Fake News Detection
A Survey of Text Classification Methods

Gilles Bou-dalha-ghoussoub
Msc. Data Science and Analytics
HEC Montreal

Abstract

This project investigates and compares state-of-the-art deep learning models for fake news detection, with a focus on transformer-based architectures that have transformed natural language processing in recent years. The study evaluates models such as the BERT family and advanced open-source large language models, including IBM's Granite, across various configurations, including model sizes, hyperparameter settings, and prompting strategies. By analyzing their performance on the WELFake dataset, we identify the most effective architectures for distinguishing fake news from legitimate articles. Furthermore, this work highlights the significant advancements deep learning has achieved in this domain, surpassing the capabilities of traditional machine learning methods.

1 Introduction

1.1 Motivation

The spread of fake news online has become a major concern, with significant social, political, and economic consequences. False information can shape public opinion, influence elections, and disrupt societal trust. Given the vast scale and speed at which fake news can proliferate, it is essential to develop automated tools that can accurately classify news articles as real or fake. Deep learning techniques have shown great promise in addressing this issue by leveraging large datasets and powerful models to improve classification accuracy. This project aims to evaluate and compare the state-of-the-art methods in fake news detection using the WELFake dataset and determine the most effective models for this task.

1.2 Definition

The central focus of this work is to evaluate and compare different techniques for fake news detection, with an emphasis on state-of-the-art architectures highlighted in recent literature. In particular, transformer-based models such as BERT and IBM's Granite are explored due to their demonstrated success in natural language processing tasks. The study aims to identify the most effective approaches for distinguishing between real and fake news, leveraging the WELFake dataset as a benchmark.

A key aspect of this research is the prioritization of computational efficiency while achieving high accuracy. High computational demands often translate into increased monetary expenses, limiting the accessibility of these models for smaller organizations. Additionally, the environmental impact of large-scale model training and inference has raised concerns.

Foundation Models for Science Workshop,38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Related Work

Text classification is a foundational task in natural language processing (NLP) underpinning various applications such as fake news detection. Over the decades, research in this domain has progressed significantly, driven by advances in feature representation techniques, classification algorithms, and computational power.

2.1 Traditional Methods

Early work in text classification relied heavily on statistical and probabilistic methods. Naïve Bayes, introduced in early NLP systems, became a foundational probabilistic model for its simplicity and efficiency in classifying textual data. Support Vector Machines gained prominence for their robustness in handling high-dimensional data and sparse representations like term frequency-inverse document frequency [Vapnik, 1998]. However, these methods heavily relied on manual feature engineering, such as bag-of-words, which limited their ability to capture semantic relationships within the text [Kowsari et al., 2019].

2.2 Deep Learning Approaches

The introduction of deep learning catalyzed a transformative shift in text classification methodologies. Recurrent Neural Networks and their variants, such as Long Short-Term Memory networks, addressed sequential dependencies in text data [Hochreiter and Schmidhuber, 1997]. CNNs, traditionally used for image data, were successfully adapted for text classification tasks [Kim, 2014], demonstrating their ability to identify local features within text windows.

2.3 Transformer-Based Models

A revolutionary step came with the development of transformers, introduced in Attention is All You Need [Vaswani et al., 2017]. Transformers replaced traditional recurrence-based mechanisms with self-attention, enabling models to capture global dependencies in text efficiently and to parallelize computations. BERT (Bidirectional Encoder Representations from Transformers) further advanced the field by pre-training models on vast corpora and fine-tuning them for specific tasks [Devlin et al., 2018], setting new benchmarks across text classification tasks.

Large Language Models, such as GPT (Generative Pre-trained Transformer) and their successors, have continued this trajectory by scaling transformers to unprecedented sizes and capabilities [Radford et al., 2018]. Different prompting techniques, such as few-shot prompting and chain-of-thought reasoning, have proven to be effective for in-context learning, enabling models to adapt to a wide range of tasks without additional fine-tuning. These advances highlight the versatility of large language models in addressing complex problems with minimal task-specific modifications.

