Team B: Survey and Innovation on Fake News Detection with NLP

Yang Lu ylu739@gatech.edu
Mingwei Zhu mzhu355@gatech.edu
Haijiao Tao htao49@gatech.edu
Junyuan Quan jquan33@gatech.edu

1 Introduction

The existence of fake news significantly impacts the quality of our social and virtual lives. Misuses of fake news could be malicious attacks against political opponents, dissemination of unnecessary social anxiety, and threats against our physical security and mental health. There has never been a more urgent time to detect and remove fake news from the online space for a clean and healthy virtual community. NLP researchers have been standing at the forefront of this mission, and our team plans to stand on the shoulders of the great and carry on their torch.

Our project aims to synthesize existing NLP attempts at fake news detection and craft an original approach that we believe most applies to evaluating reports on public issues of social and political significance. Previous studies have used classification methods such as support vector machine, naive Bayes classifier, and logistic regression to identify suspicious contents (Conroy et al., 2015; Khurana and Intelligentie, 2017; Shu et al., 2020) . Some researchers have also attempted linear regression to estimate the numeric score of truthfulness (Nakashole and Mitchell, 2014). Other studies have focused on the utility of neural networks and deep learning to capture more nuances in natural language (Thota et al., 2018). Our project will continue evaluating these methods as well as LLM models based on news and social media corpus and propose an approach with the best performance.

2 Literature Review

2.1 Machine Learning and Deep Learning Techniques

Traditional NLP fake news detection tasks leverage the utility of machine learning and deep learning. It's essential to review their benefits and limitations pertinent to our project.

Thota et al. explore the utility of deep learning in detecting fake news by analyzing the relationship between news headlines and bodies (Thota et al., 2018). The authors used the Fake News Challenge (FNC-1) dataset, containing 1,684 articles and 49,973 headline-article pairs labeled as agree, disagree, discuss, or unrelated. The preprocessing involved stop word removal, punctuation removal, and stemming. Word vectors were represented using TF-IDF and Word2Vec embeddings. The paper tested three models: TF-IDF with Dense Neural Network, Bag-of-Words with DNN, and Word2Vec with DNN. The TF-IDF with DNN model achieved the highest accuracy at 94.31%, outperforming BoW (89.23%) and Word2Vec (75.67%). However, the model has its limitations on performing poorly on the disagree stance, with only 44.38% accuracy. Our project aims to improve fake news detection by using social media datasets and can try the TF-IDF with other DL methods to explore about study.

Oshikawa et al. further introduce ways to compare and contrast fake news classification models (Oshikawa et al., 2018). They focus on the LIAR dataset consisting of 12,836 real-world short statements and six-grade truthfulness labels, the Fever dataset with 185,445 claims generated from Wikipedia, and FakeNewsNet containing fake news headlines and bodies from BuzzFeed and PolitiFact. Methodwise, they leverage TF-IDF and DrQA to identify central claims and input them along with raw textual data into testing machine learning and deep learning models. They set accuracy as the benchmark for comparison and find that binary classification CNN outperforms its multi-level counterpart. One key takeaway is to vary sources of text inputs to avoid publisher bias. Per the authors' suggestion, we will incorporate the order of truthfulness score into CNN models to improve their poor model performances.

2.2 Overall problem-solving approaches

The multifaceted nature of fake news detection requires detailed problem definition and task breakdown. Existing literature has provided useful insights on identifying areas of focus and overall problem-solving approaches.

Shu et al. introduce FakeNewsNet, a repository for fake news detection research on social media (Shu et al., 2020). The datasets include news content, social context, and spatiotemporal information. The methodology focuses on three areas: News Content Analysis (using topic models to compare linguistic differences), Social Context Analysis (examining user behavior, sentiment, and bot detection), and Spatiotemporal Information (analyzing how fake news spreads). Machine learning models like SVM and CNN were compared with the SAF (Social Article Fusion) model, which combines content and context. SAF performed best, achieving 69% accuracy on PolitiFact and 68.9% on GossipCop. This article is useful because it breaks down the question into linguistics, language processing, and social network analysis. It identifies methods that work particularly well for our NLP task while informing us the contextual and linguistic aspects to be considered.

