Comparisons of LLM and Conventional NLP Techniques in the Detection of Fake News

O'Brien, Desmond
desmond.obrien@ucalgary.ca

Wong, Isaac
isaac.wong@ucalgary.ca

Frempong, Kofi
kofi.frempong@ucalgary.ca

Abstract

The spread of fake news creates serious problems for trust in society and the accuracy of information. With news spreading quickly on digital platforms, it has become increasingly important to tell real news apart from fake. Traditional fake news detection approaches, such as sentiment analysis, rely on analyzing textual content for patterns and emotional cues to identify fake news. However, the advent of Large Language Models (LLMs), such as OpenAI's GPT and similar architectures, offers a more nuanced understanding of language, enabling context-aware classification with greater sophistication.

In this project, we explore the potential of these two approaches—sentiment analysis and LLMs—for fake news detection. Using data sets such as ISOT, LIAR, and WELFake, we analyze their performance on binary classification tasks, where news articles are labeled as either real or fake. Each dataset provides unique features, such as text, titles, and metadata, allowing for a comprehensive evaluation of the techniques.

Our study investigates whether traditional sentiment analysis methods remain competitive in detecting fake news or if LLMs, with their advanced contextual understanding, offer a significant advantage. Through quantitative comparisons and qualitative observations, we aim to provide insights into the strengths and limitations of both approaches and their applicability to real-world scenarios. The following sections will discuss the datasets and methodologies in detail and present our findings.

1 Dataset Overview

1.1 ISOT

The ISOT Fake News dataset ISOT [1, 2], collected from real-world sources, is a binary classification resource designed for fake news detection. It contains two types of articles: real news, sourced from Reuters.com, and fake news, collected from unreliable websites flagged by Politifact and Wikipedia. The dataset includes a range of topics, with a focus on political and world news. Each article is labeled as either real (1) or fake (0), making it ideal for training models to distinguish between authentic and misleading news content.

1.2 LIAR

The LIAR dataset, introduced by William Yang Wang in "Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection" [3], is an annotated source for fake news detection. Each entry in the dataset includes an ID, a unique identifier for the statement; a Label, which classifies the statement into one of six categories (true, mostly true, half true, barely true, false, pants on fire); and the Statement, which is the text of the political claim. Additional metadata includes the Subject(s), detailing the topics of the statement; the Speaker, identifying the individual who made the claim; and the Job Title, which specifies the speaker's role. The dataset also provides State Info, indicating the state associated with the speaker; Party Affiliation, specifying their political party; and a Credit History, tracking the speaker's prior statements in each of the six truthfulness categories (barely true, false, half true, mostly true, pants on fire). Finally, the Context column captures the venue or location where the statement was made.

1.3 WELFAKE

The WELFake dataset sourced from Kaggle [4], is a binary fake news detection resource containing four columns: Title (news heading), Text (news content), and Label (binary classification, where 0 = fake and 1 = real). Originally designed as a multi-category dataset with labels such as true, mostly true, barely true, pants on fire, and false, we processed it into a binary classification problem (real vs. fake) to simplify analysis and focus on distinguishing authentic news from misinformation.

2 NLP-Based Techniques in the Literature

Traditional sentiment analysis and natural language processing approaches to fake news detection can be described as 'content-level' approaches [5]. These methods work by analyzing the natural language contents of fake news articles. Content such as an article's title, publishing agency, article content, and applicable 'tags' can all be considered content-level features of news articles [5]. The traditional approach

to fake news detection using natural language processing involves a specific process designed to extract features from a given fake news dataset.

After acquiring a dataset of labelled fake and real news articles, the first step in any analysis is pre-processing the data. Common pre-processing steps include tokenization (converting a body of text into a discrete set of English words), omitting words that produce little predictive information (stopword) and lemmatization or stemming [5]. Lemmatization is the process of converting individual words (tokens) into a common base-form. For example, after lemmatization, the token 'changes' becomes 'change' [6]. Stemming, though similar to lemmatization, is a slightly different technique. Stemming involves taking a token and removing prefixes and/or suffixes. After stemming, for example, tokens such as 'computing', 'computed' become 'comput' [6]. The purpose of stemming and lemmatization is to take a body of text, also known as a corpus, and reduce it to a smaller subset of individual tokens. This reduces unnecessary information in the dataset and increases the runtime required for feature extraction [6].

