



# Real News, Fake AI

Arden Feldt<sup>1</sup>, Christopher Vasquez<sup>1</sup>, Dasha Tran<sup>1</sup>, Kevin Peng<sup>1</sup>, Malak Soubai<sup>1</sup>, and Murad Abdi<sup>1</sup>

<sup>1</sup>The University of North Carolina at Chapel Hill

## Abstract

We propose a machine-learning model fine-tuned based upon the training data of real versus fake news, specifically derived from the existing DistilBERT encoder-only model. Such techniques can be trained to be used upon a variety of input information, including article text, article title, and article source. We present two case studies to show how SFT can be used upon article text and article titles to assess the validity of potential news sources.

*Keywords:* Fake News, DistilBERT, encoder-only model, Transformers, Machine Learning Techniques, Supervised Fine-Tuning

## 1 Introduction

Fake news is defined as a story that is false, fabricated, or unverified [5]. With the widespread availability of digital content and the rise of networking websites, information is highly accessible today, with 88% of fake news found on social media platforms [3]. In the USA, 38.2% accidentally share fake news, which becomes a pressing challenge [9].

In this paper, we explore the use of artificial intelligence for detecting fake news. Specifically, we utilize a text classification method, DistilBERT, a transformer model based on the BERT architecture that is optimized for speed and accuracy [1]. We trained our model using a dataset from Kaggle, consisting of a title, a text, and a label of news articles marking the news as real or fake.

E-mail addresses: feldt@unc.edu, cjvasque@unc.edu, dashatr@email.unc.edu, keypeng@unc.edu, smalak@unc.edu, odd@unc.edu

This work is licensed under a [Creative Commons “Attribution-NonCommercial 4.0 International”](https://creativecommons.org/licenses/by-nc/4.0/) license.



Through our work, we aim to contribute to the development of electronic communication and flag misleading content to promote a more informed society and protect the integrity of digital information.

## 2 Process

### 2.1 Technology Intro

In this paper, we present a use case of encoder-only transformer models. Encoder-only transformers tokenize a variety of supplied input texts, then pass the ensuing values into the actual encoding layers. The final transformer layer outputs a single classifier label; in our use case, it simply represents the expected truthfulness of the given input: 1.0 = false news/present misinformation, 0.0 = true news/no misinformation.

- **Text classification**

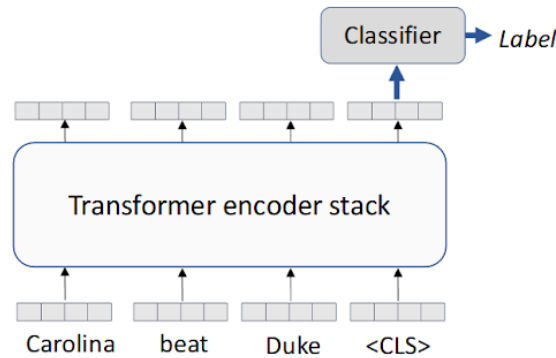
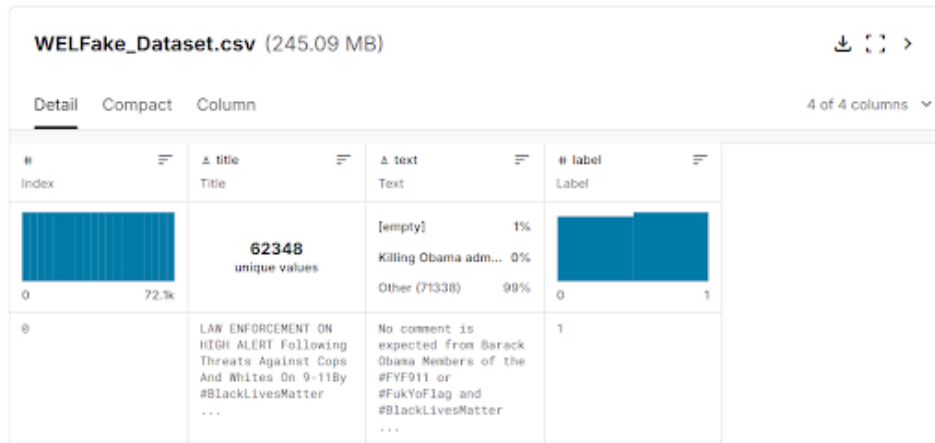


Figure 1: Text classification figure.

