Enhancing Fake News Detection Through Topic Modeling Techniques

Ping-Chun Lin, Zimo Wen, Andy Zhang, Deyang Zheng
University of Washington
pcl1225, zimow3, fz29, deyangz @uw.edu

Abstract

This study addresses the detection of fake news on social media platforms by analyzing a dataset comprising over 40,000 news articles from 2015 to 2018, including both fake and real news. Non-negative Matrix Factorization (NMF) and Latent Dirichlet Allocation (LDA) are employed for topic modeling, whose features are integrated into a BERT-based model to enhance its training. We compare the performance of this enhanced model against a baseline model without topic features, particularly focusing on the impact of these features under varying training set sizes. Our results demonstrate significant performance improvements in smaller training sets, showing that topic modeling not only enhances model accuracy but also reduces the demand for GPU resources and processing time, facilitating efficient and robust fake news detection even in resource-constrained environments.

Keywords: Fake News Detection, Topic Modeling, Non-negative Matrix Factorization, Latent Dirichlet Allocation, BERT

1 Introduction

1.1 Motivation

In the digital age, fake news has emerged as a substantial threat to society, influencing public opinion and democratic processes, especially through social media platforms (Shi et al., 2018). Fake news comes in many different forms. Besides traditional news articles, short-text messages, images, and videos are common approaches to spreading misinformation. Misinformation also exists in a wide range of topics, such as politics, the stock market, and public health, which might be challenging to classify if the machine learning algorithm is trained only on one topic. The topics of misinformation also change over time, known as the concept drift issue (Raza and Ding, 2022; Hoens et al., 2012). Because of the complexity of misinformation, the performance of a well-trained model might vary a lot (Castelo et al., 2019). Hence, we need to understand the dynamic nature of misinformation so that automated detection systems can accurately and rapidly react to different types of fake news.

1.2 Problem Definition

Most detection algorithms perform well on specific fake news topics, such as COVID-19-related misinformation (Wani et al., 2021; Gundapu and Mamidi, 2021). However, performing well on all topics or even fake news that has yet to exist is challenging. Hence, we propose a two-stage analysis that consists of two tasks that aim to extend the applicability of current fake news detection models and improve the adaptability and effectiveness of fake news detection systems. **Task 1 involves performing topic modeling on the fake news dataset to cluster and extract features from various**

Preprint. Under review.

topics. Task 2 utilizes the topic modeling results from Task 1 as input for state-of-the-art fake news detection algorithms.

Another aspect of the project is to see if dimensionality reduction and unsupervised machine learning algorithms, such as clustering, can help reduce the processing time for detection systems while still retaining high accuracy. In recent years, pre-trained large language models (LLMs) have demonstrated their ability to perform well in classification tasks. However, this often requires a large amount of data, computational resources, and time for the models to perform well. Hence, we are also exploring the opportunity to test if we can use topic modeling to reduce the computational cost for LLMs, which would potentially become easier to deploy while maintaining reasonable performance. Through a comparative analysis of topic modeling methods and "fake news trends," we anticipate developing a better understanding of fake news and how to improve the practicality of the detection algorithms.

2 Related Work

The two tasks in our project have different goals, one being classification and one being detection. Hence, we reviewed multiple studies including Choudhary et al. (2020), Altheneyan and Alhadlaq (2023), and Raza and Ding (2022).

Choudhary et al. (2020) first extract sentiment and readability features for the dataset and verify that the base learner and the integrated algorithm are efficient with a maximum accuracy of 72%. This method focuses on readability features and uses a sequential neural network model based on combining features and achieves 86% accuracy on the test set. One issue with the data is that it is large and complex. However, the authors did not mention how to pre-process the data more efficiently. The authors also demonstrated that a combined feature model performs better with the same learning model than the differentiated feature set. This result inspired us to use the TF-IDF matrix and bag-of-words model in this project to incorporate the construction of features based on the frequency of occurrence of keywords.

Altheneyan and Alhadlaq (2023) focused on the challenge of detecting fake news on social media platforms using big data and machine learning techniques. The authors employ a distributed Spark cluster to develop a stacked ensemble model for news classification. Their approach includes feature extraction using N-grams, Hashing TF-IDF, and count vectorizer techniques, followed by applying the stacked ensemble classification model. The proposed model has an F1 score of 92.45%, significantly higher than the baseline models. This model leverages the distributed computing power of Spark to handle large datasets effectively, improving accuracy and efficiency in fake news detection. While the authors improved the performance, they have yet to fully explore if similar procedures can be utilized on neural networks and transformer-based prediction models for fake news detection.