Once pre-processing of a fake news dataset has completed, the next step of the analysis is feature extraction [5]. The goal of feature extraction is to mine useful information from a dataset. Useful information includes any characteristics of the dataset that can enhance the accuracy of a classification model [6]. Perhaps the most common form of feature extraction that is used in fake news detection is TF-IDF (Term Frequency-Inverse Document Frequency) [5, 6]. This process involves determining a set of words in a corpus (the entire set of text in a dataset) that have the most significance across all documents [5].

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

Term Frequency (TF) is a document (article) specific measure in a dataset; it is the number of times a specific word appears in an article, divided by the total number of words in the same article [6]. The Inverse Document Frequency (IDF) represents the significance of a specific word across all articles in the dataset. For each word in a dataset, its TF-IDF score can be calculated by multiplying its TF by its IDF. Terms that are frequent in a given document but are infrequent across the full corpus are assigned higher scores [6]. If a specific word exhibits a strong TF-IDF score, and appears almost entirely in fake articles, for example, then finding such a word in an unlabelled dataset would be a strong indication that the article is fake news.

TF-IDF is a very prevalent technique in fake news detection using sentiment-analysis-based approaches. Researchers have applied the technique in fake news classification studies on both ISOT and LIAR datasets [7, 8]. Though the results from the literature showed favorable fake news detection results on the ISOT dataset using TF-IDF extraction [7], the results on the LIAR dataset [8] were less promising. When applying TF-IDF as the only form of feature extraction on the LIAR dataset, the accuracies of fake news classification models in the literature were only marginally better than random guessing [8]. This poor performance of TF-IDF features on fake news classification against the LIAR dataset inspired us to explore different feature extraction approaches for our fake news detection models.

3 Our Approach to NLP-Based Fake News Detection

We propose the inclusion of a new set of features for a fake news classification model: the subjectivities and polarities of content-level information of news articles in a dataset. Polarity is a measure of the sentiment of a statement or body of text. A high polarity represents a positive sentiment, while a low polarity represents a negative sentiment. The subjectivity of a text reflects how personal the assertion of facts within a text is. A subjective body of text will reflect personal opinions and knowledge, while an object article will assert its information is indisputably factual. To aid in extracting these features from natural language datasets, we found two Python libraries: TextBlob [9] and VADER [10].

3.1 TextBlob

TextBlob [9] is a popular Python library designed for natural language processing tasks. Built on top of the powerful Natural Language Toolkit (NLTK) [11] and Pattern libraries, TextBlob employs a pre-trained, lexicon-based approach to analyze the sentiment of a given text. It uses predefined word sentiment scores to classify text as positive, negative, or neutral. For example, positive words (like "great") have positive polarity scores, while negative words (like "terrible") have negative polarity scores [9]. TextBlob's sentiment analysis provides a polarity, which is a continuous score in the range of [-1, 1]. A score of -1 indicates a strongly negative sentiment, while 1 indicates a strongly positive sentiment. Scores near 0 reflect neutral sentiment [9].

3.2 VADER

VADER [10] (Valence Aware Dictionary and Sentiment Reasoner) is a Python library tailored for sentiment analysis, particularly in informal and social media contexts. Like TextBlob, VADER uses a lexicon-based approach with a curated dictionary of over 7,500 words, phrases, and idioms, each assigned a sentiment score. Positive words (e.g., "amazing") have positive scores, while negative words (e.g., "horrible") have negative scores. It also accounts for linguistic nuances such as capitalization, punctuation, and modifiers (e.g., "very"). VADER generates three sentiment scores—negativity, positivity, and neutrality—each ranging from 0 to 1, with their sum always equaling 1. For example, a text might yield 0.33 positivity, 0.33 negativity, and 0.33 neutrality, indicating balanced sentiment [10].