### 2.2 Preparing Datasets

In our search for fake news-related datasets, we narrowed the list to 7 potential options. Our favorite was initially a dataset with 40000+ entries [4]. We randomly combined the fake and true csv files, adding a label column of 1's and 0's and split them to be trained on. We moved on and ran the model using the whole text of the article. Unfortunately, our human eyes missed a detail, and the AI didn't: most of the articles in this dataset are from just a few organizations. This made it really easy for the AI to recognize the publisher without much effort.

After learning from the first dataset we chose to work with a different dataset from kaggle [8]. This dataset was chosen for a variety of reasons. The sample size was good with 60,000+ unique entries. There were no superfluous columns, everything there we needed. The four columns were: [Index, Title, Text, Label]. We trained on Title and Text, using 'Label'. Additionally, 'Label' was a binary classification for each article. This was beneficial given the nature of our model.



#	title	text	label
Index	Title	Text	Label
0	62348 unique values	[empty] 1% Killing Obama adm... 0% Other (71338) 99%	0
1	LAN ENFORCEMENT ON HIGH ALERT Following Threats Against Cops And Whites On 9-11By #BlackLivesMatter ...	No comment is expected from Barack Obama Members of the #FYF911 or #FukYoFlag and #BlackLivesMatter ...	1

Figure 2: Column names and first row of the data.

The next step was to download the data locally and use a script to prepare it for the training. The code for this can be found on our GitHub [6]. The first part of the script uses the Kagglehub library to get the data down, and the second part randomly mixes it (using a seed) to then split into training and testing data (also using a seed). We went with an 80/20 split, so 80% of our data was used to train the model, and the remaining 20% was what it was tested on.

From there, we had to upload the data so that everyone in our group had access to it. The data was far too large for a GitHub commit or most easy file sharing. We went through Hugging Face as it allowed us to upload large files, and we were using a Hugging Face model [7].

## 2.3 Training

We experimented using two strategies – using the article title and using the article text input – to compare the results and the effectiveness of the model in classifying the news. The initial DistilBERT models were downloaded from Hugging Face [2], then fine-tuned upon with the datasets retrieved from Kaggle.

## 2.4 Initial Training (Bad Dataset)

During initial training with the first dataset, true and fake news had noticeable formatting differences in how they were presented. True news often contained the location of the report in all capital letters (ex., WASHINGTON) before the content body, whereas such details were lacking in the fake news examples. Because of such discrepancies, the DistilBERT model had no difficulty deciphering the difference between the two classifications, as it only had to look at the formatting of the first few words. As such, poor dataset selection provided no indication of the model’s true ability to notice discrepancies in different information types.

## 2.5 Primary Training

With the second dataset, the inclusion of both title and text for each data point provided the opportunity to compare the model’s ability to differentiate misinformation based upon different amounts and types of data. Both models were fine-tuned with only the relevant section passed into the tokenizer as inputs (text versus title), with the same labels kept constant for both runs.

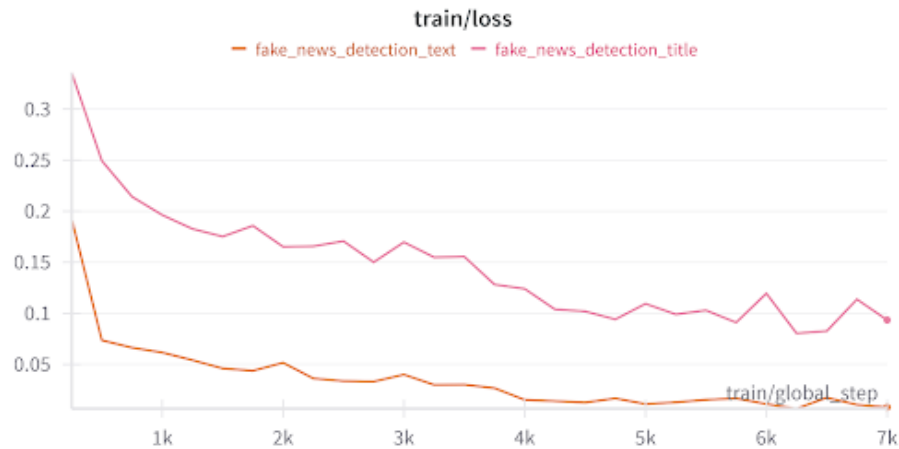


Figure 3: Text classification figure.

