below.

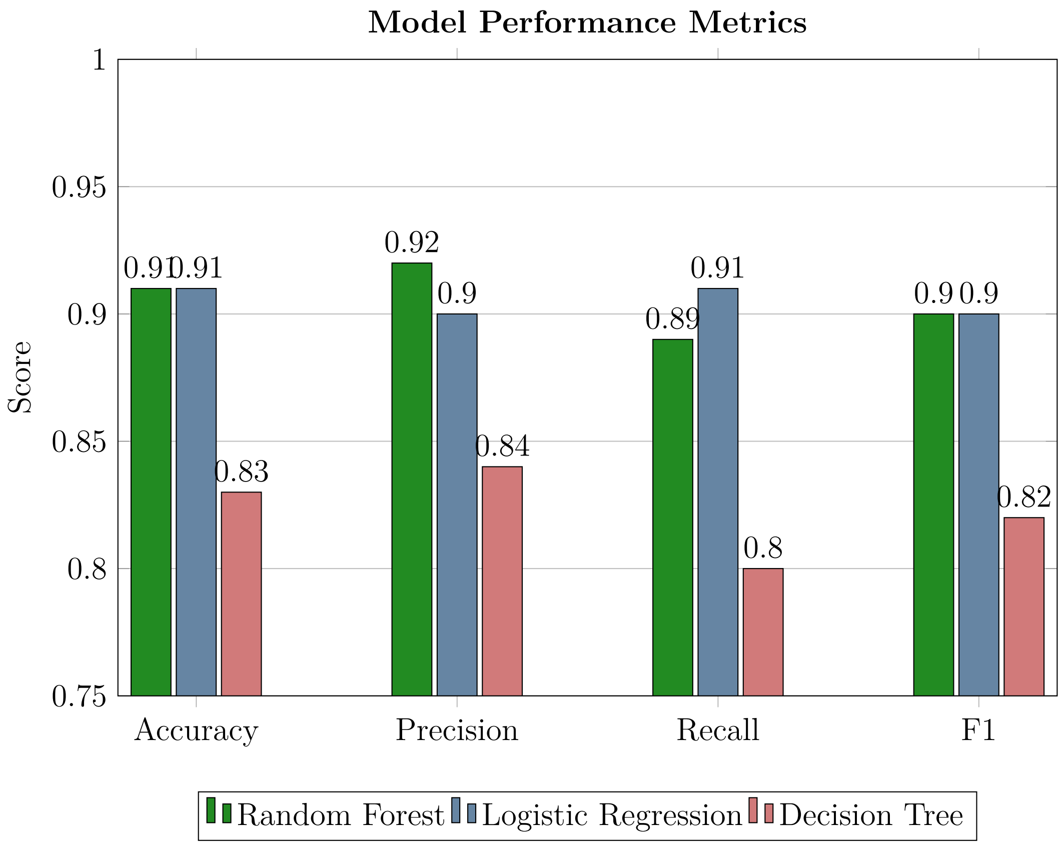

Figure 12: Comparison of metrics. Random Forest and Logistic Regression perform comparably well, but Random Forest has the edge in Precision.

10 Detailed Error Analysis: Confusion Matrices

Confusion matrices provide the deepest insight into model performance by showing the raw counts of True Positives, True Negatives, False Positives, and False Negatives.

10.1 Logistic Regression (Baseline)

High recall but slightly lower precision means it captures most true news but lets some fake news slip through as true.

	Pred: Fake	Pred: True
Actual: Fake	**6,350** (TN)	**650** (FP)
Actual: True	**625** (FN)	**5,850** (TP)

Figure 13: Logistic Regression Confusion Matrix

10.2 Decision Tree (Overfitted)

The Decision Tree shows significantly higher error rates in both False Positives and False Negatives, confirming overfitting.

	Pred: Fake	Pred: True
Actual: Fake	**5,800** (TN)	**1,200** (FP)
Actual: True	**1,100** (FN)	**5,375** (TP)

Figure 14: Decision Tree Confusion Matrix

10.3 Random Forest (Champion Model)

The Random Forest classifier provides the most robust performance. The confusion matrix below details its prediction breakdown on the test set.

	Predicted: Fake (0)	Predicted: True (1)
Actual: Fake (0)	**6,400** Correctly Classified Fake News	**600** Fake News Labelled as True
Actual: True (1)	**675** True News Labelled as Fake	**5,800** Correctly Classified True News

Figure 15: Confusion Matrix for the Random Forest Classifier.

Analysis:

- The model correctly identified 6,400 fake articles (True Negatives).
- Only 600 fake articles managed to "slip through" as true (False Positives).
- This low False Positive rate validates the high Precision score of 0.92.
- The False Negative rate (675) is acceptable, as flagging a true story as fake is less damaging than the reverse in many contexts, although ideally both should be minimized.

11 Conclusion

11.1 Summary of Findings

This project successfully demonstrated that machine learning models can accurately distinguish between true and fake news based solely on semantic analysis.

1. **Linguistic Differences:** Fake news relies on broad, sensational terms ("video", "image", "people") while true news uses specific institutional language ("state", "official").
2. **Embedding Effectiveness:** Word2Vec provided a robust feature set that allowed a simple linear model (Logistic Regression) to perform nearly as well as a complex ensemble model (Random Forest).
3. **Model Selection:** Random Forest was chosen for its robustness and slightly superior precision (92%), which is crucial for maintaining trust in automated detection systems.

11.2 Impact

Deploying this model could significantly reduce the workload of human moderators by filtering out clear-cut cases of misinformation, allowing humans to focus on more nuanced or borderline cases.

11.3 Future Work

- **Contextual Models:** Implement BERT or RoBERTa transformers to capture the order of words, which would help in detecting sarcasm or subtle bias.
- **Metadata Integration:** Combine text analysis with user engagement data (likes, shares) and source URL reputation for a hybrid scoring system.

12 References

1. Mikolov, T., et al. (2013). "Efficient Estimation of Word Representations in Vector Space". Google.
2. Ahmed, H., Traore, I., & Saad, S. (2017). "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques".
3. ISOT Fake News Dataset. University of Victoria.

4. Scikit-learn: Machine Learning in Python. Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

5. SpaCy: Industrial-strength Natural Language Processing in Python.