Raza and Ding (2022) proposed an algorithm that introduces a transformer-based model (BART) with a bidirectional encoder and a left-to-right decoder, effectively handling the complexities of language in large datasets for accurate fake news detection. It incorporates social context data, which significantly improves the distinction between fake and real news and enables the detection of misinformation shortly after dissemination. However, the model's computational demands could hinder its application in real-time systems requiring fast responses. The reliance on extensive, high-quality labeled datasets and the optimistic assumption about the uniform benefit of social metrics could limit its effectiveness. There's also a need for a more robust analysis of how the model performs under varied and challenging conditions. Future improvements could explore more efficient transformer variants or apply model distillation techniques to enhance performance without sacrificing speed.

3 Data

3.1 Dataset Description

We used the Kaggle ISOT Fake News Dataset for this project. The ISOT dataset has a total of 44898 articles, with 23481 entries from different fake news outlet resources from 2015 to 2018, primarily 2016-2017. Fake news is labeled by fact-check sites Politifact and Wikipedia. This data set has about the same number of entries for fake and true news and is categorized into multiple topics. We chose to analyze this dataset because it divided the subject of the news into a few categories. True data is classified into two categories: world news and politics, whereas fake data is classified into six topics:

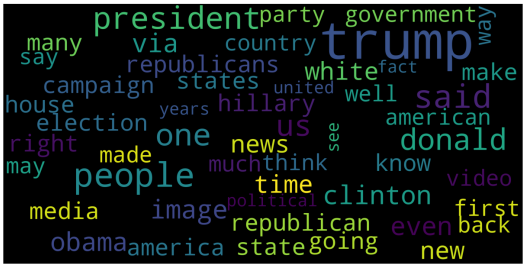


Figure 1: The word cloud from the fake news in the Kaggle ISOT Fake News Dataset.

government, Middle, US News, left news, politics, and (non-categorized) News. While there is some ambiguity in the category labels for each data, the labels can give us context when we are assigning clusters to the data.

3.2 Dataset Preprocessing

Several steps are involved to clean and standardize the text for further analysis. We first broke down the text into individual words or tokens. We converted all text to lowercase to unify variations of the same word to prevent duplication. We then removed the stop words, which are the common but less meaningful words, such as "the," "is," and "at," are removed from the tokens. We also filtered out punctuation and special characters as they generally do not add value to the content. We then performed stemming and lemmatization techniques to reduce words to their base or root forms, minimize vocabulary size, and standardize words. Figure 1 shows the word cloud of Fake news data after preprocessing the data.

Before creating the data loader, we used BERT text for vector embedding. BERT vector embeddings are contextual representations generated by the BERT model, encoding not just word information but its context within a sentence (See Section 4.3 for more detail about BERT embeddings). These embeddings are bidirectional, integrating insights from both preceding and following text. The embeddings improve contextual understanding and thus enhance performance across numerous natural language processing tasks. We also applied padding and truncation techniques to our data set to account for input of different sizes.

4 Methods

4.1 Task 1: Extracting Features from Fake News

Our first task is topic modeling, which divides the fake news dataset into smaller subsetswithe a common theme. Here, we use two methods to explore and extract features from the news dataset. We first used the Non-negative matrix factorization method (NMF) method with Term Frequency-Inverse Document Frequency (TF-IDF) matrix to learn the weights of keywords for the topics and determine the appropriate number of topics. Then we used Latent Dirichlet Allocation (LDA) to do in-depth analysis of topics to get the topic distribution for each new. While our goal is to cluster fake news based on their topic, NMF, and LDA methods are slightly different from using clustering methods introduced in class, such as K-means. Although both topic modeling and traditional clustering algorithms can cluster the data into groups, topic modeling provides more context for interpretation based on the themes. The output of topic modeling is a distribution of topics for each document, whereas clustering assigns data points to specific clusters.

Evaluation of the models are based on internal metrics such as silhouette score and visualizations of clusters. The extracted topics were also used to discover the underlying trend of fake news. Finally, we deployed the results as inputs into the next stage to evaluate the performance of Task 2.

