Dissecting Fake and Real News: A Text-Based Analysis of Linguistic and Emotional Patterns

Abstract

This study focuses on the textual feature differences between true and false news. The core research question is: Can true and false news be effectively identified through language features? The article uses the ISOT Fake News Detection dataset (with a total of 44,898 news items), and comprehensively applies sentiment analysis, topic modeling and text classification methods to compare the performance of true and fake news at the text level.

According to the results of sentiment analysis, fake news is more extreme in overall sentiment (compound mean is -0.04), while real news is more neutral and positive (+0.10). The difference in sentiment between the title and the text is greater in fake news (mean is 0.68). Through LDA topic modeling, fake news focuses on topics such as "abortion", "email gate", and "violent conflict", while real news focuses on neutral reports such as "policy implementation" and "diplomatic relations", and the topic composition is significantly different. In the modeling part, I trained a logistic regression classifier based on TF-IDF features, and the accuracy of the title + text model reached 99%. And it can reach 95% using only the title. In general, there are systematic differences in language style, topic tendency and emotional expression between real and fake news.

Introduction

With the development of social media and digital platforms, the dissemination capacity of fake news has seen explosive growth in recent years (Tandoc, 2019). Especially in sensitive issues such as political elections, public health, and international conflicts, it has caused extensive social impacts (Olan et al., 2024). During the dissemination of information related to the 2020 US presidential election, the controversy over the "origin of the novel coronavirus", and the Russia-Ukraine war, fake news has repeatedly occupied the core of online public opinion. And it has been proved to have significant interference with public cognition, political trust and social behavior (Morgan, 2018). Some studies have shown that the forwarding speed and scope of fake news even far exceed those of real news (Zhao et al., 2020). Moreover, users often share due to emotional resonance rather than factual verification (Vziatysheva, 2020).

The current research on fake news can roughly be divided into two directions. On the one hand, it explores its diffusion mechanism in social networks from the perspective of dissemination (Mahmoud, 2020). Such as the echo chamber effect of the audience and the promoting effect of political tendencies on the spread of fake news (Cinelli, 2021). On the other hand, attempts are made to identify its language features at the content level. For example, emotional extremization, clickbait strategies, conspiracy theory elements, etc. (Choudhary & Arora, 2021). However, at present, the systematic analysis of true and false news in terms of language style aspects such as emotional expression, topic structure and vocabulary usage is still relatively limited (Lazer et al., 2018). In particular, there is a lack of a comprehensive approach that combines sentiment analysis, topic modeling and text classification models.

This study focuses on the following core questions: Are there systematic differences in language style between true and false news? Can these differences be used to automatically identify with high accuracy? The motivation of the study is to deeply understand the construction logic of false information and explore language features. To serve the subsequent development of automated fake news identification tools (Shu, 2017). I conducted an empirical analysis based on the ISOT Fake News Detection dataset. The study was conducted from three perspectives: sentiment polarity, topic content, and classification performance. First, the VADER tool was used to quantify the sentiment. Then the LDA model was used to extract the potential topics of the text. Finally, the TF-IDF + logistic regression model was used to train the classifier. In addition, this article also specifically examines the emotional consistency between the title and the text. To explore whether "title party" is a common manipulation strategy for fake news.

Based on the above background and existing research, this paper proposes the following research hypotheses:

H1: Fake news tends to be more extreme than neutral in emotional expression.

H2: The differences in language features between true and false news can be effectively modeled and used for classification with high accuracy.

Methodology

Data Source and Preprocessing

The data used in this study is from the ISOT Fake News Detection Dataset collated and released by the Information Security research team of the University of Victoria in Canada. The dataset consists of two independent CSV files, corresponding to true news and false news respectively, with a total of 44,898 news samples. Each record contains four fields: title, text, subject and date.

To construct a unified analytical framework, I first merge the two sub-data sets. And add a new column "label" to mark the news type, where 1 represents true news and 0 represents false news. In terms of data cleaning, I standardized the text field, including converting it to lowercase, removing HTML tags, punctuation marks, isolated letters, numbers, special symbols, and eliminating English stop words. The processed text is saved in the "clean_text" column. During the modeling stage, I concatenated the title with the main text to generate a new text column, combined_text. It is used to capture cross-field language features in model modeling.

Table 1

Data Example (Partial)

title	text	subject	date	label
As U.S. budget fight...	WASHINGTON (Reuters) ...	politicsNews	December 31, 2017	1
U.S. military to...	WASHINGTON (Reuters) ...	politicsNews	December 29, 2017	1
Senior U.S....	WASHINGTON (Reuters) ...	politicsNews	December 31, 2017	1

Sentiment Analysis

In order to detect the difference in emotional tendencies between true and false news, I used the Valence Aware Dictionary and sEntiment Reasoner tool for sentiment analysis. VADER is a sentiment dictionary designed for social media and news short texts (Hutto & Gilbert, 2014). The dictionary can output four sentiment scores: negative, neutral, positive, and compound (Borg & Boldt, 2020). Among them, compound is a comprehensive indicator reflecting the overall emotional tendency, ranging from -1 to +1. The more negative, the more negative, and the more positive, the more positive.

The analysis objects include the text and the title. I not only calculated the overall sentiment score of each article, but also further quantified the emotional difference between the title and the text. This is to measure whether the news has "clickbait"-style exaggeration or misleading expressions. By associating these emotional variables with the labels of the news, I hope to identify whether fake news presents higher extremism or inconsistency at the emotional level. This part is used to test the research hypothesis H1.