3.3 Sentiment Classification

Before applying polarity and subjectivity features to fake news classification, we first had to ensure that these features exhibited accurate classifications on textual data. To do this, we found two datasets used for sentiment analysis classification. By performing polarity feature extraction on these datasets and training a sentiment classifier model, we could examine the test prediction accuracy that TextBlob and VADER's polarity features produced.

The first dataset we used to test a sentiment classification model is called the 'Large Movie Review Dataset' [12] (IMDB). The dataset is a repository of 50,000 IMDb user-provided reviews for movies and television productions. The dataset provides binary categorization, with each movie review being labelled as positive (1) or negative (0).

The second dataset we sourced for our sentiment classification analysis is titled 'Social Media Sentiments Analysis Dataset' [13] (TWEETS). The dataset consists of 732 short tweets with corresponding sentiment labels. This dataset uses multi-class categorization, with three possible labels for a record: positive (2), neutral (1), and negative (0). We employed our soft-ensemble of models against each dataset, extracting TextBlob and VADER polarity soles as the sole features for the model. The test prediction accuracy of our sentiment classifiers can be seen in the table below.

Dataset	Metric	Polarity (%)	Tf-Idf (%)	Both (%)
IMDB	Test	77.80	82.40	83.60
TWEETS	Test	74.90	82.30	83.30

Table 1: Sentiment classifier accuracies on sentiment datasets

Using polarity features alone, our soft-ensemble model reached roughly 75% on each dataset. These accuracies indicate that the polarity features extracted from a text, using TextBlob and VADER, do provide a degree of predictive strength for sentiment classification. We also tested the use of IF-IDF feature extraction using Sci-Kit Learn's [14] vectorizer implementation. Using TF-IDF feature extraction instead, the test accuracies of our models rose to nearly 82%. By combining both polarity and TF-IDF in our sentiment classifier, we achieved our highest test accuracies of just over 84%. These results show that employing both polarity and TF-IDF feature extraction to our fake news classification models would likely provide the strongest level of classification accuracy.

3.4 Analyzing the Polarity of Fake News Datasets

We began our fake news detection analysis by first examining each dataset. We were initially keen on examining the distribution of real and fake polarity scores across each dataset. The results can be seen in Figure 1 below.

The sentiment scores across all three datasets are largely neutral, usually hovering around 25% positive or negative according to the TextBlob scores. The VADER scores are more