2.4 Prevalent Datasets

The progression of text classification methods has been accompanied by the development of benchmark datasets, such as the 20 Newsgroups dataset, AG News, and FakeNewsNet, each presenting unique challenges like imbalanced class distributions or limited diversity. Recent efforts like the WELFake dataset [Pérez-Rosas et al., 2023] aim to address these issues by aggregating multiple sources, creating a more comprehensive and diverse benchmark. Despite these advances, deep learning-based methods face challenges, including high computational costs and the need for large, labeled datasets.

2.5 Contribution

This project builds on prior research by leveraging state-of-the-art transformer architectures and exploring their application to fake news detection using the WELFake dataset. By analyzing the trade-offs in model performance and computational requirements, we aim to provide insights into the practical implications of these advancements.

3 Methodology and Analysis

We begin by establishing a classical method as a baseline to highlight the advancements brought by deep learning. This approach allows readers to appreciate the significance of recent innovations in the field. Subsequently, we evaluate the performance of state-of-the-art methods, including transformer architectures and large language models (LLMs), as identified in related work. The report also documents our experiences with implementing these models, the challenges encountered, and the results obtained for the fake news detection task.

To ensure a fair comparison, all models are trained on the same dataset and evaluated using a consistent validation set. We minimize hyperparameter tuning on the validation set to maintain objectivity and avoid introducing bias. Performance metrics throughout this report reflect results obtained exclusively on the validation set.

Working with deep learning methods and large textual datasets presented significant computational challenges. Our dataset consists of over 70,000 articles, each containing up to 2,000 words, which exceeds the capacity of most local computational resources and poses limitations for free cloud services such as Google Colab. Additionally, training large transformer models from scratch is computationally intensive, and loading large language models into the limited GPU memory available on Colab proved restrictive. Thus, to address these issues we adopted a limited dataset to the first 10,000 articles and we used a hosted large language model via replicate API.

3.1 Traditional Methods

3.1.1 Pre-processing

We applied several text pre-processing techniques to reduce dimensionality while preserving key information. These included cleaning text with regular expressions to retain only alphabetical characters, filtering by frequency to remove rare and overly common words, and removing stop words that don't contribute meaning. Additionally, we employed stemming to normalize words to their root forms. For topic modeling (next section), we used part-of-speech (POS) tagging to retain nouns, verbs, and adjectives, which are critical for capturing the core meaning of a sentence.

3.1.2 Feature Extraction

Feature extraction involves converting textual data into numerical representations, a necessary step for applying machine learning techniques. In this work, we utilized three common methods: Bag of Words, TF-IDF, and Word Embeddings.

Bag of Words: This method counts the frequency of each word in a document without considering word order. While simple and easy to implement, BoW can lead to sparse high-dimensional feature spaces, which may reduce model efficiency.

TF-IDF: Building upon BoW, TF-IDF adjusts the weight of words by considering both their frequency within individual documents and their occurrence across the entire corpus. While terms that appear frequently within a document are given higher importance, the method reduces the weight of words that appear widely across many documents, as they are likely less informative and more generic.

Pre-trained word embeddings: Approaches like Google's Word2Vec embed words in a dense continuous vector space that captures their semantic. Word2Vec uses models like Continuous Bag-of-Words (CBOW) or Skip-Gram to capture word context, providing a more semantically meaningful representation than BoW.These embeddings are based on the distributional hypothesis and are pre-trained on large, general corpora. As a result, they may be less optimal for our particular task and context.

3.1.3 Classifiers

Support Vector Machines (SVM) are utilized for their robustness in binary classification tasks. To optimize the performance of the SVM classifier, we employed grid search hyperparameter optimization techniques, exploring multiple settings. Key hyperparameters include C (the penalty for misclassification), gamma (defines the influence of a single training example), and kernel type (linear, poly, rbf).

Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) units are well-suited for sequence data such as text. LSTM alleviates the vanishing gradient problem by retaining important words with longer-term relevance.

Table 1: Performance of traditional methods

Model	Accuracy (%)
Random	50.53
SVM (BOW)	88.91
SVM (TFIDF)	90.76
SVM (WORD2VEC)	85.11
RNN (LSTM)	85.11

3.2 Bidirectional Encoder Representations from Transformers

In this experiment, we fine-tuned and evaluated three transformer models from the BERT family—BERT, DistilBERT, and TinyBERT—to classify fake news articles. The goal was to balance predictive accuracy and computational efficiency across the models.

BERT[Devlin et al., 2019], the original transformer model, was fine-tuned using the bert-base-uncased architecture. This model is known for its high performance on a variety of NLP tasks, including text classification. However, due to its size, it can be computationally expensive and slower during inference.

DistilBERT[Sanh et al., 2019], is a distilled version of BERT that reduces the model size by approximately 60% while retaining about 97% of the performance of the original model. Its smaller size and reduced computational cost make it a practical choice for real-time applications where speed is a critical factor.

TinyBERT [Jiao et al., 2020] is an even smaller version of BERT, optimized for mobile and edge devices with severe resource constraints.

3.2.1 Fine-tuning BERT

For both BERT and TinyBERT, standard hyperparameter configurations were used, adopting an "out-of-the-box" fine-tuning approach to showcase the performance achievable with minimal adjustments. This approach shows the baseline capabilities of these models. In contrast, DistilBERT underwent more detailed experimentation with different hyperparameter settings. The aim was to demonstrate that a lightweight model like DistilBERT could deliver competitive results. The optimal configuration for DistilBERT, identified through these experiments, is described in detail below.

First, gradient accumulation was applied over 4 batches of 4 examples, reducing memory usage while maintaining stable learning. Mixed precision (fp16) training was then used to accelerate the process and further minimize memory requirements. The model was trained for 10 epochs, with early stopping if performance declined for 2 consecutive epochs. The AdamW optimizer, incorporating weight decay to prevent overfitting, was employed, along with a linear warm-up learning rate to ensure smooth training progression and preserve previously learned parameters.

To optimize training efficiency, the model's weights were initially frozen, and only the parameters of the classifier head were updated during the first 10 epochs. This approach led to rapid performance improvements, achieving an accuracy of 87% with minimal computations. Following this phase, the entire model was fine-tuned for an additional 10 epochs using the same hyperparameter setup. This strategy allowed for faster convergence during full model fine-tuning, demonstrating the effectiveness of pretraining the classifier head.

3.2.2 Performance BERT

Despite being trained for 10 epochs, all models achieved over 90% accuracy after just one epoch of fine-tuning. Notably, the Base BERT model reached an impressive 96.05% accuracy after the first epoch, highlighting the diminishing returns of extended training. This observation suggests that a single epoch of fine-tuning is sufficient to achieve strong baseline performance.

Table 2: Performance of BERT models

Model	Accuracy (%)	Size (# parameters)	Training time
TinyBERT	96.95	4M	7 min
DistilBERT	98.65	65M	18 min
Linear probing	87.00	65M	8 min
BaseBERT	98.50	108M	30 min

For example, TinyBERT achieved competitive accuracy within a training time of less than a minute on a T4 GPU. This performance surpasses traditional models without requiring any preprocessing or feature extraction. These results underscore the advancements introduced by transformer architectures, particularly the mechanisms of attention, parallelism, and automatic feature learning

3.3 Large Language Models

The goal of this experiment is to evaluate the performance of the Granite 3.0-8B large language model on the task of fake news classification. Granite 3.0-8B is a state-of-the-art language model developed by IBM that leverages transformer architecture to process and generate text. The model can perform a wide range of NLP tasks, including content classification, question answering, and text generation. Its large-scale training on diverse corpora enables it to excel in few-shot learning tasks, where minimal task-specific examples are required to achieve high performance [IBM Research, 2023].

