NLP Final Report: Fake News Detection

Shukan Shah
sshah430@gatech.edu

Jas Pyneni
jpyneni@gatech.edu

Abstract

"Fake News" is a method of news used to intentionally spread misinformation to the general public through both regular news sources and social media. Though the concept of yellow journalism has existed for a long time, this deceptive technique of online fake news has recently risen to prominence due to the widespread reach of the internet. Fake news is generally spread for generating revenue, influencing political opinions, or causing mass hysteria and has recently victimized many during the past presidential election season and the COVID-19 global crisis. It is hard to track down and get rid of fake news because of the expanse of the Internet and the rate at which information, fake or real, travels. As a result, the goal of our project was to distinguish real news from fake news. We focused on utilizing data augmentation methods to improve existing fake news detection models. For this purpose, we collected large datasets of real news from reputable sources and fake news from online datasets and conducted a comparative study of three data augmentation methods to understand how to improve existing detection models. We experimented with Naive Synonym Replacement, Back translation, and Grover neural news generation to augment our data and trained a Naive Bayes model, LSTM classifier, and a BERT model. The subsequent analysis showed us that Back translation works as the best data augmentation technique to enhance the best classifier model, BERT, for this fake news detection problem. Using the model, we built a user-friendly web tool that takes in text and classifies whether that piece of news is fake or real. The code for our data augmentation and modeling is hosted at the following link:
https://github.com/jpyneni3/Fake_News_Detector.

The web tool can be accessed at:
https://shukieshah.github.io/VerifAI/

1 Introduction and Related Work

1.1 Motivation

The motivation for this project is to build a tool that, given a text, will be able to accurately classify that text as real news or fake news. Our goal was to reimplement existing fake news classification models and to improve them using data augmentation techniques. For this goal, we focused on conducting a comparative analysis on three data augmentation techniques to extend a state of the art BERT model. Through our initial explorations, we discovered that one of the reasons this classification problem is nontrivial is because of a lack of abundance in labeled fake news data. This highlights the importance of our data augmentation approach to draw further insights using the resources we have. Though the central goal of our project remained the same from the beginning, we expanded our choice of augmentation to include three interesting techniques as we learned more about current NLP methodology. Our original choice of data augmentation technique was just neural fake news generation (using an external tool). We have expanded to also test and analyze naive synonym replacement and back translation as other data augmentation techniques.

1.2 Related Work

As the main goal of our project was to reimplement existing models and enhance with data augmentation, we drew a lot of insights from related work to bring together many ideas into our problem space.

1.2.1

Clarity Insights (2018), AI partner of Accenture, introduced a Naive Bayes model approach to fake news detection through building a model that was trained on a toy dataset (Insights, 2018). We referenced this model to expand and build our initial

baseline model, as we will explain later.

1.2.2

Researchers at Google found that BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art technique to build language representation models that require very minimal additional work to be adapted for many NLP tasks, one of which is the classification approach (Devlin et al., 2018). We adapted this BERT model to serve as another baseline model for our problem to see if we can improve the results using our data augmentation.

1.2.3

Wang, from the Computer Science Department of UC Santa Barbara, developed a labeled benchmark dataset for fake news detection, known as the LIAR dataset (Wang, 2017), by manually labelling 12.8K phrases from POLITIFACT.com. This dataset was created to be a large contribution towards solving the problem of lack of labeled data for detecting fake news and is something that we used ourselves as one of the sources for the initial seed data upon which we ran data augmentation techniques.

1.2.4

Zellers et al. (2019) designed Grover (Zellers et al., 2019), a popular generator and detector transformer model, to detect neural fake news by understanding how the AI itself can generate neural fake news. We incorporated this research into our work by using Grover as an external tool to generate fake news for one of our data augmentation techniques. Although our focus was detecting human fake news as opposed to neural fake news, we wanted to analyze how Grover augmentation affects our models in this classification problem.

1.2.5

Facebook AI Research developed four neural translation methods to submit to WMT19's shared translation task, an annual conference which focuses on developing statistical machine translation models for translating news(Ng et al., 2019). FAIR's methods utilize the fairseq framework to provide translations of news from English to German, German to English, Russian to English, and English to Russian. We combined FAIR's neural translation methods to build our back translation technique and help in our data augmentation process.

1.2.6

Wei and Zou found EDA (Wei and Zou, 2019), easy data augmentation techniques, enhances NLP classification tasks to improve performance of neural network models. One of these suggested EDA techniques is synonym replacement, which we leveraged the knowledge from to build our Naive Synonym Replacement data augmentation technique.