Topic Modeling

In terms of exploring the topic structure of true and false news, this paper uses the Latent Dirichlet Allocation method to construct a topic model(Jelodar et al., 2019). To ensure the comparability of model training, I established two LDA models for true and false news respectively, and set the number of topics for each to 10. And it is trained through the Gensim toolkit.

Each topic in the model results is composed of several keywords, reflecting its semantic core. The article manually interprets the keyword combination and assigns semantic labels to each topic. For example, "abortion controversy", "email gate incident", "campus safety", etc. In addition, I counted the distribution of true and false news on different topics to reveal the difference in topic focus between the two types of news. In addition, I also observed the correlation between topic distribution and emotional variables to explore whether fake news tends to focus on topics with high emotional tension.

Text Modeling And Classifier Construction

To evaluate the discriminability of true and false news in text features, this study trained multiple text classification models. First, use the Term frequency-inverse Document Frequency method to vectorize the text (Christian et al., 2016). Considering the existence of fixed collocations in language patterns, unigram and bigram were selected as the basic features for the study. And limit the feature dimension to 5000 to control the complexity of the model.

Three sets of input form models were constructed in the research: using only the title, using only the main text, and the spliced text of the title and the main text, combined_text. Each group of models uses the same logistic regression classifier and is fitted on the training set. And evaluate on 20% of the test set, and output indicators such as accuracy rate, precision rate, recall rate and F1 value. This part is for the verification of the research hypothesis H2.

Furthermore, to enhance the interpretability of the model, I extracted the word weight coefficients in logistic regression. I analyzed the most distinctive keywords and determine their tendencies in fake news or true news. This method is helpful for identifying words disguised as authoritative sources, repetitive emotional words, etc (Lemon et al., 2003).

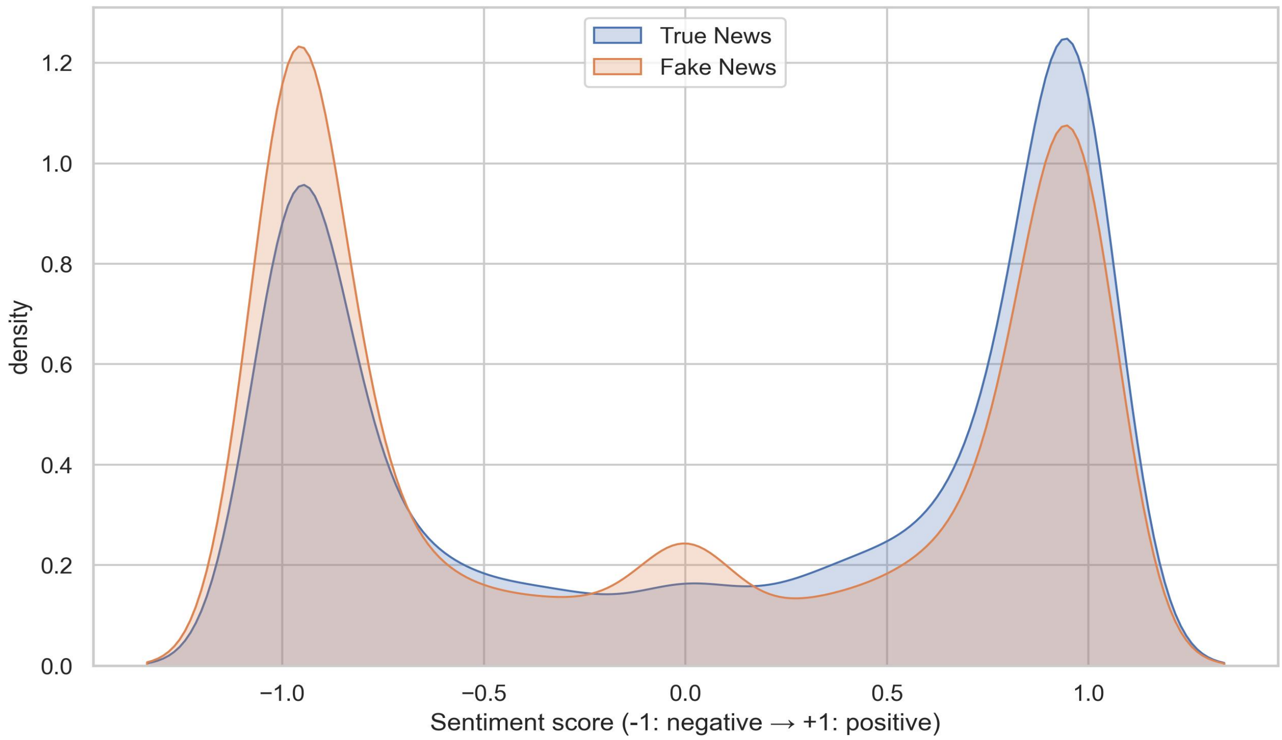
Discussion of The Results

Sentiment Analysis Results

The study first compared the overall distribution of true and false news in terms of sentiment. From the overall mean, the sentiment distribution of false news is obviously more extreme, with a mean of -0.04. On the other hand, true news is more positive, with a mean of +0.10. This shows that false news often uses negative or radical words in language, which may be more likely to evoke emotional responses from users.

