

Fake News Detection

1. Objective

The spread of fake news has become a significant challenge in today's digital world. With the massive volume of news articles published daily, it's becoming harder to distinguish between credible and misleading information. This creates a need for systems that can automatically classify news articles as true or fake, helping to reduce misinformation and protect public trust.

In this assignment, we developed a Semantic Classification model that uses the Word2Vec method to detect recurring patterns and themes in news articles. Using supervised learning models, the goal is to build a system that classifies news articles as either fake or true.

We aim to:

- Develop vector representations of news articles using **Word2Vec** embeddings.
- Train a supervised machine learning model to classify news articles as **true** or **fake**.
- Analyze the effectiveness of semantic information in detecting fake news.

2. Methodology

2.1 Data Overview

We use two datasets:

- True.csv — 21,417 articles labeled as **true**
- Fake.csv — 23,502 articles labeled as **fake**

Each dataset contains:

- title: Headline of the article
- text: Full content of the article
- date: Publication date (not used in modeling)

We concatenate the **title** and **text** columns to form the input for semantic representation.

2.2 Data Preprocessing

Steps:

- Lowercasing text
- Removing punctuation, Remove text in square brackets, Remove words containing numbers
- perform POS tagging and lemmatization
- remove all words that are not tagged as NN or NNS.

- Removal of English stopword

```
def clean_text(text):
```

```
# Convert to lower case
```

```
text=text.lower()
```

```
# Remove text in square brackets
```

```
text=re.sub(r'\[.*?\]', '', text)
```

```
text=re.sub(r'\s+', ' ',text)
```

```
# Remove punctuation
```

```
text=re.sub(r'[^\\w\\s]', '', text)
```

```
# Remove words with numbers
```

```
text=re.sub(r'\\b\\w*\\d\\w*\\b', '', text)
```

```
return text.strip()
```

```
def pos_lemma(text):
```

```
doc = nlp(text)
```

```
new=[token.lemma_ for token in doc if token.pos_=='NOUN' and not token.is_stop]
```

```
text=" ".join(new)
```

```
return text
```

2.3 Word Embedding with Word2Vec

We used **Word2Vec model** from word2vec-google-news-300 using the gensim library to generate **300-dimensional embeddings** for words based on their context.

Each text is converted into a fixed-length vector by averaging the embeddings of all words present in the text.

```
from gensim.models import KeyedVectors

file_path = r"C:\Users\hp\Downloads\GoogleNews-vectors-negative300.bin"

wv = KeyedVectors.load_word2vec_format(file_path, binary=True)

def vectorize(text):
    tokens=text.split()
    return wv.get_mean_vector(tokens)
```

2.4 Supervised Model Training

We applied traditional classification algorithms using the averaged Word2Vec vectors:

- **Random Forest**
- **Logistic Regression**
- **Decision tree**

We split the dataset into **70% training** and **30% testing** and evaluated performance using standard metrics.

Results & Key Insights

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9040	0.8990	0.8963	0.8977
Logistic Regression	0.8781	0.8664	0.8755	0.8709
Decision tree	0.8199	0.8168	0.7949	0.8057

Conclusion

True news tends to use more factual and neutral words. Semantic word embeddings (like Word2Vec) capture the contextual meaning of words, helping differentiate differences in word usage patterns between real and fake news. It improves the ability to detect fake news by leveraging meaning and context, not just keywords or simple lexical features

Random Forest Classifier emerged as the best performing model.

F1-score was prioritized as the key evaluation metric because it balances precision and recall. Since false positives and false negatives are both costly, incorrectly flagging real news and missing fake news have significant consequences.

Using semantic word embeddings provided rich contextual representations, enabling the classifier to capture language differences between true and fake news. Random Forest model's ensemble nature improved generalization over simpler classifiers.

This approach significantly improves automated fake news detection accuracy compared to baseline lexical or keyword-based methods. The balanced focus on precision and recall helps minimize harmful consequences from misclassification.