2 Methods

2.1 Data

For this project, we gathered a dataset of real news and human fake news and subsequently used data augmentation techniques to grow that data. We got our base news datasets from three sources: RSS feed scraping, Kaggle, and the LIAR dataset. The scraper tool uses RSS web feeds to download website data in a set format from reputable sources, such as NY Times, Washington Post, and NPR, to get real news. Information about the specific RSS web feeds we used can be seen in the News RSS json file in the Notebooks directory in our Github repository. We used the Python article library to access and download news content from these RSS feeds. From Kaggle, we got three different datasets: one with just fake news, one with just real news, and one with a mix of both. The fake news Kaggle dataset (A) (Risdal, 2017) downloaded data from websites tagged as "fake" by a Chrome extension (Wakefield, 2016). The real news Kaggle dataset (B) (Thompson, 2017) was built with an RSS scraper and the BeautifulSoup library, similar to our first data source, to download and extract real news from reputable sources. The mixed news Kaggle dataset (C) (Club, 2018) was developed by the UTK Machine Learning Club. The last source for our baseline data is the LIAR dataset (Wang, 2017) that was built at the UC Santa Barbara Department of Computer Science from manually labelling 12.8K short statements found on POLITIFACT.com for the purpose of fake news detection. The below table summarizes statistics for each of the datasets we have collected:

Source	Total Articles	No. of Real	No. of Fake	Avg Size
RSS Web Scraper	2347	2347	0	377 Words
Kaggle (A)	12273	0	12273	373 Words
Kaggle (B)	24646	24646	0	441 Words
Kaggle (C)	20800	10387	10413	419 Words
LIAR	12791	5657	7134	10 Words
TOTAL	72857	43037	28820	357.64 Words

The last row shows the total of each type of article before data augmentation.

2.1.1 Data Augmentation

This baseline data was used as seed data and increased through data augmentation for experimentation. We used three different types of data augmentation in order to grow out data for a comparative analysis. The three data augmentation techniques were Naive Synonym Replacement, Back Translation, and Grover Generation.

Naive Synonym Replacement(NSR): The first type of data augmentation we did was Naive Synonym Replacement which randomly replaced some words in a given text string with synonyms of those words, based upon NLTK platform's WordNet Interface, a lexical database of English. The idea and the reference of this augmentation came from Wei and Zhou's EDA proposals (Wei and Zou, 2019). Our Naive Synonym Replacement method randomly selects 30% of the non stop words from a given input string and replaces those words with a synonym from WordNet. We found 30% to be the size of change to maintain structure and meaning of input sentences based upon testing. We ran the Naive Synonym Replacement on our two datasets that had both real news and fake news, the mixed Kaggle (C) dataset and the LIAR dataset, to get a data augmentation output for both fake news and real news.

Back Translation (BT): The second type of data augmentation we did was back translation which uses Neural Machine Translation models to translate a given text into a foreign language and then translates that foreign output back into English (Ng et al., 2019). The idea behind back translation is that contextual information is persisted across two representations of the same information, while increasing data size. The two NMT models are that we used are transformer.wmt19.en-de.single_model and transformer.wmt19.de-en.single_model which use the big Transformer architecture in fairseq, a sequence modeling toolkit in PyTorch, to translate from English to German and German to English, respectively. We coupled both the models together to run our back translation, relying on Google Colab GPU's to translate our mixed datasets, the mixed Kaggle (C) dataset and the LIAR dataset, from English to German and then back to English. Similar to Naive Synonym Replacement, we ran this data augmentation method on our mixed datasets to get an output that is relatively balanced in fake news and real news, but we limited to only texts under 3500 characters in order to ensure we finish running our method within the allotted time by Google Colab.

Grover: The third augmentation we have implemented is the neural generation of fake news by developing an automated pipeline to feed news headlines from existing articles into the Grover transformer model (Zellers et al., 2019). We fed all the real news and fake news text from the LIAR dataset into the Grover model. Using the input as the starting point, Grover generated 12.8k fake neural-generated news articles.

The following examples are excerpts of text taken from our data augmentation:

Original Phrase (**Source: LIAR DATASET, Label: Fake**):

Says the annies list political group supports third-trimester abortions on demand.

Naive Synonym Replacement:

Enunciate the annies listing political group supports third-trimester miscarriage on demand.