4.1.1 Non-negative matrix factorization method (NMF)

For clustering the topics in fake news data, we started with the NMF algorithm. Given a non-negative n by p matrix V, the non-negative matrix factorization method factorize V into two non-negative

matrices W and H such that:

$$V \approx WH$$

where W is n by k and H is k by p. Usually k is a number much smaller than n or p and it is chosen at our discretion. Here, we have the V as the $TF - IDF$ matrix of our fake news dataset. Similar to the idea of latent factor models, we aim to create two matrices to describe "Fake news factors" and "Word factors" that correspond to distinct topics of discussion. Each row of W is a much lower k-dimensional representation of a piece of fake news through the topic weights.

A popular way for to find the factorization is through the multiplicative weight update method. First two non-negative matrices W and H are initialized, then it performs:

$$H_{ij}^{[n+1]} \leftarrow H_{ij}^{[n]} \frac{((W^{[n]})^T V)_{ij}}{((W^{[n]})^T W^{[n]} H^{[n]})_{ij}} \tag{1}$$

$$W_{ij}^{[n+1]} \leftarrow W_{ij}^{[n]} \frac{(V(H^{[n+1]})^T)_{ij}}{(W^{[n]} H^{[n+1]} (H^{[n+1]})^T)_{ij}} \tag{2}$$

The updates are performed element-wise, and the update terms (multiplicative factors) of W and H are the approximate ones when it converges ($V \approx WH$).

4.1.2 Latent Dirichlet Allocation (LDA)

Another algorithm we used to perform topic modeling is LDA. LDA is a generative statistical model that explains sets of observations, such as words in texts, arising from unobserved groups, known as topics. Each topic is characterized by a distribution over a fixed vocabulary.

The mathematical foundations of the model consist of three aspects: (1) Topics: Each topic Z is a distribution over a vocabulary of words. (2) Documents: Each document D is modeled as a random mixture of latent topics. (3) Words: Each word in a document is attributable to one of the topics assigned to that document.

The generative process for a document in LDA can be described with these steps:

- Choose $\theta_i \sim \text{Dirichlet}(\alpha)$, where θ_i is the topic distribution for document i and α is the parameter of the Dirichlet prior on the per-document topic distributions.
- For each of the words w_{ij} in document i:
  - Choose a topic $z_{ij} \sim \text{Multinomial}(\theta_i)$.
  - Choose a word $w_{ij} \sim \text{Multinomial}(\beta_{z_{ij}})$, where $\beta_{z_{ij}}$ is the word distribution for topic z_{ij}.

For parameter estimation, LDA relies on posterior inference to infer the latent structures (topics and word and topic distributions). The posterior distribution of interest is:

$$p(\theta, z \mid w, \alpha, \beta) = \frac{p(\theta, z, w \mid \alpha, \beta)}{p(w \mid \alpha, \beta)}$$

This distribution is typically approximated using techniques like Variational Bayes or Gibbs Sampling.

4.2 Task 2: Fake News Detection with Topic modeling

4.2.1 Transformers

In Task 2, we aim to evaluate Transformer models' performance in fake news detection and explore whether topic modeling can improve accuracy. By comparing the performance of topic modeling data as embeddings and the baseline model, we tested whether topic modeling could improve the performance of fake new prediction models.

We also investigated the impact of feature-specific training on the algorithms' variance and susceptibility to concept drift when detecting fake news under different topics. This analysis helps us understand how the fake news topic influences detection algorithms' accuracy.

Altheneyan and Alhadlaq (2023) applied clustering to classification machine learning algorithms such as Logistic Regression, Decision Trees, and Random forest models. We want to expand on this concept and explore the possibility of applying topic modeling to transformers, one of the most popular algorithms for fake news detection in recent years.

For Task 2, we use BERT (Bidirectional Encoder Representations from Transformers), a pre-trained Transformer model similar to the idea proposed by Raza and Ding (2022). Figure 2 visualizes the Transformer algorithm based on Vaswani et al. (2017). Developed by Google, BERT's key innovation is applying the bidirectional training of Transformer, a popular attention model, to language modeling. This is achieved through two new strategies: Masked Language Model (MLM) and Next Sentence Prediction (NSP). BERT has been pre-trained on a large corpus of text and then fine-tuned for specific tasks, leading to state-of-the-art performances on a wide range of NLP tasks, including question answering, sentiment analysis, and language inference.

