Fake news detection with Natural Language Processing

Dmitri Lebrun
ENSAE Paris
dmitri.lebrun@ensae.fr

Abstract

Fake news has a massive negative impact on today's society. For this reason, various methods have been developed to detect and remove it from public view. Machine learning techniques—particularly those based on Natural Language Processing (NLP)—are especially promising in this context, as they can handle the enormous volume of content produced by social media platforms and news outlets. However, while these methods often perform well on specific datasets, they tend to struggle when applied to real-world data, which involves a wide range of languages and topics. This paper aims to evaluate the generalizability of several NLP approaches.

1 Introduction

The proliferation of fake news has emerged as a significant societal issue, particularly in recent years, due to its rapid dissemination through social media platforms. These platforms, characterized by their vast reach (5.2 billion users worldwide [1]), serve as a fertile ground for the spread of misinformation. Research has shown that fake news is widely spread on social networks [2]. According to the Pew Research Center, 48% of the American population now gets their news from social media, underscoring the central role these platforms play in shaping public opinion [3]. Given this widespread reliance on social media for news consumption, the detection and mitigation of fake news is more important than ever to ensure an accurately informed public.

Many methods have been tested to detect fake news but most of them struggle with generalizability, meaning they have a good accuracy on the dataset they were trained on but when faced with completely new data, the results become a lot worse. In this short article, we will try several feature extraction methods, train them using a gradient boosting method on a specific dataset and see how they perform on a completely new dataset. These feature extraction methods are: bag-of-words, Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, BERT and linguistic cues extraction.

2 Description of the state-of-the-art

Traditional NLP models, such as those based on token representations like TF-IDF and BERT, often perform well on usual fake news datasets but struggle to generalize well on new data. This issue arises due to biases introduced by coarsely labeled training data, where articles are labeled based on their publisher, leading to overfitting and poor performance on unseen data. For this reason different methods have been tested to improve generalizability such as the extraction of elements like stylistic features, semantic attributes and even a "social-monetisation' feature estimating the economic incentive behind each news (number of advertisments, sharing links on various social media...) [4].

Designing robust techniques for fake news detection can also be done by integrating multiple modalities, such as text, images, and videos. These various sources of data, can be leveraged to use ensemble techniques that combine textual analysis using BERT and image analysis using CNN's.

Preprint. Under review.

This makes the model more generalizable as it becomes more flexible to analyze information coming from various data sources [5].

Other researchers improve generalizability by using more traditional techniques like data augmentation. They are several ways to performs these : replace or remove entities, events or images in articles or creating fake samples. This creates more diverse data that improve the generalization capabilities of machine learning models [6].

Several specific models have also been proposed for this task. The Entity-centric Multi-domain Transformer (EMT) finds recurrent elements in news samples to learn domain-invariant and domain-specific news representations. This allows the model to perform fake news detection on a very large variety of domains and adapt to completely new data even in domains where it has seen a small number of samples [7].

Some authors have also used Large language models paired with domain classification to improve the generalization ability in each domain. This also limits a recurrent problem with large multi-domain models, which is that imbalanced datasets limit their ability to learn fake news structure on all the domains. For instance, the Llama 2 model can be fine tuned to detect fake news in specific domains by detecting the characteristics of these fake news in each field [8].

3 Description of the feature extraction methods

First, we describe the five feature extraction methods we used and which we then use as input for the gradient boosting classifier model.

3.1 Bag-of-words (BoW)

The first feature extraction method we implement is the bag-of-words method. The BoW model is a vectorization technique that transforms a corpus of text documents into fixed-length feature vectors by representing each document as a multiset of its words, disregarding grammar and word order but preserving multiplicity. Given a corpus of D documents and a vocabulary

$$V = \{w_1, w_2, \ldots, w_n\}$$

extracted from the corpus, BoW constructs an $|D| \times |V|$ matrix $\mathbf{X}$, where each entry x_{ij} denotes the frequency of word w_j in document d_i.

Thus, BoW takes into account the frequency of appearance of words but not the order of appearance or the place a certain word has in a given document.