Back translation:

The group supports third-trimester abortions on demand.

Grover:

Anne Hathaway, Jennifer Garner, Josh Gad and

Grégory Clement have signed on to star in Molly Parker's still untitled political thriller, which also will feature Pamela Adlon and Jackie Long.

The below table summarizes statistics for the data generated through augmentation:

Method	No. of Real	No. of Fake	Total No. Articles
NSR	16044	17547	33591
BT	11714	11200	22914
Grover	0	12790	12790

2.1.2 Data Expansion

In our midway report, the only data we collected before data augmentation was very limited: 905 scraped real news and 4702 human fake news from Kaggle. As we learned that this is more consistent for a toy dataset, we expanded our baseline pre-augmentation by collecting more label real and fake news. As explained above, we found two more Kaggle datasets, one which gave us a pretty balanced set of 20716 articles and another which has a collection of 24656 real news articles from reputable sources. We further expanded our data by adding the LIAR data, which contains both real and fake phrases

We also placed an emphasis on preprocessing our data to ensure better results and efficient models. We semi cleaned the initial seed data to ensure that we fed our data augmentation methods non null data. We also aggressively cleaned the seed data and the resultant data of data augmentation, before feeding to models, to remove stop-words and punctuation. In both cleaning phases, we also took out emojis as emojis are foreign characters that can easily break certain models.

2.2 Models/Analysis

For the task of fake news detection, we investigated and implemented three popular models in the text classification space:

1. **Naive Bayes**: The Naive Bayes classifier is a simple, yet effective classification model based on Bayes' theorem that assumes strong independence among features. The reason why this model is effective is because it is much faster than deep learning approaches during training and testing. Further, the model has been shown to outperform highly sophisticated classifiers when given access to only minimal amounts of data.

   We reimplemented Naive Bayes based on the Clarity Insights fake-news detector model (Insights, 2018) and used TF-IDF on the text to convert all the string data into vectorized numerical data. We intended to use Naive Bayes as a strong baseline to compare against more sophisticated NLP approaches.

2. **Long Short-Term Memory (LSTM) Network**: LSTMs are a powerful variant of Recurrent Neural Networks and have been shown to be extremely useful for sequence to sequence classification tasks. In this type of deep neural network, three types of gates (input, forget, and output) serve to regulate the amount of information passed on to each subsequent layer. In essence, these gates help the network overcome the vanishing gradient problem in traditional RNNs and help "remember" long-term dependencies in data such as text passages and news articles. Bidirectional LSTMs perform even better as they are able to use information from the "past" and "future" to develop better context.

   We reimplemented a standard 2-layer bidirectional LSTM model using a Pytorch guide (Pai, 2020) and used pretrained Glove word embeddings (with a dimension size of 100) to convert our input data into vector representations. We used pretrained embeddings rather than having the network learn the embeddings from scratch because doing so increased our testing accuracy during experiments.

3. **Bidirectional Encoder Representations from Transformers (BERT)**: BERT is a revolutionary "deeply bidirectional" model, based on the transformer architecture (Devlin et al., 2018). It is pretrained on the entirety of Wikipedia and the Google Books corpus and can be used to extract high-quality features from text data. What's incredibly useful about the model, however, is that it can be efficiently fine-tuned on a variety of downstream tasks (such as text classification) by adding a few simple output layers. It is currently the state-of-the-art for most NLP tasks and has caused a breakthrough in the field.

We reimplemented a standard BERT classifier model, using the popular Hugging-Face transformers library (Wolf et al., 2019) and an online tutorial (McCormick, 2019). The specific variant we used was the "base-uncased" version due to resource limitations. After cleaning/preprocessing and tokenizing our data, we fine-tuned our model by training on our corpus of news articles. BERT served as our main model for the project and further details are discussed in the Results section.

After performing a comparative analysis on each of these models, we enhanced and adapted our best model using the data augmentation techniques discussed in the previous section. Ultimately, we wanted to figure out whether or not data augmentation would improve the baseline performance of our classifier and if so, by how much. Moreover, we sought to provide insights into which augmentation techniques performed the best and address the advantages and disadvantages of certain methods over others.

2.3 Baseline Models

A. In the first set of experiments (Experiment Set A), we performed a comparative analysis by training the Naive Bayes, LSTM, and BERT models on our original, non-augmented dataset. The purpose of these experiments were to compare the performance of our main model, BERT, against the other models we implemented. This would give us further insight into the effectiveness of various models for the task of fake news detection. Thus, in these experiments, our Naive Bayes and LSTM classifiers served as the baseline models.