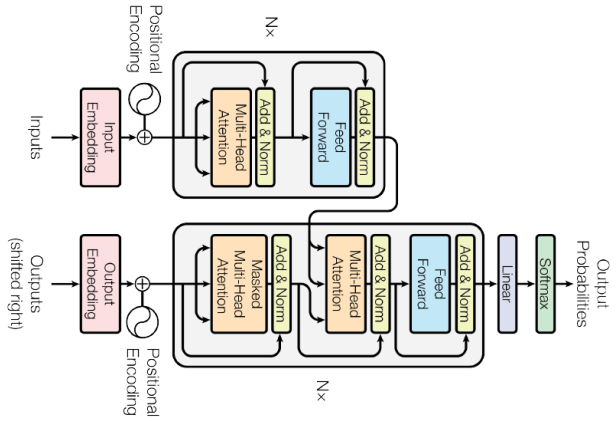


Figure 2: The Transformer model from Vaswani et al. (2017). BERT utilizes the encoder unit, which is the unit on the top (or left when flipped) in the figure (Devlin et al., 2018).

Because the focus of this paper is not a comprehensive review of Transformer models, we only highlight some of the mechanisms of the BERT model in this section. One advantage of BERT, a transformer-based model, is that it uses the self-attention mechanism to weigh the importance of different words in a sentence relative to each other. The formula for self-attention is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

Q (query), K (key), and V (value) are matrices derived from the input embeddings.d_k is the dimension of the key vectors. The input embedding for the BERT model is the sum of three types of embeddings: token embeddings, segment embeddings, and position embeddings (Devlin et al. 2018). This concept leads to multi-head attention that allows the model to focus on different parts of the input sequence simultaneously Vaswani et al. (2017).

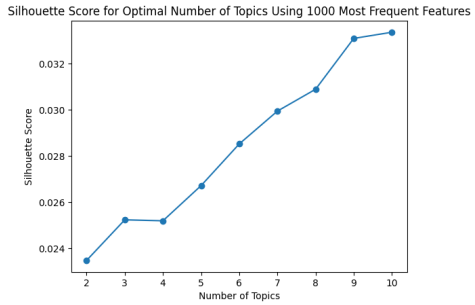
5 Results

5.1 Method 1: NMF

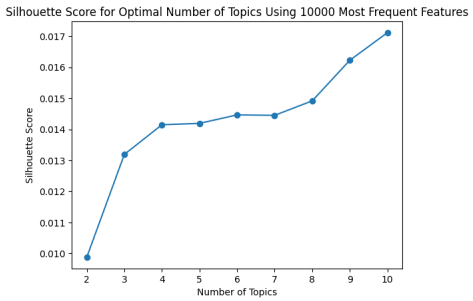
We first convert the pre-processed fake news data into a 23,481 by 183,327 TF-IDF matrix. Here, 23,481 is the number of fake news articles. We set the parameter `max_df` to 0.95, which filters out words appearing in 95% of the fake news articles. This results in 183,327 remaining terms.

We then apply NMF to the TF-IDF matrix, which gives us a 23,481 by k matrix W and a k by 183,327 matrix H. We associate each fake news article with the topic that influences it the most, that is, the argmax of each row of W.

In order to choose the optimal number of topics k, we experiment on restricting the maximum number of features and measure the silhouette score of clusters, with respect to k from 2 to 10.



(a) Silhouette Score for Optimal Number of Topics Using Top 1000 Features.



(b) Silhouette Score for Optimal Number of Topics Using Top 10000 Features.

Figure 3: Silhouette Scores With Respect to k for Different Feature Sets

5.1.1 Silhouette Score

The silhouette score is a measure of how similar an object is to its own cluster compared to other clusters. It is defined for each sample and can be calculated as:

1. Compute the mean intra-cluster distance: the mean distance between a sample and all other points in the same cluster, denoted as a.

2. For each cluster not containing the sample, compute the mean distance between the sample and all the points in the given cluster. The minimum of these mean distances is the mean nearest-cluster distance, denoted as b.

3. The silhouette score for a sample is given by:

$$s = \frac{b - a}{\max(a, b)}$$

The silhouette score ranges from -1 to 1, where a high value indicates that the sample is well-matched to its own cluster and poorly matched to neighboring clusters. If the silhouette score is close to 0, it indicates that the sample is on or very close to the decision boundary between two neighboring clusters.