On the other hand, two major categories was provided to solve the fake news problem: linguistic cue approaches and network analysis approaches (Conroy et al., 2015). Both approaches typically incorporate machine learning techniques for training classifiers to suit the analysis. Linguistic cue approaches methods rely on analyzing the language used in deceptive messages. Techniques include the "bag of words" approach, deep syntax analysis, and semantic analysis. These methods primarily observe the frequency and patterns of word usage, sentence structure, and coherence to identify deception. Machine learning classifiers such as SVM and Naive Bayes classifiers are used to train models. Among them, comparison between human judgment and SVM classifiers showed 86% accuracy in detecting negative deceptive opinion spam. Network approaches methods use metadata and network behavior to assess the truthfulness of information. Network-based techniques analyze the relationships between entities and facts to determine the likelihood of truth. The linguistic and network approaches methods have demonstrated high accuracy in classification tasks within limited domains, and techniques from different approaches may be combined into a hybrid system. The methods described by the authors have limitations in handling complex language content and are unable to assess the content itself. Our project, by utilizing large language models, aims to overcome these limitations through deeper semantic understanding.

Furthermore, previous research have investigated the dissemination and impact of fake news on Twitter during the 2016 U.S. presidential election (Grinberg et al., 2019). According to the authors' analysis, the spread of fake news was highly concentrated, with a small portion of users, only 1%, encountering 80% of the fake news, while 0.1% of individuals shared the majority of it. These super sharers and super consumers were predominantly older, conservative, and politically engaged individuals. Despite the prevalence of fake news, the majority of political news exposure for individuals across political affiliations still came from mainstream media. The authors measured the exposure to and sharing of fake news, identifying a small group of "super sharers" and "super consumers." They also analyzed the co-exposure network, showing the relationships between fake news sources and mainstream media to determine whether the news is fake. Although this article is not directly related to language processing, it highlights the importance of varying dataset sources to avoid bias and provide quality training data for modeling. We also attempt to leverage BERT and GPT to accurately identify and check fake news dissemination without relying on network behavior data.

Zhixuan, Zhou, et al. (2019) critically evaluate the performance of NLP-based fake news detectors and highlight significant vulnerabilities. Through testing with Fakebox, the authors demonstrate that such systems are susceptible to fact-tampering attacks, resulting in a high false positive rate for real news that's stylistically underwritten or controversial. Specifically, they reveal that Fakebox achieves a classification accuracy of only 52.77%, with a false positive rate of 56.03% for real news, which underlies the urgency for integrating more robust fact-checking mechanisms. The study suggests a crowdsourced knowledge graph as a potential enhancement to improve detection accuracy by validating the factual content against a dynamically updated database.This paper primarily critiques Fakebox, but the findings might not generalize across all NLP-based fake news detectors that may employ different architectures or datasets. Therefore, although similar problems might be potential in our

project, we can still proceed by developing or integrating advanced NLP models that not only analyze linguistic features but also understand contextual nuances and factual consistency, possibly through deep learning techniques that can better understand narrative structures and logical flows.

2.3 Innovation & Data Quality

Other studies also attempted to innovate traditional methods for model accuracy and further emphasized the importance of data quality.

FactChecker, a language-based truth-finding algorithm that significantly improves on traditional methods was presented in earlier study (Nakashole and Mitchell, 2014). FactChecker uses linguistic features to assess the objectivity of sources and the credibility of facts mentioned in similar contexts. According to the experiments, FactChecker achieves a high accuracy ranging from 70% to 88% across different datasets, outperforming other approaches by at least 10% in certain categories such as book authors, movie directors, and athlete teams. It employs an objectivity score that combines linguistic objectivity and co-mention analysis, which enhances the accuracy of truth assessments in information-spread scenarios. This approach marks a substantial advance in handling misinformation by leveraging language nuances to verify factual statements effectively. However, we still need to be cautious about this because the reliance on linguistic features may limit the model's effectiveness across different languages or in texts where nuanced expressions of bias are present, while our dataset indeed contains news in different languages.

Additionally, Sadeghi et al. (2022) apply natural language inference and test the veracity of articles based on a set of reliable news. This "hypothesis-testing" approach leverages Bidirectional LSTM (BERT) and Bidirectional GRU neural network to infer the correctiveness of a given claim based on previously confirmed related news. FakeNewsNet dataset is used for 2-label classification while the LIAR dataset is used for the 6-label problem with categories "pants-fire", "false", "barely-true", "half-true", "mostly-true", and "true". The result is an overall performance improvement for both problems compared to existing models. This study is a great example of integrating existing NLP approaches and introducing inferences into fake news detection. A key takeaway is on the selection of the dataset — we need to ensure the integrity of the baseline texts so that models are learning from reliable sources and yield useful performance metrics.