3.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical measure that, like BoW, aims to evaluate the importance of a word in a document relative to a corpus. Unlike BoW however, which simply counts how often a word occurs in a document, TF-IDF adjusts this count based on how common the word is across the entire corpus. Thus, it reduces the weight of frequently occurring but less informative words such as 'the', 'a', 'and'...

Given a corpus of D documents and a vocabulary $V = \{w_1, w_2, \ldots, w_n\}$, the TF-IDF weight for word w_j in document d_i is defined as:

$$\text{TF-IDF}_{ij} = \text{TF}_{ij} \cdot \text{IDF}_j$$

where $\text{TF}_{ij} = \text{count}(w_j, d_i)$ is the term frequency of word w_j in document d_i (so the number of times the word appears in the document divided by the number of words in the document), and the inverse document frequency is given by:

$$\text{IDF}_j = \log\left(\frac{|D|}{1 + \text{DF}_j}\right)$$

Here, DF_j is the document frequency, i.e., the number of documents in which the word w_j appears, and $|D|$ is the total number of documents in the corpus. The addition of 1 in the denominator prevents division by zero. Thus, if a word appears in many documents, its weight is reduced.

The resulting TF-IDF score increases proportionally with the number of times a word appears in a document but is offset by the frequency of the word in the entire corpus.

3.3 Word2Vec

Word2Vec is a neural-network-based model for learning continuous vector representations of words. It captures semantic and syntactic relationships based on word context. Unlike Bag-of-Words or TF-IDF, which produce sparse and high-dimensional representations, Word2Vec embeds each word $w \in V$ into a low-dimensional vector space $\mathbb{R}^d$, where $d \ll |V|$, such that words with similar contexts appear close together in that space.

In this paper, we use the Skip-gram version of Word2Vec which works as follows.

Given a sequence of training words $w_1, w_2, \ldots, w_T$, the Skip-gram model aims to maximize the average log probability:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j} \mid w_t)$$

where c is the context window size (five in our implementation), and $P(w_{t+j} \mid w_t)$ is the probability of observing the context word w_{t+j} given the center word w_t.

This conditional probability is modeled using the softmax function:

$$P(w_O \mid w_I) = \frac{\exp(\mathbf{v}'_{w_O} \cdot \mathbf{v}_{w_I})}{\sum_{w \in V} \exp(\mathbf{v}'_w \cdot \mathbf{v}_{w_I})}$$

where $\mathbf{v}_{w_I}$ is the input (center word) vector, $\mathbf{v}'_{w_O}$ is the output (context word) vector, and V is the vocabulary.

To reduce computational complexity, training often employs optimization techniques such as negative sampling, where only a small number of negative samples are used to approximate the softmax denominator.

Once training is complete, each word $w \in V$ is associated with a dense vector $\mathbf{v}_w \in \mathbb{R}^d$, where d is the dimensionality of the embedding space. To obtain document-level embeddings the individual word vectors are averaged over each document.

This averaged embedding $\mathbf{d}$ is used as a fixed-length feature vector representing the document, which can then serve as input for our gradient boosting algorithm.

Thus, the result of training is a word embedding matrix $\mathbf{W} \in \mathbb{R}^{|V| \times d}$, where each row corresponds to a word vector.

3.4 Bidirectional encoder representations from transformers (BERT)

The BERT model is used to extract contextualized embeddings for our input texts. In our implementation, we use the model bert-base-uncased, a pre-trained transformer with 12 layers and 110 million parameters. It has been trained on a large English corpus of documents.

The input texts are tokenized and encoded using the tokenizer. These are then passed to the BERT model, and the resulting tensor (called `last_hidden_state` in the implementation) contains contextual embeddings of all tokens for each input sequence.

The embedding used to represent each text is derived from the output vector of the special `[CLS]` token, which the BERT model inserts at the beginning of every sequence. This vector is intended to aggregate information about the entire sequence. Mathematically, if the output of the last hidden layer is

$$\mathbf{H} \in \mathbb{R}^{B \times T \times d}$$

where B is the batch size (32 in our case), T is the sequence length, and d is the hidden dimension (768 for `bert-base-uncased`), then the embedding for each input is:

$$\mathbf{e}_i = \mathbf{H}_{i,0,:}$$

where $\mathbf{H}_{i,0,:}$ corresponds to the embedding of the `[CLS]` token for the i-th input in the batch.