Figure 1

Difference in Distribution Density Between True And Fake News (Compound)

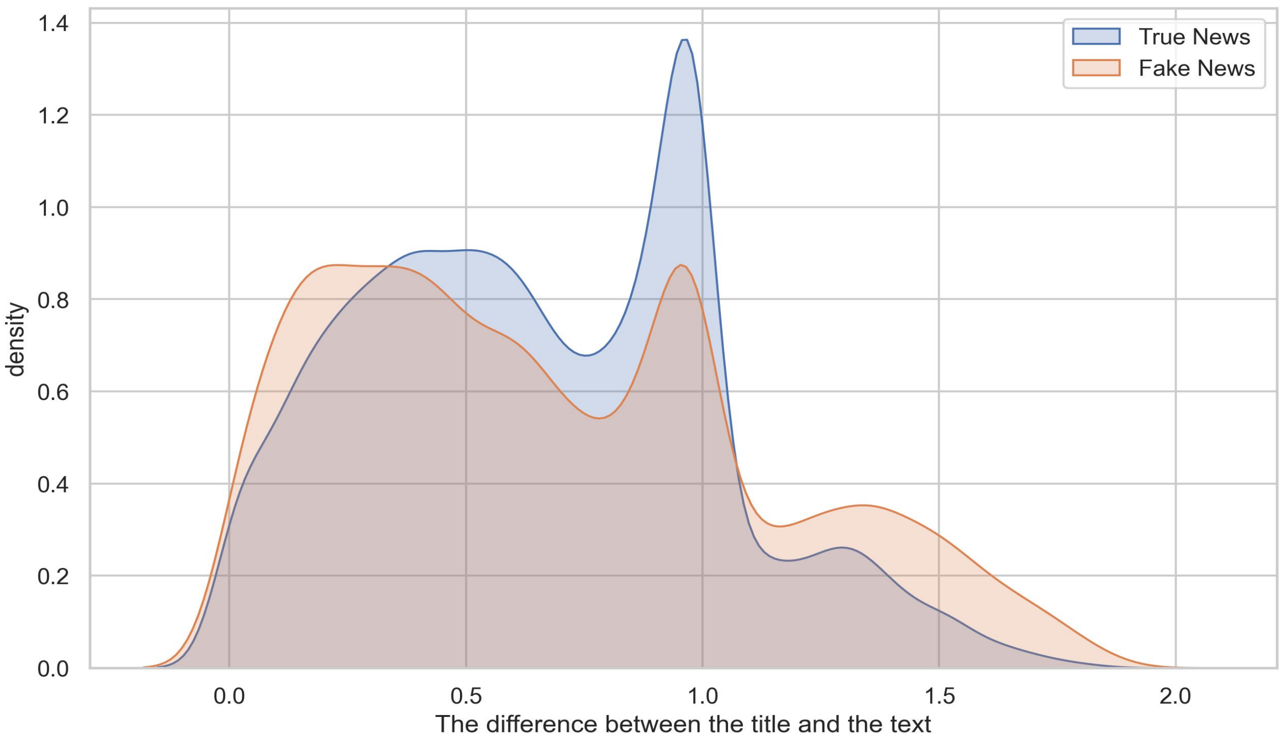


The sentiment curve of fake news shows a wider bimodal pattern, indicating that it uses more extreme expressions. The true news, on the other hand, is concentrated in the neutral or slightly positive range. The average negative score in fake news was higher than that in true news (0.142 vs. 0.115), while the neutral component was significantly less (0.682 vs. 0.754). It is indicated that fake news is more "positional" rather than "reporting" in emotional expression.

Furthermore, I analyzed the emotional consistency between the title and the main text. The results show that the average emotional difference between the title and the main text of fake news is 0.684, while that of real news is 0.657. Although the difference between the two is not significant, the distribution map (see Figure 2) indicates that there are more typical cases of "extreme headlines and moderate text" in fake news. This strategy might be a misleading mechanism in fake news, used to attract clicks but avoiding risky expressions in the main text.

Figure 2

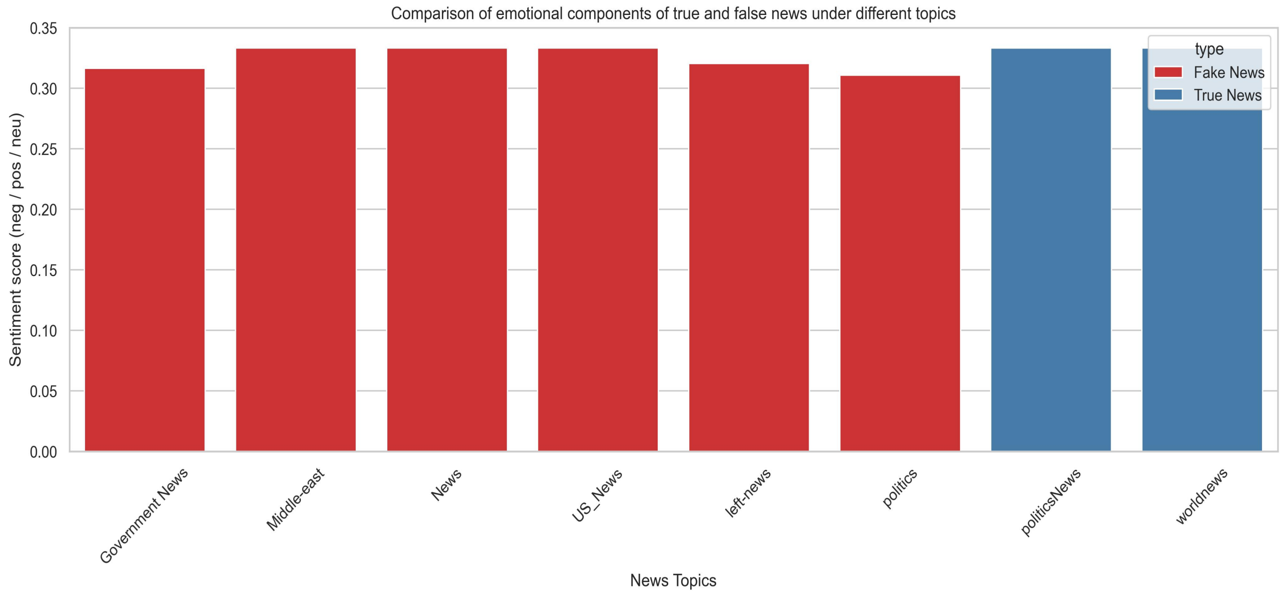
Differences in Emotional Consistency Between Title and Texts of True and Fake News