The plot above shows the mean silhouette across all samples in the dataset with respect to the number of topics. We noticed that the magnitude of silhouette scores is quite small, which implies poor cluster separation due to the dimension of the features being too large. However, the more features are included, the richer vocabulary we have to improve topic summary. We choose $k = 9$ here as there's a noticeable increase in the silhouette score from $k = 8$ to $k = 9$ and the slope starts to decrease from then on. Then, we sort each row of the "word factor" matrix H to find the most relevant words with the highest weights associated with each topic. The result is as follows:

Topic	Keywords	Count
0	people, would, republicans, bill, state, women, said, care, law, health	4431
1	trump, donald, president, campaign, image, said, like, people, white, realdonaldtrump	5919
2	boiler, acr, room, pm, radio, broadcast, animals, tune, episode, alternate	225
3	clinton, hillary, campaign, sanders, foundation, emails, email, democratic, bill, state	2016
4	us, syria, russia, wire, news, century, 21st, russian, media, military	2326
5	police, black, officers, gun, lives, said, white, officer, people, video	4371
6	fbi, comey, investigation, director, russia, james, russian, information, department, emails	1283
7	obama, president, barack, house, white, administration, first, michelle, iran, lady	2038
8	cruz, ted, republican, party, rubio, sanders, gop, candidate, campaign, senator	872

Table 1: Topics, Keywords, and Count of Fake News Using Top 10,000 Features

We notice that using the top 10,000 features results in keywords that summarize each topic well and is computationally efficient. We will use the topic distribution row vectors from the left matrix W as additional features in the latter classification task. To compare the generated topics with those using only 1000 features, for instance, we can see from the keywords for topic 0 below, which doesn't suggest a clear focus and seems to mix up multiple topics.

people	said	women	one	like	would	know	america	us	want

Table 2: Keywords from Topic 0 Using Top 1000 Features

5.2 Method 2: LDA

Compared to NMF, LDA provides a probabilistic framework that is better able to capture the nuances of topic distribution. Dirichlet priors can effectively deal with sparse data, which can benefit when dealing with high-dimensional text data. But at the same time, this results in much longer running times for LDA compared to NMF, which is suitable for tasks that require fast topic classification. The interpretation degree of this model was evaluated by calculating perplexity. When $topic = 9$, the value is -8.575950999615614. The preliminary verification model has good interpretability.

```
(0, '0.038*"trump" + 0.018*"republican" + 0.011*"hillary" + 0.011*"clinton" + 0.010*"vote"')
(1, '0.018*"trump" + 0.014*"president" + 0.012*"said" + 0.009*"russia" + 0.009*"house"')
(2, '0.023*"clinton" + 0.015*"medium" + 0.015*"news" + 0.012*"hillary" + 0.009*"story"')
(3, '0.010*"like" + 0.010*"people" + 0.008*"one" + 0.007*"know" + 0.007*"said"')
(4, '0.010*"million" + 0.009*"year" + 0.008*"state" + 0.007*"000" + 0.007*"money"')
(5, '0.014*"woman" + 0.009*"student" + 0.009*"school" + 0.008*"black" + 0.008*"child"')
(6, '0.060*"trump" + 0.028*"twitter" + 0.019*"com" + 0.016*"pic" + 0.013*"donald"')
(7, '0.021*"u" + 0.009*"state" + 0.007*"country" + 0.007*"syria" + 0.007*"war"')
(8, '0.017*"police" + 0.012*"said" + 0.008*"gun" + 0.008*"officer" + 0.007*"shooting"')
Perplexity: -8.575950999615614
```

Figure 4: LDA topic and perplexity

Therefore, we use NMF to determine the appropriate number of topics(=9 from above), and then use lda for in-depth analysis of topics. We get topic distribution for every new. (see fighre5)