B. In the second set of experiments (Experiment Set B), we used our best classifier from Experiment Set A (BERT) to evaluate and compare our different data augmentation techniques. Specifically, we evaluated BERT with Grover augmentation, BERT with naive synonym replacement, and BERT with back translation against the baseline of BERT with no augmentation. This was the primary contribution of our project.

The details and analysis of each set of experiments are further expanded in the Results section of this paper.

3 Results

3.1 Experiment Setup A

As discussed above, Experiment Set A consisted of a comparative analysis between different models trained on the original non-augmented dataset. The goal was to ascertain the performance of various approaches of fake news detection and determine the best classifier to use for Experiment Set B.

Although our original dataset size consisted of approximately 73,000 instances, it was imbalanced. Hence, we omitted 13,000 real news instances to create a balanced dataset of 60,000 instances. We used an 80/20 train/test split for our data. For further description regarding the specific models, please refer to the Models/Analysis section.

3.2 Results Comparison A

The training and testing accuracies for each model is reported below.

Model	Avg Train Accuracy	Avg Test Accuracy
Naive Bayes	80.31%	79.05%
Bidirectional LSTM	90.99%	88.84%
BERT (fine-tuned)	**92.53%**	**90.47%**

As the results demonstrate, we found that BERT far out-performed Naive Bayes and had a small improvement over the LSTM classifier. These results matched our expectations because Naive Bayes, unlike an LSTM, is a simplistic classifier and cannot capture the complex interdependencies between long text passages. Bidirectional LSTMs do a much better job of learning deeper meaning representations but fall slightly short to fine-tuned BERT models that already have an ingrained/pretrained representation of the English language.

3.3 Experiment Setup B

Experiment Set B was a comparative analysis of the different data augmentation techniques we implemented to grow the size of our dataset and improve our model's performance. The model used to evaluate the augmentation techniques was BERT, as it was the highest performing classifier from Experiment Set A.

The specific BERT model that was used was the smaller, "base-uncased" version due to resource limitations. Our model truncated all sentences to

BERT's maximum token length of 512 as we were classifying longer news articles. Our batch size was 32 and dropout rate was 10%. We finetuned the model on the augmented datasets for a total of 5 epochs with an Adam optimizer and learning rate of 2e-5. These optimal parameters were found by hyperparameter tuning our model on validation sets. Some of our other parameters were set to recommended defaults. We used an 80/20 train/test split with 10% of training data used for our validation set.

3.4 Results Comparison B

The following table shows the comparison between the performance of BERT fine-tuned on various augmented datasets.

Data Augmentation	Dataset Size	BERT Avg Train Accuracy	BERT Test Accuracy
None	59857	92.53%	90.47%
Grover	59857 + 25790* = 85647	91.41%	89.70%
NSR	59857 + 33591 = 93448	93.91%	92.14%
BT	59857 + 22914 = 82771	**96.52%**	**94.60%**

***Note:** We only augmented 12790 fake news articles using Grover, as it serves solely as a fake news generator. We offset this augmentation using 13000 of the extra real news data we omitted earlier in Experiment Set A.

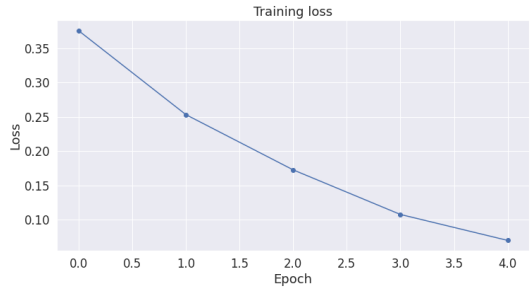

Figure 1: This is the training loss for our best model trained on back translation augmented data

Overall, the results supported our initial hypothesis that data augmentation would meaningfully improve model performance. Two of the three data augmentation techniques resulted in a substantial, nontrivial improvement in the accuracy of the BERT model. Particularly, fine-tuning BERT on the back-translated dataset caused the greatest jump in test-accuracy to an impressive 94.6%. Naive Synonym Replacement also improved the baseline test accuracy of 90.47% to 92.14%. Though the general takeaway is that data augmentation improves the accuracy of classifying fake news, that back translation slightly improves the model more than Naive Synonym Replacement can be explained through NSR being naive whereas back translation captures more of the context. For NSR, as the replacements are naive, each word is considered independently and as verbs in different tenses are considered synonyms, such replacements, though hold the same structure as the original sentence, could alter the content. Back translation on the other hand works through contextually translating the whole text into German and back and in the process, is able to better capture the meaning as back translation is essentially paraphrasing a given text. This difference is explored in this example:

Original Phrase (**Source: LIAR dataset**):

when did the decline of coal start? it started when natural gas took off that started to begin in (president george w.) bushs administration.