In addition, this study also analyzed the emotional composition of true and false news under different topic categories. It can be observed that the average intensity of emotional expression is roughly the same for both true and false news, mainly concentrated between 0.31 and 0.33. This shows that from the perspective of different topic classifications, the difference in overall emotional expression between true and false news is not significant. This shows that the emotional extreme characteristics of false news are more likely to appear in the text rather than the topic.

Figure 3

Comparison of Emotional Components of True and False News under Different Topics



Topic Modeling Results

I used LDA to model fake news and real news separately, and extracted the 10 most representative topics of each. The preprocessed text was used in model training, and the topic labels were summarized based on keywords. Only the top 5 are shown here, as shown in Tables 2 and 3.

Table 2

True News Topics (Top 5)

Topic	Keywords
Urban Security	'said', 'people', 'reuters', 'police', 'city', 'year', 'one', 'two', 'island', 'home'
Energy Policy	'said', 'state', 'company', 'department', 'trump', 'administration', 'would', 'agency', 'energy', 'oil'
Law and Human Rights	'said', 'court', 'right', 'myanmar', 'state', 'law', 'people', 'government', 'rohingya', 'reuters'
International Relations	'said', 'north', 'state', 'united', 'korea', 'china', 'trump', 'would', 'minister', 'president'
Elections	'party', 'trump', 'said', 'election', 'vote', 'campaign', 'clinton', 'presidential', 'republican', 'would'

Table 3

Fake News Topics (Top 5)

Topic	Keywords
Identity Politics	'medium', 'white', 'political', 'black', 'group', 'like', 'event', 'wire', 'social', 'one'
Hillary email controversy	'clinton', 'fbi', 'email', 'news', 'russian', 'hillary', 'report', 'story', 'intelligence', 'medium'
Illegal immigration	'state', 'law', 'federal', 'court', 'government', 'illegal', 'immigration', 'said', 'would', 'order'
Trump's Remarks and Media	'trump', 'donald', 'president', 'news', 'said', 'cnn', 'medium', 'like', 'fox', 'time'
Middle East War	'syria', 'state', 'war', 'military', 'obama', 'muslim', 'country', 'terrorist', 'united', 'president'

From the perspective of topic distribution, fake news focuses on socially sensitive and politically polarized topics such as "abortion controversy", "email gate incident", "immigration and terrorism", etc., while real news focuses on institutional topics such as "government budget", "China-US diplomacy", "court rulings", etc. This difference indirectly verifies the construction logic of fake news. That is, it strengthens readers' sense of involvement and beliefs through emotional and highly controversial topics, rather than objectively presenting information.

Classification Model Performance

This paper builds three sets of logistic regression classification models, which are trained based on the TF-IDF feature vectors of the title, the text, and the combination of the two. All models are evaluated on the 20% test set, and the results are shown in Table 4. Figure 4 is the confusion matrix of the "title only" and "text only" models.

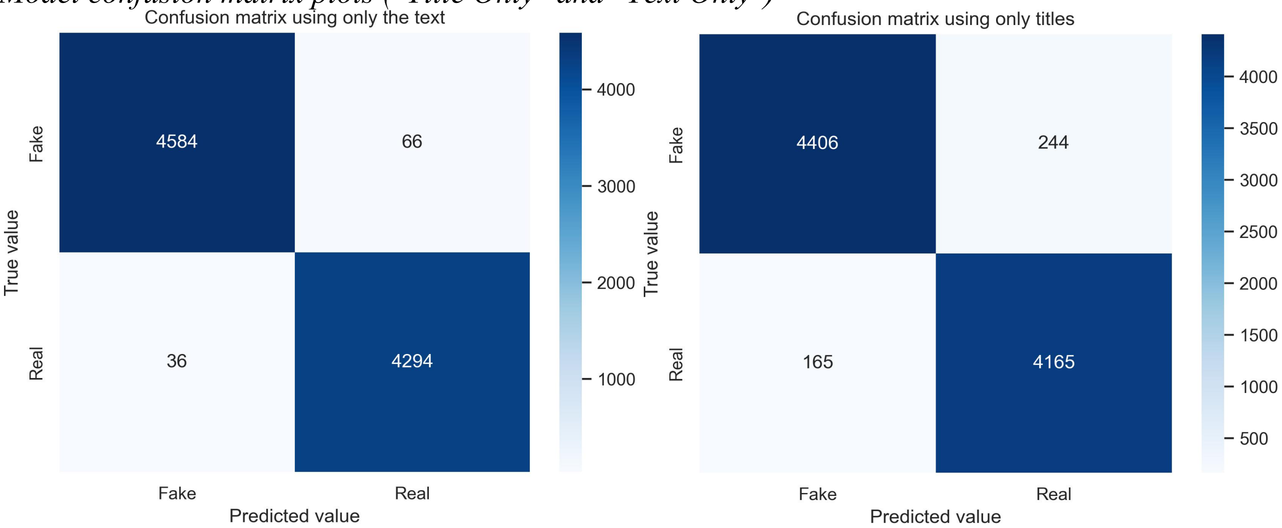
Table 4

Model Performance

Model Input	Accuracy	F1 score
Title	95%	0.95
Text	99%	0.99
Title + Text	99%	0.99

Figure 4

Model confusion matrix plots ("Title Only" and "Text Only")