	text	label	Topic_0	Topic_1	Topic_2	Topic_3	Topic_4	Topic_5	Topic_6	Topic_7	Topic_8
0	GAZA/CAIRO (Reuters) - Palestinian factions, i...	1	0.0000	0.5129	0.4837	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	21st Century Wire says B ACTING: President Ob...	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.6851	0.0000	0.3062	0.0000
2	SARAJEVO (Reuters) - In the 1990s he was the b...	1	0.0000	0.0438	0.8851	0.0314	0.0389	0.0000	0.0000	0.0000	0.0000
3	The main stream media has done a great job of ...	0	0.1221	0.2544	0.0612	0.1874	0.3617	0.0000	0.0000	0.0000	0.0113
4	Remember when colleges and university were one...	0	0.0000	0.0000	0.4441	0.0000	0.0000	0.0000	0.1319	0.0000	0.4195
5	LONDON (Reuters) - A record majority of Briton...	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.1897	0.0000	0.8061	0.0000
6	Starbucks SJWs call cops on customer for reque...	0	0.0000	0.0000	0.9643	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	Just two years ago, the Obama White House welc...	0	0.1468	0.0618	0.0000	0.7892	0.0000	0.0000	0.0000	0.0000	0.0000
8	WASHINGTON (Reuters) - U.S. President Donald T...	1	0.8260	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1658
9	Here is his epic response to her criticism: La...	0	0.0000	0.0000	0.4795	0.0000	0.1989	0.0000	0.3064	0.0000	0.0000
10	CNN host Chris Cuomo definitely drew the short...	0	0.0000	0.0000	0.3636	0.0000	0.3229	0.1752	0.0805	0.0567	0.0000

Figure 5: LDA topic distribution

5.3 Task 2: Fake News Detection with Topic modeling

We split our dataset with a test size of 0.2 and employed a pre-trained BERT for tokenization. Given the dataset's size, we utilized the "CUDA" parallel computing tool on Google Colab's GPU. However, limited computational power restricted our ability to train a deep transformer model (BERT's hidden size is 768), resulting in a low test accuracy of approximately 0.6.

In Task 1, we extracted features based on fake news topics; here, we will incorporate the extracted features into the training process of Transformer models. This will allow us to assess whether integrating topic modeling can enhance the robustness and adaptability of fake news detection systems, ensuring their effectiveness even as fake news evolves. There are a few metrics we used to evaluate the performance of fake news detection:

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Recall} = \frac{TP}{TP + FN}$$

Given the substantial size of the dataset, it was initially impossible to train the models using the limited free GPU access provided by Google Colab. Consequently, we opted to pay for the Colab Pro service, which offers additional computing units and access to a more powerful GPU (NVIDIA A100). Even with improvements in computational resources, the whole data set still costs about 1 hour to train.

5.3.1 Algorithm 1: Modified BERT on small dataset

Added the topic distribution of each news as an additional nine parameters to the bert model for training.
The concatenated features are passed through the classifier (linear layer) to obtain the final classification result (logits).

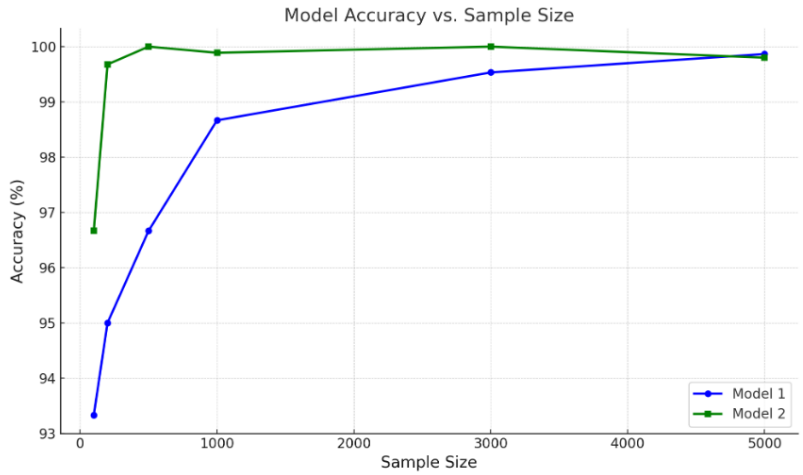


Figure 6: Modified BERT on small dataset

Result Analysis:
1. The results show that there is some improvement when the sample size is small. Still, obviously, the improvement is not very significant, which means the original model already has high robustness even on small data sets.
2. Due to the small sample size, the occasionally of detection results will also increase, so a large number of experiments and statistical analyses, such as observation of p-value, can more effectively prove this conclusion. (Also mentioned in Future work)

5.3.2 Algorithm 2: BERT under the topic

The initial dataset comprised 44,898 news articles grouped into distinct topics using clustering algorithms. By clustering the data and reducing the size of the training sets, we observed a significant reduction in training time across all models. This outcome demonstrates the efficiency gains achievable through targeted training on smaller, topic-specific datasets (Figure 7; Figure 8).