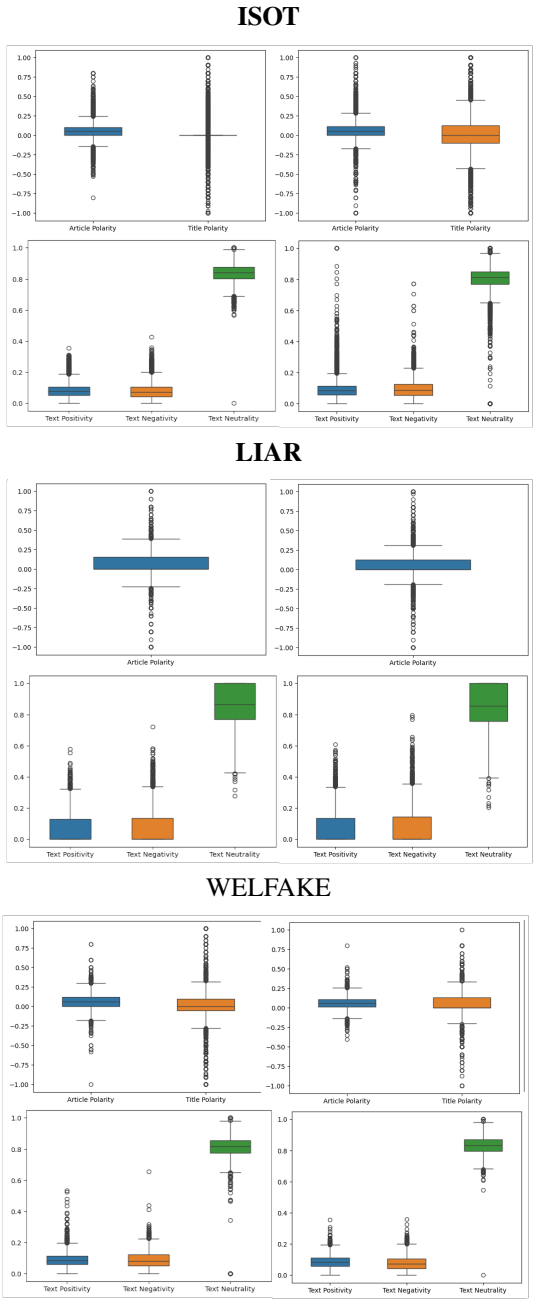


Figure 1: TextBlob (top) and VADER (bottom) polarity Scores for real (left) and fake (right) news articles

restricted, with the average sentiment score of both news article content and titles lying around 80% neutral. It should be noted that the LIAR dataset exhibits a significantly larger variance in sentiment scores as opposed to the ISOT and WELFAKE datasets. The polarity distributions of real and fake articles are largely consistent within each dataset. The clear exception to this is the ISOT dataset, in which the titles of real news articles are overwhelmingly neutral, while fake articles vary from 50% negative to 50% positive; this is according to the polarity scores extracted via TextBlob.

Judging from these distributions, it is clear that the performance of polarity features in fake news classification may

be relatively weaker against the LIAR dataset. The significantly higher level of variation in polarity across news articles is likely to result in a weaker fitting of our model to the dataset. The variation across the WELFAKE and ISOT datasets is largely consistent, but the tight distribution of sentiment scores extracted from news titles in the ISOT dataset may indicate the potential for stronger classification accuracy against the latter.

3.5 Application and Performance of our Conventional NLP Models

The classification accuracy of our conventional NLP model varied across the WELFAKE, ISOT, and LIAR datasets. As was predicted from the analysis in section 2.5, the LIAR dataset proved to be the most difficult dataset to classify. The classification accuracies obtained by our model against each dataset can be seen in Figure 6.

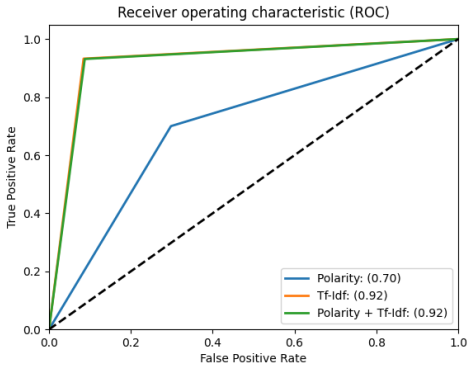
Dataset	Metric	Polarity (%)	Tf-Idf (%)	Both (%)
WFAKE	Validation	88.43	99.47	99.60
	Test	70.10	92.40	92.20
ISOT	Validation	86.56	100.00	100.00
	Test	74.47	99.28	99.26
LIAR	Validation	76.99	90.56	90.72
	Test	62.07	63.17	63.42

Table 2: Classification accuracies of our conventional NLP models across each fake news dataset.