Judging from the results, the text features are sufficient to support high-accuracy modeling. Although modeling with titles alone can achieve a classification accuracy rate of 95%, the main text remains the most crucial source of information for the model. This also indicates that fake news is not only reflected in attractive titles in terms of language style, but its overall language structure and wording are also significantly different from those of real news. This part of the results supports the research hypothesis H2.

Model Interpretation Results

To explain the basis for the model to distinguish true from false news, I extracted the keyword weights in the logistic regression model. Positive weight words (Figure 5) such as "reuters" and "president donald" are widely used in mainstream news reports. They have formal, objective, and well-structured language characteristics. The model is more likely to predict true news when these words appear.

In contrast, negative weight words (Figure 6) such as "video" and "breaking" are more common in false news. These words have strong provocative and visually guided tendencies. This result verifies that there are indeed systematic language differences in the text, and the model can effectively distinguish true from false news based on this.

Figure 5

Keywords with the Most True News Characteristics

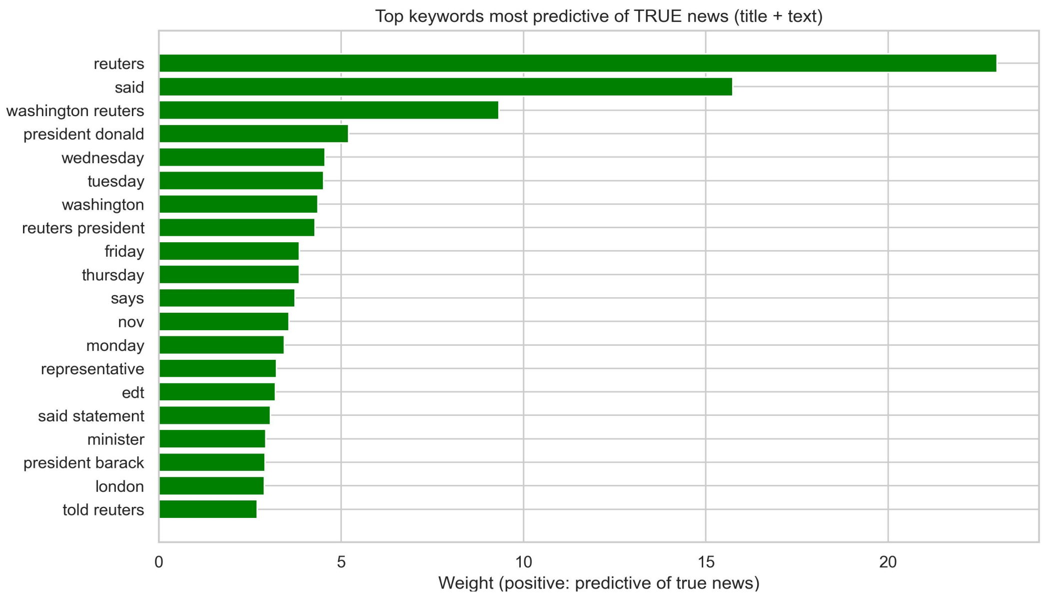
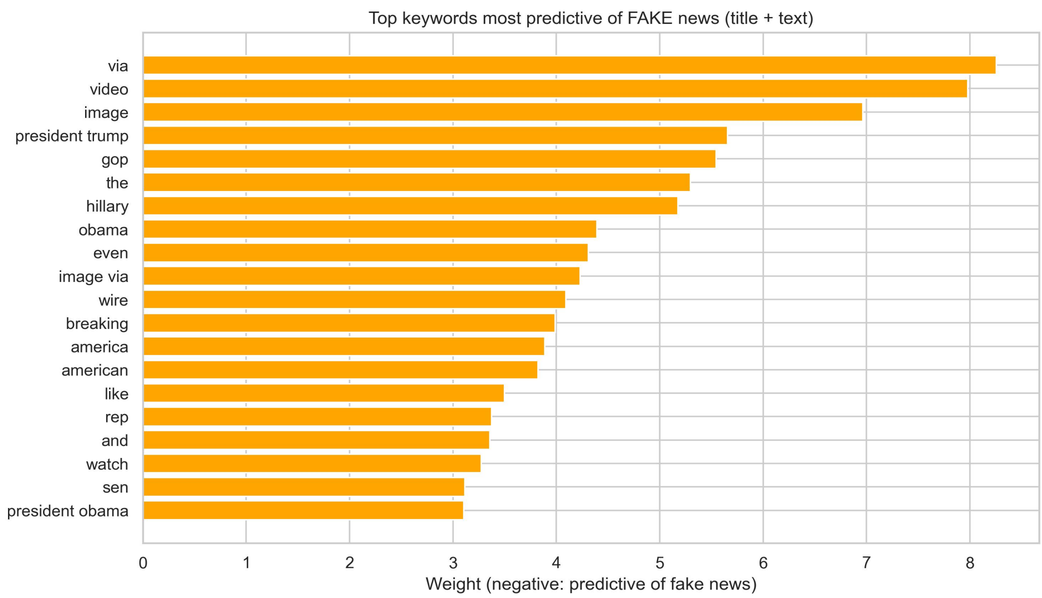


Figure 6

Keywords with the Most Fake News Characteristics



Conclusion

This study is based on the ISOT fake news dataset and comprehensively employs methods such as sentiment analysis, topic modeling, keyword weight interpretation and text classification. The differences between true and false news at the textual language level were discussed. The core objective of the research is to verify two hypotheses: First, fake news is more extreme in emotional expression (H1). Secondly, it is possible to effectively classify true and false news only relying on language features (H2).