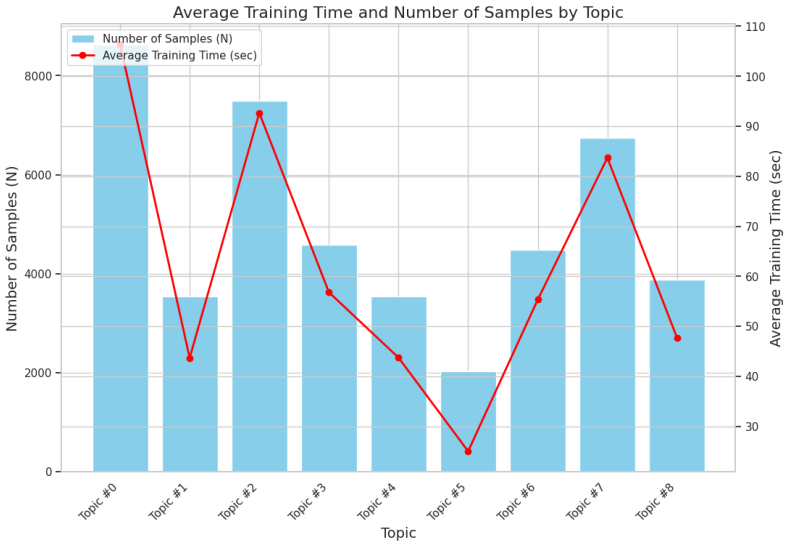


Figure 7: Training time for BERT on topic specified datasets

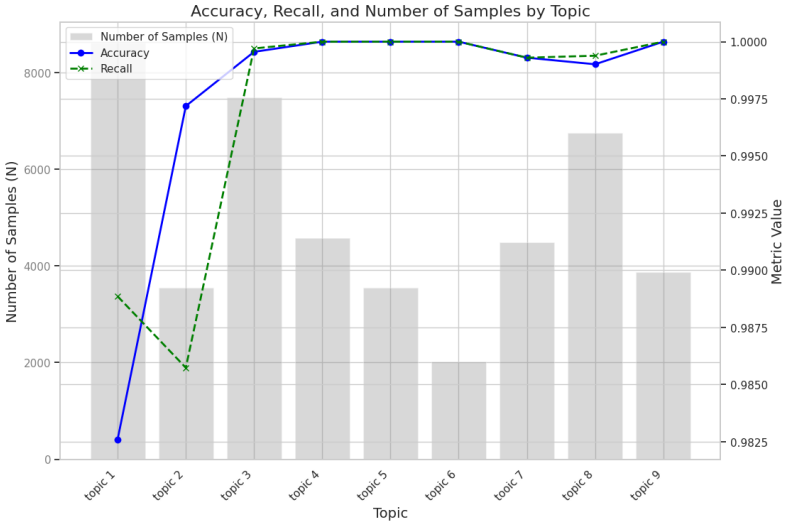


Figure 8: Performance for BERT on topic specified datasets

- **Reduced Training Time:** By clustering the data and reducing the size of the training sets, we observed a significant reduction in training time across all models, with the entire workflow completed in approximately 1 hour.
- **High Model Performance:** The models achieved very good performance metrics, with accuracy rates around 99%. Despite the reduced training data size, there was only a minimal impact on the accuracy and recall of the models.

6 Discussion, Insights and Error Analysis

For Task 1, our optimized number of topics in the fake news data from Task 1 was nine, different from the six categories listed on the Kaggle ISOT dataset. This is because the labels in the dataset might not reflect all of the topics in the data. For example, the largest category in the fake news dataset is the "News" label, which is ambiguous. There also might be an overlap between the categories that only have one label in the dataset. Multiple topics in the results from Tables 1, 2, and 4 have common themes that could be interpreted as multiple categories.

Cluster Evaluation Our priority for Task 1 during this project was to incorporate the extracted features into the detection model in Task 2. So, our evaluation method for Task 1 results was only evaluated based on their performance for detection. For Task 1, an independent comparison between topic modeling methods (NMF, LDA) and clustering methods such as K-means could be evaluated based on topic coherence and topic diversity in clustering to determine if the understanding of the context of fake news could affect the performance of clustering topic that are unknown (Grootendorst, 2022).