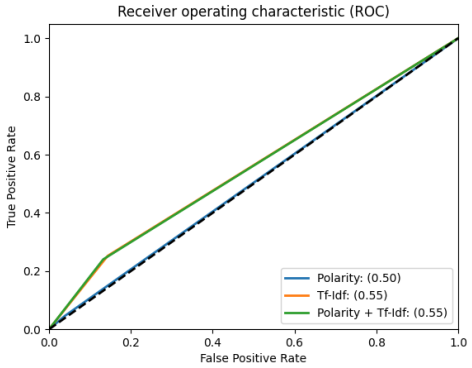
We tested each model using three different feature sets: polarity features only, Tf-Idf features only, and combined polarity and Tf-Idf features. Across each dataset, the use of only polarity features consistently produced the lowest classification accuracy. The use of Tf-Idf features produced strong classification accuracies across the WELFAKE and ISOT datasets. Combining polarity features with Tf-Idf features produced test accuracies that were nearly identical to a solely Tf-Idf-based model. With nearly 99% test accuracy against the ISOT dataset, it may be safe to say that classification of this dataset can be solved using Tf-Idf vectors. Our models also performed well against the WELFAKE dataset, obtaining test accuracies of 92% when using Tf-Idf features. The LIAR dataset proved to be the most difficult to classify, with our combined polarity and Tf-Idf feature model obtaining only 63% testing accuracy. The LIAR dataset is different from WELFAKE and ISOT, in the sense that it is comprised of statements made by individuals, as opposed to news articles published by organizations. The difficulty in classifying such a dataset may have to do with the stronger presence of polarizing sentiments in news articles, or the lack of a larger and more diverse corpus. The comparison of true positive and false positive rates of each model can be seen in the receiver operating characteristic (ROC) charts in Figure 2..

The ROC curves indicate that the TextBlob and VADER polarity features extracted against the WELFAKE and ISOT database did provide predictive power. That being said, Tf-Idf features produced far stronger predictive power that completely overshadows the polarity approach. For the LIAR dataset, we see that using polarity features in an attempt to classify the LIAR dataset produced a classifier that was no better than random guessing. The inclusion of Tf-Idf produced a classifier that has marginal predictive power against the dataset.

WELFAKE



LIAR



ISOT

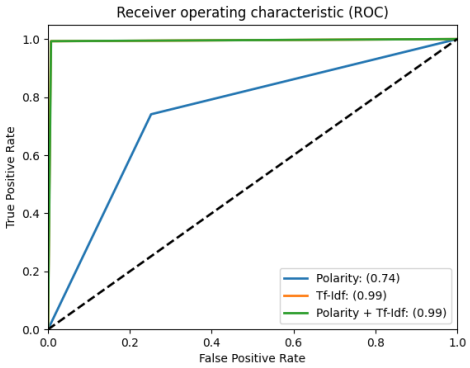


Figure 2: Receiver Operator Characterists (ROC) of conventional NLP fake new classifiers

4 LLM-Based Techniques in the Literature

Large Language Models (LLMs) have changed how natural language processing (NLP) tasks can be approached, including tasks related to fake news detection. LLMs are good at capturing deep semantic relationships within text by using pre-trained embeddings. These embeddings represent the contextual meaning of words, phrases, and sentences numerically, enabling advanced classification models. [15] Since LLMs excel at understanding language patterns, this semantic depth makes them particularly suited for detecting subtle cues in fake news articles. [16]

LLM-based approaches can be described as 'model-level' approaches, where the focus is on the model's capability to extract meaningful features and classify news articles based on learned linguistic patterns, rather than just analyzing individual words or text features.

Commonly used pre-trained LLMs for fake news detection include BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer). Similar to traditional NLP techniques, preprocessing the data is the first stage, which usually includes steps involving tokenization, stopword removal, and lemmatization Tokenization divides the text into smaller units like words or sub-words, while lemmatization tries to reduce words to their root forms. These steps help in reducing noise and improving the efficiency of the model during training.[17] Once pre-processing is complete, LLMs are typically trained on the dataset for fake news detection. During fine-tuning, the model learns patterns specific to the dataset, such as the linguistic styles of fake versus real news. The goal is to use the model's pre-trained weights and fine-tune them on the specific fake news dataset. After this the model is evaluated, and results are derived. [17] LLMs models have been integrated with other data types such as images [18] and other machine learning techniques to make hybrid models that further improve fake news detection techniques. [19] LLMs have proven to be a powerful tool for detecting fake news.