3 Proposed Methodology

This study evaluates the efficacy of zero-shot prompting in detecting fake news using BERT and GPT models. The primary objective is to assess the ability of large language models (LLMs) to classify news articles as either "real" or "fake" without additional training or fine-tuning. This approach leverages the pre-trained knowledge and inference capabilities of the models to analyze the textual content of articles directly.

In the proposed method, each article is processed by providing its textual content as input to the models, accompanied by a structured prompt designed to elicit accurate classifications. This zero-shot approach minimizes preprocessing and feature engineering, focusing on the models' ability to utilize their existing linguistic and contextual understanding. The simplicity of this methodology ensures that any observed differences in performance are primarily attributable to the inherent capabilities of the models rather than external modifications.

To ensure a robust evaluation, the performance of BERT and GPT models will be compared under identical conditions. Their outputs will be analyzed for accuracy in identifying misinformation and for their ability to capture contextual nuances and factual consistency. Additionally, the results of these LLM-based approaches will be benchmarked against traditional machine learning models, including Support Vector Machines (SVM) and Naive Bayes classifiers, trained on the same dataset. This comparison will highlight the strengths and limitations of zero-shot LLMs relative to conventional models.

Finally, the models' performance will be evaluated using standard metrics, including accuracy, precision, recall, and F1-score. The analysis will examine the results across multiple subsets of the dataset to identify variations in effectiveness based on context length and content type. By directly comparing LLMs to traditional models, this study seeks to provide a comprehensive understanding of their capabilities in fake news detection and explore their potential as tools for combating misinformation.

4 Dataset

We used the Kaggle LIAR fake news dataset, which includes over 10,000 labeled articles classified as

real or fake. This data set will provide a solid foundation for training, testing, and validating our models.

The data we collected contains about 10,240 entries for training, 1.2K for validation, and 1.2K for testing. Although the data was well-structured, certain columns were irrelevant and removed to streamline the analysis.

There are some columns that have missing values, for example speaker_job_title, state_info, and context. We dropped these and other unnecessary columns such as ID, barely_true_counts, false_counts, half_true_counts, mostly_true_counts, and pants_on_fire_counts. We also performed feature engineering to turn Mostly-True and Half-True to True and Barely-False and Pants-Fire to False. The purpose is to pool the samples to reduce bias in the original categories. After removing duplicates and rows with missing values, the data set was cleaned and ready for analysis.

5 Experimental Results

We experimented with SVM and Naive Bayes classifiers, just to gauge their performances on the dataset. For SVM we experimented with two types of classifiers: LinearSVC and SVC with a Sigmoid Kernel. Both classifiers were evaluated using a TF-IDF representation of the text data, followed by dimensionality reduction using truncated singular value deposition(SVD). We began by transforming the text data into a TF-IDF matrix using a bi-gram representation (ngram_range=(1, 2)) and applied TruncatedSVD to reduce the dimensionality of the TF-IDF matrix to 500 components. This was done to improve the efficiency of the models during training and testing.The models were trained on 80 percent of the original dataset and tested on the remaining 20 percent.

We follow the same method as SVM for text preprocessing, feature weighting, and dimensionality reduction. One of the unexpected challenges was that the MultinomialNB classifier does not accept negative values in the input feature matrix, which arose when using TruncatedSVD for dimensionality reduction. The initial solution applied was the MinMaxScaler to ensure all values were non-negative, which allowed the model to run without errors. However, this adjustment led to suboptimal performance in terms of accuracy.

The Transformer model's core architecture is enhanced with residual connections and layer normalization to stabilize training and speed convergence. Training uses cross-entropy loss with the Adam optimizer, and initial hyperparameter tuning was conducted for learning rate and dropout probability. A major issue encountered was overfitting. To mitigate this, we included dropout layers and plan to further refine hyperparameters and explore additional regularization techniques to improve the model's generalization on unseen data.

The classification accuracy for all models are summarized in the table below.