### 3 Results and Conclusion

#### 3.1 Initial Training

The flawed nature of the initial training data provided to the model enabled a 99.8% tested accuracy rate, the highest value out of all the models. However, due to the model simply recognizing formatting errors that may only be relevant for specific styles of transcripts, these results are not conclusive of any true gain.

#### 3.2 Final Training

When fine-tuned with the second dataset, the title-only model was able to achieve an accuracy level of 95.6%, while the text-only model was able to achieve an accuracy of 99.4%, after running 2 epochs of training and selecting the highest performing model.

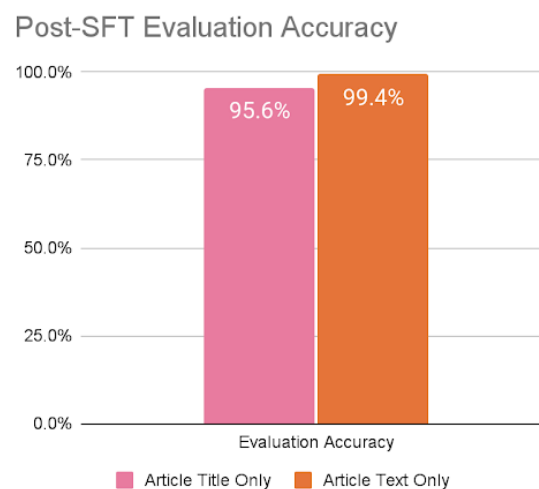


Figure 4: Evaluation Accuracy.

### 3.3 Conclusion

SFT with an encoder-only based transformer model such as DistilBERT provides significant and highly promising results in fake news classification. Both title-only and text-only models were able to achieve over 95% accuracy, and the text-only model was able to achieve over 99% accuracy. The title-only method provides a potential streamlined option, due to the lower token inputs of titles, while text-only methods provide highly accurate classification abilities, albeit at a slightly lower pace. Therefore, both models possess advantages that enable potential use cases upon further optimization and study.

### Data Availability

Availability of all data and codes used to process or generate the data is required in a [publicly accessible and persistent](#) repository at the time of publication.

### References

- [1] Distilbert, . URL <https://www.sciencedirect.com/topics/computer-science/distilbert>. ScienceDirect topic page.
- [2] Distilbert, . URL [https://huggingface.co/docs/transformers/en/model\\_doc/distilbert](https://huggingface.co/docs/transformers/en/model_doc/distilbert). Hugging Face documentation.
- [3] C. Beauvais. Fake news: Why do we believe it?, 2022. URL <https://pmc.ncbi.nlm.nih.gov/articles/PMC9548403>.
- [4] clmentbisailon. Fake and real news dataset, Apr. 2024. URL <https://www.kaggle.com/datasets/clmentbisailon/fake-and-real-news-dataset>.
- [5] S. Desai and J. A. Oehrli. “fake news,” lies and propaganda: How to sort fact from fiction: What is “fake news”?, June 2023. URL <https://guides.lib.umich.edu/fakenews>. University of Michigan.
- [6] A. Feldt. Arden-feldt/comp560finalproject: Group’s repo for final project, 2025. URL <https://github.com/Arden-Feldt/Comp560FinalProject>. GitHub.
- [7] A. Feldt. Newcombined, Apr. 2025. URL <https://huggingface.co/datasets/Feldt/NewCombined>. Chapel Hill. Retrieved April 2025.
- [8] S. Shahane. Fake news classification, Oct. 2023. URL <https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification>.
- [9] A. Watson. Frequency of seeing false news online u.s. by age 2023, 2024. URL <https://www.statista.com/statistics/1462057/false-news-consumption-frequency-us-by-age/>.