5 Our Approach to LLM-Based Fake News Detection

We leverage pre-trained LLMs to extract embeddings from text data for fake news classification. We used multiple different models of LLM for this task:

BERT: A transformer-based model that encodes contextual embeddings bidirectionally. BERT understands the meaning of a word by analyzing its surrounding words in both directions. For fake news detection, the article text is passed through BERT to generate embeddings that encapsulate word relationships and sentence structures. [17]

SentenceTransformers (a.k.a. SBERT): Another transformer-based model that is very similar to BERT and is used to compute embeddings, but is focused on sentence-level word embeddings rather the word level, allow it to better capture the sentiment of whole sentences.[20]

GPT: While GPT is a decoder that is primarily used for text generation, it can also generate embeddings for fake news detection. Its auto-regressive nature captures patterns in a forward direction, focusing on sequential dependencies in the text. [21]

5.1 Dataset Preparation, Tokenization and Embedding Generation

The text data for our project was sourced from three labeled datasets: LIAR, ISOT, and WELFake. Each dataset was processed to ensure consistency in labeling, with a 'label' column where '1' indicates real news and '0' indicates fake news.

For the ISOT and WELFake datasets, article titles and contents were concatenated into a single full-text record to be processed. The LIAR dataset, consisting of short statements excluded titles as a column and so the single text column was used for processing.

For GPT-based processing, we used the 'GPT2Tokenizer' and 'GPT2Model', while the 'BertTokenizer' and 'BertModel' were initially used for BERT-based processing after which the SentenceTransformer 'all-MiniLM-L6-v2' was used for the model. All of the tokenizers and models are tools from the Hugging Face library.

5.2 Application and Performance of our LLM-Based Models

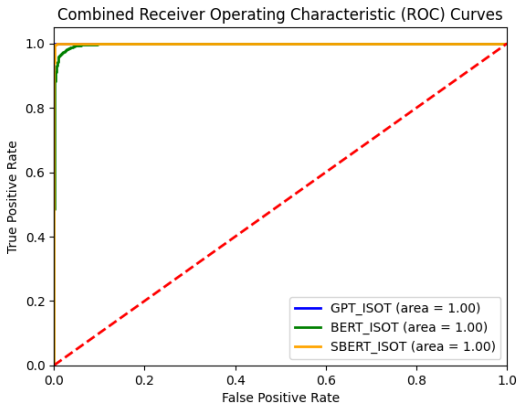
For each of the three datasets, the dataset is split into train, validation and test sets and then for each of these sets, the BERT and GPT tokenizers are used to tokenize the text data input and from this input, the BERT and GPT models are used to generate the embeddings. The SentenceTransformer model was later used for each of these sets as well and was able skip the tokenization step and directly encode the text input to produce embeddings. We used these test set embeddings to train an ensemble model consisting of a combination of Logistic Regression, Random Forest and SVC (Support Vector Classification) classifiers. After this, the ensemble model was validated on the validation set embeddings and tested on the test set embeddings. The ISOT dataset had nearly perfect classification accuracy for both models. Similarly, the WELFAKE dataset also showed a high performance, especially with the GPT based model.

As observed in our conventional NLP analysis, the LIAR dataset again proved to be the most difficult to classify. It's likely that this is due to the LIAR dataset consisting of relatively short statements instead of whole or mostly whole articles like with the WELFake (WFAKE) and ISOT datasets. The classification accuracies obtained by our LLM-based models for each dataset are shown in Figure 10.