The results of the study generally support these two hypotheses. Sentiment analysis shows that fake news often shows higher emotional polarity in the title. The text is relatively mild, forming a structure of "emotional title + neutral text". It has misleading communication characteristics. Relatively speaking, real news is more stable and consistent in emotional expression, and is mainly neutrally distributed. Topic modeling further reveals that although real and fake news have overlaps in terms of topics involved. However, there are significant differences in keywords and semantics.

In terms of classification modeling, whether using the title, the text, or the text content of the combination of the two, the classification accuracy rate can reach more than 95%. The highest can reach 99%. This result verifies the strong explanatory power of the text itself in the identification of true and false news. Afterwards, keyword weight analysis further revealed that real news prefers to use neutral and objective words such as "reuters", "said", and "official". Fake news more often contains emotionally rendered or attractive expressions such as "video", "image", and "breaking".

In conclusion, true and false news show consistent differences in terms of emotional intensity, structural arrangement, language strategy and expression methods. Text language can serve as an effective basis for recognition.

References

Borg, A., & Boldt, M. (2020). Using VADER sentiment and SVM for predicting customer response sentiment. Expert Systems with Applications, 162, 113746. https://doi.org/10.1016/j.eswa.2020.113746

Choudhary, A., & Arora, A. (2021). Linguistic feature based learning model for fake news detection and classification. Expert Systems with Applications, 169, 114171. https://doi.org/10.1016/j.eswa.2020.114171

Christian, H., Agus, M. P., & Suhartono, D. (2016). Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). ComTech: Computer, Mathematics and Engineering Applications, 7(4), 285-294. https://doi.org/10.21512/comtech.v7i4.3746

Cinelli, M., De Francisci Morales, G., Galeazzi, A., Quattrociocchi, W., & Starnini, M. (2021). The echo chamber effect on social media. Proceedings of the national academy of sciences, 118(9), e2023301118. https://doi.org/10.1073/pnas.2023301118

Hutto, C., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the international AAAI conference on web and social media (Vol. 8, No. 1, pp. 216-225). https://doi.org/10.1609/icwsm.v8i1.14550

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimedia tools and applications, 78, 15169-15211. https://doi.org/10.1007/s11042-018-6894-4

Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., ... & Zittrain, J. L. (2018). The science of fake news. Science, 359(6380), 1094-1096. 10.1126/science.aao2998

Lemon, S. C., Roy, J., Clark, M. A., Friedmann, P. D., & Rakowski, W. (2003). Classification and regression tree analysis in public health: methodological review and comparison with logistic regression. Annals of behavioral medicine, 26, 172-181. https://doi.org/10.1207/S15324796ABM2603_02

Mahmoud, H. (2020). A model for the spreading of fake news. Journal of Applied Probability, 57(1), 332-342. 10.1017/jpr.2019.103

Morgan, S. (2018). Fake news, disinformation, manipulation and online tactics to undermine

democracy. Journal of cyber policy, 3(1), 39-43.
https://doi.org/10.1080/23738871.2018.1462395

Olan, F., Jayawickrama, U., Arakpogun, E. O., Suklan, J., & Liu, S. (2024). Fake news on social media: the impact on society. Information Systems Frontiers, 26(2), 443-458.
https://doi.org/10.1007/s10796-022-10242-z

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1), 22-36.
https://doi.org/10.1145/3137597.313760

Tandoc Jr, E. C. (2019). The facts of fake news: A research review. Sociology Compass, 13(9), e12724.
https://doi.org/10.1111/soc4.12724

Vziatysheva, V. (2020). How fake news spreads online?. International Journal of Media & Information Literacy, 5(2).
10.13187/ijmil.2020.2.217

Zhao, Z., Zhao, J., Sano, Y., Levy, O., Takayasu, H., Takayasu, M., ... & Havlin, S. (2020). Fake news propagates differently from real news even at early stages of spreading. EPJ data science, 9(1), 7.
https://doi.org/10.1140/epjds/s13688-020-00224-z