The extracted embeddings $\mathbf{e}_i \in \mathbb{R}^d$ are stacked across all inputs to form a matrix $\mathbf{E} \in \mathbb{R}^{N \times d}$, where N is the number of input texts. These fixed-length embeddings can then be used directly as input for our model.

3.5 Linguistic cue extraction

Finally, we use a feature extraction method more specific to the article we base our work on. This method consists of extracting a large number of linguistic information from each sample such as the word count, the syllable count or more specific information such as the percentage of common verbs (like "run", "walk", "swim", "go", "come", "make", "do"...) in a given sample. More than 20 features of this type are extracted. The full list can be found in TABLE 1 of the article that inspired this paper [9].

In practice, we removed the sentiment polarity measure and a few other features as they took a significant time to be computed and we obtained good results without these information.

4 Description of the two datasets

4.1 The ISOT dataset for learning

The ISOT Fake News Dataset is a publicly available corpus compiled by the University of Victoria's Information Security and Object Technology (ISOT) research lab, specifically designed to facilitate the training and evaluation of models for fake news detection. The dataset is structured for binary classification and is composed of labeled news articles, categorized as either fake or real.

The real news come from respected news sources such as Reuters, the biggest British news agency. The fake article were scrapped from unreliable news websites flagged for spreading misinformation.

The dataset contains 44,898 articles and it relatively balanced between the two classes (21,417 real news and 23,481 fake news).

It is worth noting that all the articles in the dataset are in English.

Thus, this dataset was analyzed using the five feature extraction techniques we saw in the previous section and used to train the gradient boosting algorithm model.

4.2 The Fake or Real News data to test generalization

This dataset has a similar structure to the one described in the previous question. It contains 6335 news articles and is also close to being balanced between the two classes ; it contains 3164 fake news articles and 3171 real articles coming from reliable news outlets.

This dataset will be used to test the generalizability of the feature extraction techinques and model we use.

5 The Gradient boosting algorithm model

Gradient Boosting is an ensemble machine learning technique that builds a strong predictive model by combining multiple weak learners, decision trees, in a sequential manner. Gradient boosting constructs trees iteratively ; each new tree is trained to correct the residual errors made by the ensemble of previously trained trees.

More formally, given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ are input features and y_i are the corresponding targets, the model builds an additive predictive function:

$$F_M(x) = \sum_{m=1}^{M} \gamma_m h_m(x)$$

where $h_m(x)$ is the m-th weak learner (e.g., a decision tree), and γ_m is a step size or learning rate.

The training process proceeds as follows:

1. Initialize the model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma)$$

2. For $m = 1$ to M, where M is the number of boosting rounds:
   (a) Compute the pseudo-residuals:
   $$r_i^{(m)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F=F_{m-1}}$$
   where L is the loss function.
   (b) Fit a weak learner $h_m(x)$ to the residuals $r_i^{(m)}$
   (c) Compute the optimal step size:
   $$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{N} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right)$$
   (d) Update the model:
   $$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

6 The results

For each feature extraction technique we used the feature vector to train our model. We now analyze our results on both the dataset used for training and the new data.

6.1 Results on the ISOT dataset

Table 1: Performance Comparison with the gradient boosting model

Method	Accuracy	Precision	Recall	F1 Score
Bag of Words	0.9935	0.9900	0.9970	0.9935
TF-IDF	0.9925	0.9880	0.9970	0.9925
Word2Vec	0.9855	0.9830	0.9879	0.9854
BERT	0.9720	0.9718	0.9718	0.9718
Linguistic	0.9595	0.9635	0.9547	0.9591

The results are written in the table above. First, we notice that all the feature extraction methods yield great results in all the metrics. We also notice that these results are close to the ones presented in the article this study is based on [9].