Model	Accuracy
Logistic	64%
SVM	62%
Naive Bayes	56%
BERT	56%
Decoder-Only Transformer	65%

Figure 1: Results

6 Comparison with other baseline

To evaluate the effectiveness of our LLMs such as BERT and Transformer, we compared its performance with traditional baseline models, including Logistic Regression, support vector machines, and Naive Bayes. The models were trained and tested on the same dataset to ensure a consistent and fair comparison.

Interestingly, the results indicate that LLMs do not consistently outperform simpler models. Especially for Logistic regression and SVM, the transformer does not give a significant boost. While the Transformer achieved slightly higher accuracy overall, the difference was not substantial enough to signify a clear advantage.

We presume that these results may be partially due to data quality issues. The data consists of very short paragraphs with limited contextual information, leaving little opportunity for the Transformer model to leverage its contextual embedding capabilities effectively. It is also possible that some lexical cues have already been picked up during feature engineering and dataset creation, during which a lot of lexical features are summarized and labeled. As a result, the simpler models are good enough to utilize these features and yield performances comparable to more complex LLMs.

7 Ablation Study

In line with the findings in Section 6, the proposed Transformer design did not demonstrate a decisive advantage over simpler models. However, we do see the effect of pooling and feature engineering in improving classification accuracy. After integrating the labels, all models showed significant improvements in classification accuracy, as illustrated in Figure 2.

Model	Accuracy
Logistic	27% → 64%
SVM	26% → 62%
Naive Bayes	27% → 56%
BERT	22% → 56%
Decoder-Only Transformer	21% → 65%

Figure 2: Results Comparison before and after feature engineering

By cleaning and simplifying the dataset, the models were able to concentrate on meaningful features without being hindered by irrelevant or redundant data. Integrating labels as part of the feature engineering effort effectively reduced category imbalances, which proved essential for learning reliable patterns. Notably, the Decoder-Only Transformer achieved an accuracy of 65%.

8 Conclusion

Our findings suggest that the tested models, including both traditional baselines and LLMs, achieved moderate accuracy levels overall. Contrary to initial expectations, the Transformer model did not provide a significant advantage over simpler approaches such as Logistic Regression or SVM.

Several factors likely contributed to these results. The dataset's quality and structure played a pivotal role—short text passages with limited context constrained the ability of advanced models to capture additional insights. Furthermore, much of the critical lexical information was already captured during feature engineering and dataset labeling, reducing the need for complex contextual representations.

We believe that the reasons could be data quality issues, limited contexts in short paragraphs, and lexical cues being picked up before modeling.

These findings underline the importance of tailoring model complexity to the dataset's characteristics and the challenges of leveraging LLMs when context is inherently limited.

9 Limitations/Next Steps

Our next steps focus on refining the Transformer-based model by tuning multi-head attention parameters and embedding dimensions to better capture text nuances. To address data imbalance, we'll implement techniques like class weighting, oversampling, or data augmentation, improving model performance on underrepresented categories.

After model refinement, we will thoroughly evaluate it using metrics such as accuracy, F1-score, precision, and recall. This comprehensive assessment will ensure balanced performance across classes, with F1-score helping address any residual class imbalance. Through iterative tuning and evaluation, including testing different Transformer variants (e.g., BERT, GPT), we aim to develop a robust, adaptable fake news detection model that can generalize well to various misinformation types.

References

Nadia K Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the association for information science and technology*, 52(1):1–4.

Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. 2019. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378.

Urja Khurana and Bachelor Opleiding Kunstmatige Intelligentie. 2017. The linguistic features of fake news headlines and statements. *Diss. Master's thesis, University of Amsterdam*.

Ndapandula Nakashole and Tom Mitchell. 2014. Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1019.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.

Fariba Sadeghi, Amir Jalaly Bidgoly, and Hossein Amirkhani. 2022. Fake news detection on social media using a natural language inference approach. *Multimedia Tools and Applications*, 81(23):33801–33821.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188.

Aswini Thota, Priyanka Tilak, Simrat Ahluwalia, and Nibrat Lohia. 2018. Fake news detection: a deep learning approach. *SMU Data Science Review*, 1(3):10.

Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat, and Justin Hsu. 2019. Fake news detection via nlp is vulnerable to adversarial attacks. *arXiv preprint arXiv:1901.09657*.