To achieve this, we employed three prompting techniques: zero-shot, few-shot, and expert-informed prompting. Given our limited expertise in fake news detection, we utilized topic modeling methods to identify key topics that could aid in distinguishing between real and fake news. These extracted topics served as the foundation for constructing expert-informed prompts. Importantly, topic modeling was performed exclusively on the training set to avoid any information leakage into the validation set, ensuring the integrity of our evaluation.

3.3.1 Topic Modeling

BERTopic is a topic modeling technique that uses transformer-based embeddings to generate contextual representations of documents. These embeddings are reduced in dimensionality using UMAP and clustered with HDBSCAN to form groups of semantically similar documents. For each cluster, representative keywords are extracted using scoring methods like TF-IDF, creating interpretable topics. The model optionally merges similar topics based on cosine similarity between topic embeddings. Key hyperparameters include the pre-trained transformer embedding model, UMAP parameters (e.g., n neighbors and min dist), HDBSCAN parameters (e.g., min cluster size), and settings for topic representation and merging. The hyperparameters n neighbors, min dist, and min cluster size affect BERTopic's results. Smaller n neighbors create tighter, fine-grained clusters, while larger values produce broader, more general clusters. Smaller min dist results in compact clusters, and larger values spread clusters out. Smaller min cluster size allows more, smaller topics, but larger values ensure robust, larger clusters, possibly ignoring small ones. BERTopic outputs interpretable topic-keyword pairs, document-topic probabilities, and interactive visualizations, making it a powerful tool for contextual topic discovery.

We were able to identify the following main topics among others :

Climate Change: Misinformation campaigns have falsely asserted that climate change is a hoax perpetrated by scientists for financial gain.

Taxes: Fake news stories have claimed that certain tax reforms would lead to massive tax increases for middle-class families, despite evidence to the contrary.

Healthcare: False claims circulated that the Affordable Care Act ("Obamacare") included provisions for "death panels" to decide end-of-life care for patients.

Iran Nuclear Deal: Fabricated stories alleged that the Iran nuclear deal included secret clauses allowing Iran to develop nuclear weapons.

3.3.2 Information Extraction

We tried to do information extraction to recognize the main words that are mostly associated or differentiate the two classes of fake news and real news. We used TF-IDF representation along with chi-square method. The chi-square test assesses whether observed categorical data deviates significantly from what is expected under the hypothesis of no association between classes. To apply it, we create a contingency table listing observed frequencies of categories (like attribute levels) across two classes. We calculate expected frequencies for each table cell assuming no class-attribute relationship. The chi-square statistic is then computed by summing the squared differences between observed and expected frequencies, each divided by the expected frequency. A significant test result indicates that the attribute significantly discriminates between the two classes.