More precisely, Bag-of-Words and TF-IDF yield the best results with a 99% or more value in almost all of the metrics. Word2Vec is slightly lower with a 98% value across all metrics, BERT obtains a 97% score and the linguistic approach yields the "worst" results with around 95%.

It is interesting to notice that the most complex models yield slightly worse results than the most basic ones. This may indicate that the data is very simple and that the fake news and real news in the dataset have very distinct characteristics.

The first two feature extraction methods may thus be overfitting on the ISOT dataset. We will see that the results of the next section tend to confirm that theory and research online confirms that in this dataset the fake news articles are often stylistically very different from the real ones, which may make classification easier than in real-world settings [10].

6.2 Generalization results

Table 2: Performance Metrics on the new data

Method	Accuracy	Precision	Recall	F1 Score
Bag of Words	0.5122	0.6738	0.0495	0.0922
TF-IDF	0.5121	0.6835	0.0470	0.0879
Word2Vec	0.5001	1.0000	0.0013	0.0025
BERT	0.6586	0.7238	0.5140	0.6011
Linguistic	0.5634	0.6576	0.2665	0.3793

With these results, we notice, like in the original articles that on completely new data, coming from a different dataset, we obtain significantly worse results. The worst results come from the most basic models (Bag-of-words, TF-IDF, Word2Vec) which achieve around 50% accuracy which is as if they were classifying the samples randomly. This also tends to confirm that these methods lead to overfitting on the ISOT dataset.

BERT, the most complex model we used for feature extraction, gets decent results on this new data achieving 65% accuracy and 72% precision. This tends to show that BERT is able to extract more complex meaningful features from the ISOT dataset.

Finally, the linguistic cues feature extraction method achieves slightly better than random results which shows that this method tends to find deep characteristics of fake and real news samples.

7 Conclusion

Thus, generalization is a true challenge for fake news detection models. Models can almost perfectly learn the characteristics of the original dataset but yield significantly worse results on new data.

This shortcoming can be explained by the structure of the training dataset, that takes all its real news articles from a limited numbers of sources while aggregating fake news from many different news outlets. This makes the real data distinct from the fake one.

Overcoming this can be done by training the models on more diversified dataset, presenting data from a large numbers of news outlets.

Another solution, presented in the state-of-the-art review section of this paper could be to train models on specific fields (sport, politics, culture...) to make sure they capture domain-specific characteristics of fake news.

References

[1] Stefan Larson. Social media users 2025 (global data statistics), 2025. `https://prioridata.com/data/social-media-usage/`.

[2] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[3] Pew Research Center. News consumption across social media in 2021, 202. `https://www.pewresearch.org/journalism/2021/09/20/news-consumption-across-social-media-in-2021/`.

[4] Theodora Koulouri Nathaniel Hoy. An exploration of features to improve the generalisability of fake news detection models. *Expert Systems with Applications (Elsevier)*, 2025.

[5] Chunmei Liu Mohammed Al-alshaqi, Danda B.Rawat. Ensemble techniques for robust fake news detection: Integrating transformers, natural language processing, and machine learning. *Sensors 2024*, 2024.

[6] Ralph Ewerth Eric Müller-Budack Sahar Tahmasebi, Sherzod Hakimov. Improving generalization for multimodal fake news detection. *ICMR 2023*, 2023.

[7] Azadeh Shakery Abbas Maazallahi Parisa Bazmi, Masoud Asadpour. Entity-centric multi-domain transformer for improving generalization in fake news detection. *Information Processing and Management: an International Journal*, 2024.

[8] Mohammad Qasim Mohammad Alnabhan. *Advancing Cross-Domain Fake News Detection: Enhanced Models to Improve Generalization and Tackle the Class Imbalance Problem*. PhD thesis, University of Ottawa, 2025.

[9] Theodora Koulouri Nathaniel Hoy. Exploring the generalisability of fake news detection models. *IEEE International Conference on Big Data*, 2022.

[10] Abdullah Marish Ali Fuad A. Ghaleb Bander Ali Saleh Al-Rimy Fawaz Jaber Alsolami, Asif Irshad Khan. Deep ensemble fake news detection model using sequential deep learning technique. *Sensors 2022*, 2022.