Appendix

```
import pandas as pd
import numpy as np
import nltk
import re
import string
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import swifter
from tqdm.notebook import tqdm
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import seaborn as sns
import matplotlib.pyplot as plt
from gensim import corpora
from gensim.models import LdaModel
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Load true and fake news
true_df = pd.read_csv('/Users/zach/Desktop/text _analysis/True.csv')
fake_df = pd.read_csv('/Users/zach/Desktop/text _analysis/Fake.csv')

# assign labels
true_df['label'] = 1
fake_df['label'] = 0

# concatenate into a single DataFrame
df = pd.concat([true_df, fake_df], ignore_index=True)

# View basic information
print(df.info())
print(df.head())

# Skip SSL certificate verification
import ssl

try:
    _create_unverified_https_context = ssl._create_unverified_context
except AttributeError:
    pass
else:
    ssl._create_default_https_context =
_create_unverified_https_context
```

```
import warnings
# Ignore warning messages
warnings.filterwarnings("ignore", category=UserWarning,
module="xgboost")

# Download related function
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Initialize the processor
stop_words = set(stopwords.words('english'))# Stop words
lemmatizer = WordNetLemmatizer()

# Define the cleaning function
def clean_text(text):
    # Convert to lowercase
    text = text.lower()
    # Remove URLs, punctuation, and numbers
    text = re.sub(r"http\S+|www\S+|https\S+", '', text,
flags=re.MULTILINE)
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Word segmentation
    tokens = nltk.word_tokenize(text)
    # Remove stop words and restore lemmas
    cleaned = [lemmatizer.lemmatize(word) for word in tokens if word
not in stop_words and len(word) > 2]
    return ' '.join(cleaned)

#Show progress bar
tqdm.pandas()
# Use swifter to speed up text cleaning
df['clean_text'] = df['text'].progress_apply(clean_text)

# Add text length features
df['word_count'] = df['text'].apply(lambda x: len(str(x).split()))
df['char_count'] = df['text'].apply(lambda x: len(str(x)))

# Word frequency statistics
from collections import Counter
all_words = ' '.join(df['clean_text']).split()
word_freq = Counter(all_words)
common_words = word_freq.most_common(20)

# Display the top 20 words
print("Top 20 common words:")
for word, freq in common_words:
    print(f"{word}: {freq}")
```

```
# Initialize VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()

# Define function: Get sentiment scores
def get_sentiment_scores(text):
    return analyzer.polarity_scores(text)

# Emotionally score the cleaned text
sentiment_scores =
df['clean_text'].progress_apply(get_sentiment_scores)

# Split out the specific score column
# Negative sentiment score
df['neg'] = sentiment_scores.apply(lambda x: x['neg'])
# Neutral sentiment score
df['neu'] = sentiment_scores.apply(lambda x: x['neu'])
# Positive sentiment score
df['pos'] = sentiment_scores.apply(lambda x: x['pos'])
# Overall sentiment score (range -1 to +1)
df['compound'] = sentiment_scores.apply(lambda x: x['compound'])

# Setting the style
sns.set(style='whitegrid')

# Plot the distribution of compound sentiment scores for fake vs true
news
plt.figure(figsize=(10, 6))
sns.kdeplot(df[df['label'] == 1]['compound'], label='True News',
shade=True)
sns.kdeplot(df[df['label'] == 0]['compound'], label='Fake News',
shade=True)
plt.title("(Compound Score Distribution)")
plt.xlabel("Sentiment score (-1: negative → +1: positive)")
plt.ylabel("density")
plt.legend()
plt.savefig("compound_score_distribution.png", dpi=300,
bbox_inches='tight')
plt.show()

# Display average sentiment scores grouped by label
mean_sentiments = df.groupby('label')[['neg', 'neu', 'pos',
'compound']].mean()
print("Mean Sentiment Scores for True and False News:")
print(mean_sentiments)

# Calculate the average sentiment score for each subject and label
```

```
sentiment_by_subject = df.groupby(['label', 'subject'])[['neg', 'pos',
'neu']].mean().reset_index()

# Convert label to text for legend display
sentiment_by_subject['label'] = sentiment_by_subject['label'].map({0:
'Fake News', 1: 'True News'})

# Use melt to convert the data into long format to facilitate seaborn
drawing
sentiment_melted = pd.melt(sentiment_by_subject, id_vars=['label',
'subject'],
                           value_vars=['neg', 'pos', 'neu'],
                           var_name='Sentiment Type',
value_name='Score')
# Visualization: sentiment type × topic × true or false news
plt.figure(figsize=(14, 6))
sns.barplot(data=sentiment_melted, x='subject', y='Score',
hue='label', palette='Set1', ci=None)
plt.title("Comparison of emotional components of true and false news
under different topics")
plt.xlabel("News Topics")
plt.ylabel("Sentiment score (neg / pos / neu)")
plt.xticks(rotation=45)
plt.legend(title='type')
plt.tight_layout()
plt.savefig("sentiment_breakdown_by_subject.png", dpi=300,
bbox_inches='tight')
plt.show()

# Do sentiment analysis on the title
df['title_clean'] = df['title'].fillna('').apply(clean_text)
df['title_sentiment'] = df['title_clean'].progress_apply(lambda x:
analyzer.polarity_scores(x))

# Extract the compound score of the title
df['title_compound'] = df['title_sentiment'].apply(lambda x:
x['compound'])

# Calculate the difference in sentiment between the title and the
text
df['sentiment_gap'] = abs(df['title_compound'] - df['compound'])

# Print mean comparison
gap_summary = df.groupby('label')['sentiment_gap'].mean()
print("Difference in sentiment between title and body text：")
print(gap_summary)

# Visualize the distribution of sentiment differences
```

```
plt.figure(figsize=(10, 6))
sns.kdeplot(df[df['label'] == 1]['sentiment_gap'], label='True News',