Noise and Denoise in Task 2 Due to the very high accuracy of the baseline BERT model on the whole dataset, there is not much space for further improvement. So before we turn to research on reducing computing time, we also tried to add some noise by replacing words with their synonyms and flipping a few labels. Buy we finally find the noise will also decrease the accuracy of the new model which added more features. There was still no significant difference between the experimental and control groups. Even if noise is added, the BERT model still maintains good robustness. That is one of the reasons we changed the research direction.

In our preliminary model, to address issues with input length variance in the current Transformer baseline model, we applied padding and truncation. Attention masks significantly helped the model distinguish between actual input tokens and padding, thereby ignoring the latter during attention calculations.

Challenge: Computational Resources limitations for Task 2 Although we chose the BERT-base as a baseline, which is the smaller version of BERT than BERT-large. It was still impossible to be sucessfully train on Google Colab's free GPU resource. Even with an upgraded A100 GPU, the whole workflow still costs about one hour. That's why we focused on time optimization when performing analysis.

Challenge: High Accuracy from BERT Our results demonstrated that BERT's performance was so high that attempting to enhance it further by incorporating topics as additional features proved ineffective on the whole dataset. Hence, our research focused on improving other aspects, such as reducing training costs and time without compromising the model's effectiveness. We tried two directions: (1) incorporating topics as additional features on the smaller datasets, and (2) instead of the whole dataset, we trained the model on data subsets by topics. Adjusting the training data by size or topic will not negatively affect the performance too much

7 Future Work

- A further investigation of the topic numbers and the context of the topics
- We need to continue to compare the topic models with traditional clustering methods for interpretative evaluation.
- The prediction models for Task 2 took a lot of time. Due to our limited computational resources, we had to work with a small network size. With additional resources provided, we can train with larger network sizes.
- We trained and tested the performance on the Kaggle ISOT Fakenews dataset. Although the models generally achieve high accuracy, we must test on other data sets, such as the FakeNewsNet, with different formats to see if the performances are similar.
- For the transformer structural improvement, RoBERTa is another option that offers improvements in pre-training that could enhance performance on NLP tasks.

References

[1] Choudhary, A., & Arora, A. (2021). Linguistic feature based learning model for fake news detection and classification. https://doi.org/10.1016/j.eswa.2020.114171. Article 114171. [2] Altheneyan, A., & Alhadlaq, A. (2023). Big Data ML-Based Fake News Detection Using Distributed Learning. IEEE Access, 11, 29447–29463. https://doi.org/10.1109/ACCESS.2023.3260763

[4] Raza, S., & Ding, C. (2022). Fake news detection based on news content and social contexts: A transformer-based approach. International Journal of Data Science and Analytics, 13(4), 335–362. https://doi.org/10.1007/s41060-021-00302-z

[5] Shi, T., Kang, K., Choo, J., & Reddy, C. K. (2018). Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations. Proceedings of the 2018 World Wide Web Conference, 1105–1114.

[6] Hoens, T.R., Polikar, R., Chawla, N.: V: Learning from streaming data with concept drift and imbalance: an overview. Prog. Artif. Intell. 1, 89–101 (2012)

[7] Castelo, S., Almeida, T., Elghafari, A., Santos, A., Pham, K., Nakamura, E.,& Freire, J. (2019). A Topic-Agnostic Approach for Identifying Fake News Pages. Companion Proceedings of The 2019 World Wide Web Conference, 975–980. https://doi.org/10.1145/3308560.3316739

[8] Wani, A., Joshi, I., Khandve, S., Wagh, V., & Joshi, R. (2021). Evaluating Deep Learning Approaches for Covid-19 Fake News Detection (Vol. 1402, pp. 153–163). AAAI 2021

[9] Gundapu, S., & Mamidi, R. (2021). Transformer based Automatic COVID-19 Fake News Detection System (arXiv:2101.00180). arXiv. https://doi.org/10.48550/arXiv.2101.00180

[10] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (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. https://doi.org/10.1089/big.2020.0062

[11] Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure (arXiv:2203.05794). arXiv. https://doi.org/10.48550/arXiv.2203.05794

[12] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. Advances in Neural Information Processing Systems, 30.

Ping-Chun Lin: Introduction, Related Work, Data, Prepossessing, Methods, Discussion, Future Work
Andy Zhang: Problem formulation, NMF Part, Data Processing.
Zimo Wen: LDA topic modeling; Algorithm 1: Modified BERT on small dataset; Error Analysis;
Deyang Zheng: Transformer Modeling, Dataset separation.