Dataset	Metric	SBERT (%)	BERT (%)	GPT (%)
WFAKE	Validation	93.22	87.95	97.03
	Test	92.40	89.32	97.39
ISOT	Validation	99.71	99.45	99.99
	Test	99.65	99.35	99.98
LIAR	Validation	59.88	63.25	61.07
	Test	61.33	61.12	61.73

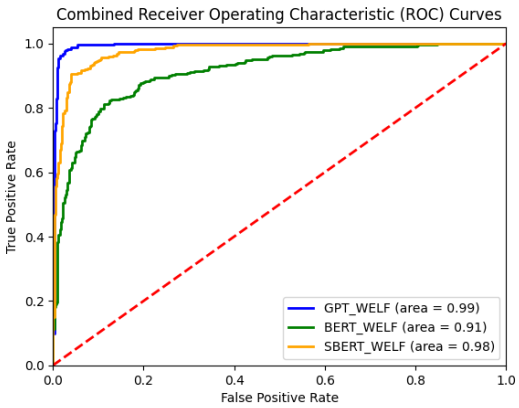
Table 3: Classification accuracies of our LLM-based models across each fake news dataset.

ISOT

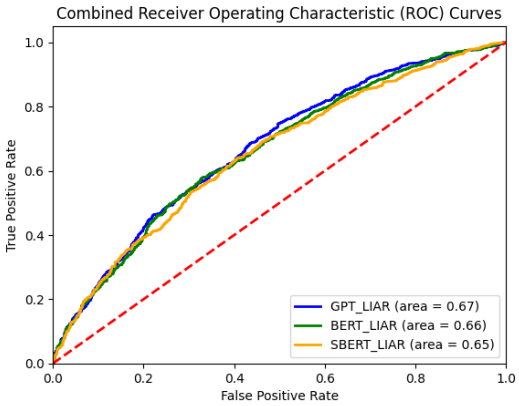


Figures 11, 12, and 13 show the receiver operating characteristic (ROC) curves for each model. These figures illustrate that the GPT based models have the best predicting performance, especially for the WELFake dataset. For the ISOT dataset, all the models have perfect or near perfect curves. For the LIAR dataset, all the models are similarly low, these ROC curves suggest only small improvements over random guessing.

WELFAKE



LIAR



6 Conclusions

In an attempt to enhance the approaches to conventional NLP fake news detection that is found in the literature, we explored the use of polarity-based feature extraction. Our findings indicate that polarity features extracted from fake news datasets can provide predictive power for fake news classifiers. When compared to the more common Tf-Idf approach for feature extraction, however, polarity-based features pale in comparison. The classification accuracies obtained against the ISOT, LIAR, and WELFAKE datasets are consistent with those found in the literature that also took a Tf-Idf-based approach to NLP fake news detection [6, 7].

In our exploration of LLM-based approaches to fake news detection, we demonstrated that embeddings generated by pre-trained language models such as BERT and GPT-2 can provide high classification performance for fake news classifiers. The results show that LLMs, particularly GPT-based models, are very effective at capturing the language patterns present in news articles. The classification accuracies obtained against the ISOT and WELFAKE datasets are mostly consistent with those found in the literature using GPT in their models [22, 23], and somewhat lower than those using BERT with WELFake [24, 25]. The next step is the comparison with the conventional NLP approach.

Overall, the findings suggest that both LLM and conventional NLP-based approaches to fake news detection can provide strong classification models. Our combined polarity and Tf-Idf NLP models appear to almost identical to the SBERT models we produced. The BERT models were less effective, with our NLP models outclassing BERT on the WELFAKE dataset by about 3%. The strongest models across all datasets were the GPT model, with our GPT model outperforming our best NLP model by 5% on the WELFAKE dataset. The results of our GPT models indicate that a decoder-only architecture is significantly more effective at detecting fake news articles than the encoder-only architecture provided by BERT. Recommendations to further compare LLM and NLP fake news detection methods could include an alternative method of detecting polarizing sentiments in fake news articles, and a stronger focus on GPT or alternative decoder-only models.

References

[1] Ahmed H, Traore I, Saad S. "Detecting opinion spams and fake news using text classification", Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

[2] Ahmed H, Traore I, Saad S. (2017) "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).

[3] W. Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection," in arXiv preprint arXiv:1705.00648, 2017.

[4] Pawan Kumar Verma, Prateek Agrawaland Radu Prodan, "WELFake dataset for fake news detection in text data", IEEE Transactions on Computational Social Systems, no. doi: 10.1109/TCSS.2021.3068519. Zenodo, pp. 1–13, Feb. 25, 2021. doi: 10.5281/zenodo.4561253.