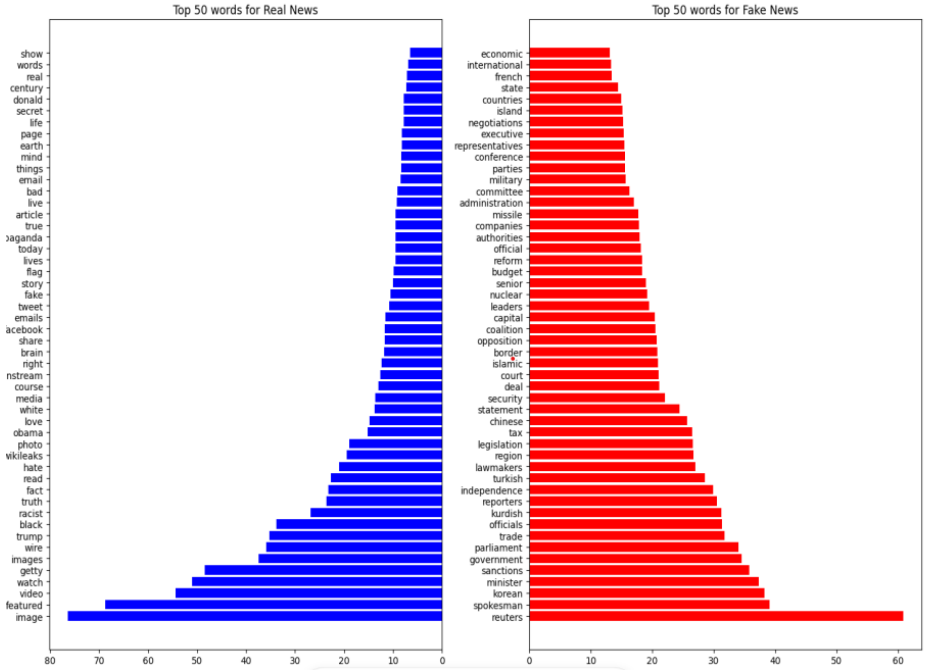

Figure 1: Chi-Square Information Extraction

From the graph, we interpreted the results as follow: the real news are based on evidence or online platforms that can be investigated (either images, videos, features, Facebook, Twitter etc.) or containing words identifying the fake news (propaganda, fake etc.), or affirming the facts (truth, true, fact, real), other words may have specific topics associated to them which are hard to interpret with just words without context. On the other hand, the highest chi-square score was for Reuters (a reliable news agency) which led us to believe that a lot of fake news associate themselves with real news agency to try to legitimize their fake news and make them believable. The other words are very similar to the ones found in topic modeling and touch upon national security, politics, law and legislation etc.

3.3.3 Granite

As previously mentioned, the large language model used in this section is IBM's Granite 3.0-8B, an instruct model accessed via the Replicate API. For all in-context learning strategies, a system prompt was designed to explicitly guide the model's behavior:

"You are an expert in fake news detection, and your task is to classify articles as either 'Real' or 'Fake'. Only respond with 'Real' or 'Fake'."

This approach helped minimize the potential for creative or unintended outputs, ensuring that the model's responses were strictly aligned with the binary classification task. To manage API rate limits and prevent service disruptions, a delay was introduced between API calls.

Given the costs associated with using a hosted model, we limited the validation dataset to 500 examples, compared to 2000 examples for the previous models. This reduction was necessary because the articles in the dataset are lengthy, and model usage incurs charges based on the number of input tokens and generated tokens.

To further reduce costs, several techniques were implemented. The number of generated tokens was capped at 5, as the expected response is a simple binary classification ("Real" or "Fake"). For few-shot learning, only a small set of examples was provided. Additionally, to minimize sampling diversity and ensure consistent outputs, the model's temperature was set to zero, forcing the model to select the most probable response. The top-k parameter was also restricted to k=10 to reduce randomness and enhance response consistency.

For the one shot prompting, the provided prompt is as follows:

```
The following are examples of news articles labeled as Real or Fake.

Classify the given article as either "Real" or "Fake."

Example 1:
Article: "NASA has discovered a new planet that may be habitable."
Label: Real

Example 2:
Article: "Scientists say drinking bleach cures COVID-19."
Label: Fake

Classify the given article as either "real" or "fake."
Article: "{article_text}"
Label:
```

The expert informed prompt is the following:

```
The following are clues to detect whether news articles are real or fake.

Articles related to politics
(mainly in USA and middle eastern and asian countries)
or national security and defense or terrorism
or climate change or those coming from Reuters
have higher chance to be fake.

Articles that cite reputable sources
and have image and video evidence, are more likely to be real.

Now classify this article:
Article: "{article_text}"
Label:
```

3.3.4 Performance Granite

The model's performance was evaluated using accuracy as the primary metric, comparing predictions to ground truth labels. However, the task presented unforeseen challenges as the model occasionally generated ambiguous outputs (e.g., incomplete or unrelated responses). Three response categories emerged:

"Real": Correctly labeled as real news.

"Fake": Correctly labeled as fake news.

Ambiguous: Responses that deviated from the binary "Real" or "Fake" classification.