Naive Synonym Replacement:

when did the decline of char start? it come out when instinctive brag subscribe off that come out to commence in (president george ii w.) bushs administration.

Back translation:

When did the decline of coal begin? it began when natural gas started flowing in (President George W.) Bush's administration.

Here we can see that the back translation of the original statement contains the same message but in different words whereas the synonym replacement example does not make as much sense. By naively replacing 'natural' with 'instinctive', a synonym of 'natural', we lose the context that the phrase is specifically referring to natural gas. These differences in terms of involvement with context cap-

turing also present themselves in the run-time of these methods as Naive Synonym Replacement only took 3 minutes to augment almost 33.5k data points whereas Back translation took 21 hours for 23k data points.

Grover augmentation, unfortunately, resulted in a slight dip in training and testing accuracies. This can be explained by the fact that, since Grover fake news is generated from real news examples, the augmented data shares a similar distribution to real news articles. For example, our model mistakenly classified some of the Grover generated fake news as real news.

3.5 Software Tool

For the final phase of this project, we took our best performing model from Experiment Set B – BERT with Back Translation – and developed a Python Flask Rest API to serve model predictions in real-time. The API and saved model files are deployed and hosted on an AWS EC2 instance. We then created a user-friendly front-end React application that interacts with the API. The tool can be accessed through this link: https://shukieshah.github.io/VerifAI/. The github repository for the tool can be found at: https://github.com/shukieshah/VerifAI.

We hope that VerifAI serves as a useful tool for the Georgia Tech community. Please note that the tool is, by no means, perfect. The tool is not meant for neural fake news detection and sometimes classifies fake news as real. This is due to the wide variance of fake news in the real world that our model was not trained to detect. After all, this is precisely what makes reliable fake news detection such a difficult problem to solve!

3.6 Work Division

For the work division, Jas was responsible for collecting, cleaning and pre-processing the data, from building a web scrapper tool to get real news to exploring the LIAR dataset. He built and ran the Naive Synonym Replacement and Back translation data augmentation techniques. He was also tasked with building the basic Naive Bayes implementation. Shukan aided in data augmentation by implementing an automated pipeline to generate neural fake news from Grover. He was also responsible for implementing and experimenting with the LSTM and BERT models as well as building the software tool.

4 Conclusion

Our work in this project brought together various related work in separate problems to specifically help in our target problem space of detecting fake news. Therefore, we were able to leverage Wang's LIAR dataset along with other datasets as seed data for data augmentation by by coupling together FAIR's Neural Translation models. This combination of the seed data and the back translated data proved to useful in enhancing the accuracy of the BERT classifier, allowing for a highly accurate fake news detection tool. In the process, we were able to draw comparisons with the Grover generator and Naive Synonym replacement for the biggest takeaway that data augmentation definitely benefits this problem space of fake news detection. Overall, by packing our findings together and developing VerifAI, we have built a tool to help detect fake news. In the future, this work can be expanded to try these data augmentation techniques in other problem spaces to test the widespread effects of these tools in the NLP field. This project could also possibly be further improved by using the latest version of FAIR's Neural Translation model as Facebook seems to be building a new model each year with modifications and improvements to the translations. Also, to combat the documented errors of Naive Synonym Replacement, we can run an experiment to see how n-length context synonym replacement does as a method for data augmentation, where we can replace synonyms by considering a group of n neighboring words together.

References

UTK Machine Learning Club. 2018. Fake news.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

C. Insights. 2018. Using nlp and ai to detect fake news with 99% accuracy. *Clarity Insights*.

Chris McCormick. 2019. Bert fine-tuning tutorial with pytorch. *Chris McCormick*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission.

Aravind Pai. 2020. Build your first text classification model using pytorch. *Analytics Vidhya*.

Megan Risdal. 2017. Getting real about fake news.

Andrew Thompson. 2017. All the news.

Wakefield. 2016. Fake news detector plug-in developed. *BBC*.

William Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. pages 422–426.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32*.