[5] I. Ali, et al., "Fake news detection techniques on social media: A survey," Wireless Communications and Mobile Computing, vol. 2022, pp. 1–17, Aug. 2022. doi:10.1155/2022/6072084

[6] A. H. Almarashy, M.-R. Feizi-Derakhshi, and P. Salehpour, "Enhancing fake news detection by multi-feature classification," IEEE Access, vol. 11, pp. 139601–139613, 2023. doi:10.1109/access.2023.3339621

[7] S. V. Balshetwar, A. RS, and D. J. R, "Fake news detection in social media based on sentiment analysis using classifier techniques," Multimedia Tools and Applications, vol. 82, no. 23, pp. 35781–35811, Mar. 2023. doi:10.1007/s11042-023-14883-3

[8] B. Bhutani, N. Rastogi, P. Sehgal, and A. Purwar, "Fake news detection using sentiment analysis," 2019 Twelfth International Conference on Contemporary Computing (IC3), Aug. 2019. doi:10.1109/ic3.2019.8844880

[9] S. Loria, "TextBlob," TextBlob, A Python Package. 2014

[10] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014

[11] S. Bird, E. Loper, "NLTK: The Natural Language Toolkit," in Proceedings of the ACL Interactive Poster and Demonstration Sessions, 2004, pp. 214–217.

[12] Maas, A., et al, "Learning Word Vectors for Sentiment Analysis," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 142–150.

[13] https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset/data

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

[15] arXiv:2407.00737v2 [cs.CV] 27 Aug 2024 https://arxiv.org/html/2407.00737v1

[16] Optimization Techniques for Sentiment Analysis Based on LLM (GPT-3) https://arxiv.org/pdf/2405.09770

[17] Dhiman P, Kaur A, Gupta D, Juneja S, Nauman A, Muhammad G. GBERT: A hybrid deep learning model based on GPT-BERT for fake news detection. Heliyon. 2024 Aug

[18] Essa, E., Omar, K. & Alqahtani, A. Fake news detection based on a hybrid BERT and LightGBM models. Complex Intell. Syst. 9, 6581–6592 (2023). https://doi.org/10.1007s40747-023-01098-0

[19] Segura-Bedmar, Isabel, and Santiago Alonso-Bartolome. "Multimodal fake news detection." Information 13.6 (2022): 284. https://www.mdpi.com/2078-2489/13/6/284

[20] https://marketbrew.ai/a/sentence-bert

[21] Kenton, J. D. M. W. C., & Toutanova, L. K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of naacL-HLT (Vol. 1, p. 2). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[22] K. Hemina, F. Boumahdi, A. Madani, M.A. Remmide A cross-validated fine-tuned GPT-3 as a novel approach to fake news detection Proceedings of the International Conference on Applied CyberSecurity, Springer (2023), pp. 41-48 https://link.springer.com chapter/10.1007/978-3-031-40598-3_5

[23] Kuntur, S.; Krzywda, M.; Wróblewska, A.; Paprzycki, M.; Ganzha, M. Comparative Analysis of Graph Neural Networks and Transformers for Robust Fake News Detection: A Verification and Reimplementation Study. Electronics 2024, 13, 4784. https://doi.org/10.3390/electronics13234784

[24] Verma, P.K., Agrawal, P., Madaan, V. et al. MCred: multi-modal message credibility for fake news detection using BERT and CNN. J Ambient Intell Human Comput 14, 10617–10629 (2023). https://doi.org/10.1007/s12652-022-04338-2

[25] R. Udayakumar, N. Yamsani, S. L. Sajja, Y. Ashok Kumar and L. K. R, "Automatic Fake News Detection on Social Networks using Multimodal Approach of BERT and ResNet110," 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT), Bengaluru, India, 2023, pp. 1-5, doi: 10.1109/EASCT59475.2023.10393050.