Ambiguous predictions constituted approximately 15% of the total. Performance was assessed under two scenarios:

Ambiguous Responses Mapped to "Fake": Accuracy was calculated by assigning all ambiguous predictions to the "Fake" category, resulting in an accuracy of 24.11%.

Ambiguous Responses Ignored: Ambiguous predictions were excluded from the dataset, yielding an accuracy of 22.70%.

Interestingly, a detailed inspection revealed an apparent label inversion, where the model frequently mislabeled real news as fake and vice versa. This phenomenon might stem from the selected few-shot examples being non-representative of their respective classes. By recalibrating the results to account for this inversion, the adjusted accuracies were 75.89% for the mapped scenario and 77.30% for the ignored scenario. These recalibrated results suggest a reasonable level of performance, considering that Granite 3.0-8B is a general-purpose LLM not fine-tuned for fake news detection.

Similarly, the same issues were raised in zero shot and expert informed prompting.

Table 3: Performance of Granite

Model	Accuracy (%)
Zero Shot	73.00
One Shot	75.89
Expert Informed	78.88

Despite the modest raw accuracy, the experiment demonstrated Granite 3.0-8B's adaptability to classification tasks using a few-shot learning approach. Its ability to generalize with minimal task-specific training showcases the potential of large language models for text classification. However, a few limitations were noted. First, a significant proportion of ambiguous predictions highlighted the importance of robust prompt design and post-processing. Second, while using a hosted model reduced infrastructure requirement, it introduced dependencies on API availability, latency, and cost limitations. Third, the model's performance is highly sensitive to the quality and representativeness of the few-shot examples, underscoring the need for more systematic example selection.

4 Conclusion

This study investigated state-of-the-art techniques for fake news detection, with a particular emphasis on transformer-based architectures such as BERT, DistilBERT, TinyBERT, and IBM's Granite 3.0-8B. Leveraging the WELFake dataset, we conducted a comprehensive evaluation of these models using a variety of training strategies, including fine-tuning, linear probing, zero-shot, few-shot, and expert-informed prompting.

Our findings highlight the accuracy, flexibility, and ease of use of transformer-based models. Despite their computational demands, these models demonstrated significant improvements over traditional methods by offering superior contextual understanding, eliminating most manual preprocessing steps, and effectively leveraging transfer learning.

Notably, smaller models like TinyBERT showcased strong performance while significantly reducing computational costs and infrastructure requirements. DistilBERT achieved the highest accuracy of 98.65%, though further fine-tuning of BERT could likely get better results.

A key limitation of our work lies in the balance between breadth and depth. While we conducted a broad review of text classification methods and evaluated numerous approaches, we did not devote extensive time to fine-tuning specific models. As a result, our findings serve as general guidelines for out-of-the-box models and techniques rather than highly optimized solutions.

In conclusion, this study demonstrates that fake news classification can be effectively addressed with transformer models from the BERT family, particularly when fine-tuned for the specific task.

Future work could expand this research in several directions. For instance, alternative fine-tuning strategies for BERT, such as parameter-efficient tuning methods like LoRA, could be explored to reduce resource requirements. Additionally, experimenting with other prompting techniques such as chain-of-thought prompting, could provide improved performance. Finally, evaluating the effectiveness of other recent llm like LLaMA 3.3, could uncover new opportunities for improving fake news detection.

References

[1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 4171–4186.

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[3] IBM Research. (2023). Granite 3.0 language models. Retrieved from GitHub Repository.

[4] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. arXiv preprint arXiv:2003.03387.

[5] Kim, Y. (2014). Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751.

[6] Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E. (2019). A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology (TIST), 10(5), 1–41.

[7] Pérez-Rosas, V., Kleinberg, B., Lefevre, J., & Mihalcea, R. (2023). WELFake dataset for fake news detection in text data.

[8] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI Technical Report.

[9] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

[10] Vapnik, V. (1998). Statistical Learning Theory. John Wiley & Sons.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 30, 5998–6008.