shade=True)
sns.kdeplot(df[df['label'] == 0]['sentiment_gap'], label='Fake News',
shade=True)
plt.title("Differences in emotional consistency between headlines and
texts of true and false news")
plt.xlabel("The difference between the title and the text")
plt.ylabel("density")
plt.legend()
plt.savefig("sentiment_gap_distribution.png", dpi=300,
bbox_inches='tight')
plt.show()

# word segmentation function
def simple_tokenize(text):
    return text.split()

# False news and true news segmentation
fake_tokens = df[df['label'] ==
0]['clean_text'].apply(simple_tokenize).tolist()
true_tokens = df[df['label'] ==
1]['clean_text'].apply(simple_tokenize).tolist()

# Fake news dictionary and corpus
fake_dictionary = corpora.Dictionary(fake_tokens)
fake_corpus = [fake_dictionary.doc2bow(text) for text in fake_tokens]

# Train the LDA model
fake_lda_model = LdaModel(corpus=fake_corpus,
                          id2word=fake_dictionary,
                          num_topics=10,
                          random_state=42,
                          passes=10,
                          alpha='auto',
                          per_word_topics=True)

# Print topic keywords
print("Fake news topic keywords：")
for idx, topic in fake_lda_model.print_topics(-1):
    print(f"Theme {idx}: {topic}")

# Real News Dictionary and Corpus
true_dictionary = corpora.Dictionary(true_tokens)
true_corpus = [true_dictionary.doc2bow(text) for text in true_tokens]

true_lda_model = LdaModel(corpus=true_corpus,
                          id2word=true_dictionary,
```

```
                                 num_topics=10,
                                 random_state=42,
                                 passes=10,
                                 alpha='auto',
                                 per_word_topics=True)

print("\nTrue news topic keywords：")
for idx, topic in true_lda_model.print_topics(-1):
    print(f"Theme {idx}: {topic}")

# Use title and body combination to enhance classification effect
df['combined_text'] = df['title'].fillna('') + ' ' + df['clean_text']

# Initialize TF-IDF vectorizer
vectorizer = TfidfVectorizer(max_features=5000, ngram_range=(1, 2))

# Convert text to vector
X = vectorizer.fit_transform(df['combined_text'])
y = df['label']

# Divide into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Train the model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predict test set
y_pred = model.predict(X_test)

# Output evaluation metrics
print("Classification Report：")
print(classification_report(y_test, y_pred))


print("Confusion Matrix：")
print(confusion_matrix(y_test, y_pred))

# Use title only to classify
# Vectorize title text
vectorizer_title = TfidfVectorizer(max_features=5000, ngram_range=(1,
2))
X_title = vectorizer_title.fit_transform(df['title'].fillna(''))
y = df['label']

# Divide into training set and test set
X_train_title, X_test_title, y_train_title, y_test_title =
train_test_split(
```

```
    X_title, y, test_size=0.2, random_state=42
)

# Train the logistic regression model
model_title = LogisticRegression(max_iter=1000)
model_title.fit(X_train_title, y_train_title)
y_pred_title = model_title.predict(X_test_title)

# Output evaluation indicators
print("Classification effect using title only：")
print(classification_report(y_test_title, y_pred_title))
print("Confusion Matrix：")
print(confusion_matrix(y_test_title, y_pred_title))

# Use text only to classify
vectorizer_text = TfidfVectorizer(max_features=5000, ngram_range=(1,
2))
X_text = vectorizer_text.fit_transform(df['clean_text'])
X_train_text, X_test_text, y_train_text, y_test_text =
train_test_split(
    X_text, y, test_size=0.2, random_state=42
)

model_text = LogisticRegression(max_iter=1000)
model_text.fit(X_train_text, y_train_text)
y_pred_text = model_text.predict(X_test_text)

print("\n Classification effect using text only：")
print(classification_report(y_test_text, y_pred_text))
print("Confusion Matrix：")
print(confusion_matrix(y_test_text, y_pred_text))

def plot_confusion(cm, labels, title, filename):
    plt.figure(figsize=(6, 5))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=labels, yticklabels=labels)
    plt.xlabel('Predicted value')
    plt.ylabel('True value')
    plt.title(title)
    plt.tight_layout()
    plt.savefig(filename, dpi=300, bbox_inches='tight')
    plt.show()

# Call the function and save the image
plot_confusion(cm_title, labels, "Confusion matrix using only titles",
"conf_matrix_title.png")
```

```
plot_confusion(cm_text, labels, "Confusion matrix using only the
text", "conf_matrix_text.png")

# Extract keywords and their corresponding weights
feature_names = vectorizer.get_feature_names_out()
coef = model.coef_[0]

top_n = 20

# Negative weights
top_fake_idx = np.argsort(coef)[:top_n]
top_fake_words = [feature_names[i] for i in top_fake_idx]
top_fake_weights = [coef[i] for i in top_fake_idx]

# Positive weights
top_true_idx = np.argsort(coef)[-top_n:]
top_true_words = [feature_names[i] for i in top_true_idx]
top_true_weights = [coef[i] for i in top_true_idx]
# Visualization: Fake News Keywords
plt.figure(figsize=(10, 6))
plt.barh(top_fake_words[::-1], np.abs(top_fake_weights[::-1]),
color='orange')
plt.xlabel("Weight (negative: predictive of fake news)")
plt.title("Top keywords most predictive of FAKE news (title + text)")
plt.tight_layout()
plt.savefig("keywords_fake_news.png", dpi=300)
plt.show()

# Visualization: Real news keywords (positive weight)
plt.figure(figsize=(10, 6))
plt.barh(top_true_words, top_true_weights, color='green')
plt.xlabel("Weight (positive: predictive of true news)")
plt.title("Top keywords most predictive of TRUE news (title + text)")
plt.tight_layout()
plt.savefig("keywords_true_news.png", dpi=300)
plt.show()
```