CSCI 5525 Final Project: Fake News Detection

Collin Evans, Franklin Xavier, Vincent Cao, Shesha Sairam, Prakrit Sinha

Introduction

If you had to guess whether the following statements were true or false, could you do it?

Speaker	"News"	Real News?
Blog Poster	"Hillary Clinton in 2005 co-sponsored legislation that would jail flag burners."	true
Tom Barret	"Under Tom Barrett's leadership, violent crime in Milwaukee has decreased by over 20% -- to its lowest levels in more than 20 years."	false
Associated Industries, Florida	"The working tax cut created over 40,000 new jobs in just the last four years."	false
'oregon-2014'	"In fiscal year 2011-2012, 4,191 abortions were paid for by taxpayers via the Oregon Health Plan."	true
Bernie Sanders	"African-American youth unemployment is 51 percent. Hispanic youth unemployment is 36 percent."	half-true

Detecting fake news is a very pressing need in today's society where millions of people fall victim to believing and spreading fake news posts all the time. This can vary widely from a minor white lie about a political candidate to something quite serious like misinformation about COVID-19. Trying to detect fake news using neural networks and machine learning is one of the forefront challenges in today's world. By taking this on as a project, the hope is to be able to make some progress in a relevant area of research. The goal for this project is to detect fake news using various machine learning methods on the "Liar" Dataset from HuggingFace.

Dataset

The "Liar" Dataset is mostly composed of social media and other online platform posts with various speakers from holders of political offices, candidates, newscasters, blog posters, and more. The dataset comes with this list of features: statement, subject, speaker, position/job, state, and political party affiliation (if appropriate). The labels for the data are split into the following categories: true, mostly-true, half-true, barely-true, false, and "Pants on Fire". One of the real strengths of this dataset is that each statement has been evaluated by researchers who determined a score of how truthful it is. This range allows for variability instead of strictly rating something completely true or false. For example, a statement where the speaker is spreading dangerous misinformation would be labeled as "Pants on Fire" in the dataset. Although the fields for "subject" and "speaker" are all present, two of the fields have some missing. The first is "position/job" with 28% missing, and the second is "state" with 22% missing.

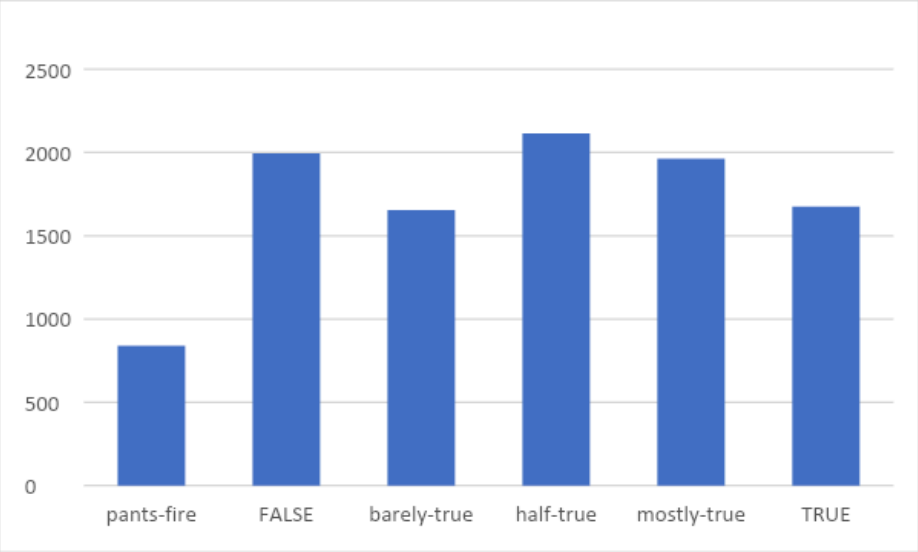

Figure 1. Label counts for each truthfulness value in the "LIAR" range.

As seen in Figure 1, the label counts are evenly distributed except for the "Pants on Fire" category which has less than ½ of the number of entries compared to the other categories. This is a factor that likely contributes to some inaccuracy in performance which can be seen in Figure 2.

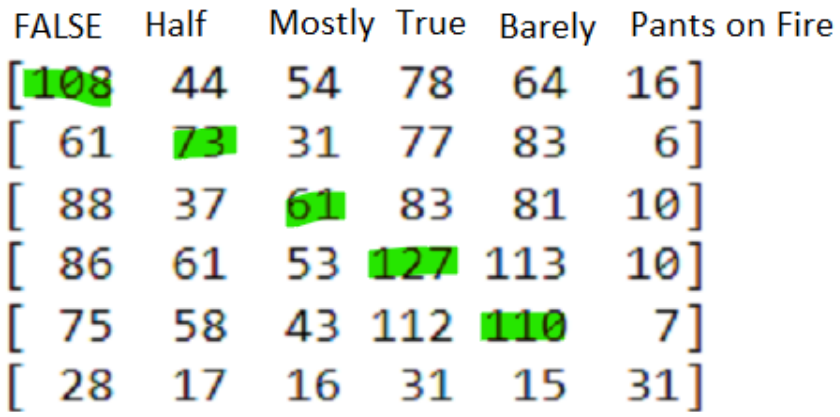

Figure 2. Confusion matrix with accuracy scores for truthfulness values.

In Figure 2, one can see that the "Pants on Fire" category on the far right had the lowest accuracy scores in the confusion matrix. While the other categories did not have particularly high values, they were noticeably better when compared to the far-right category.

Baseline Method

To start off with, a logistic regression model was used to set the benchmark. The metrics across the six categories are shown below, and it is safe to say that there is a lot of room for improvement.

```
Accuracy: 0.2490234375
Confusion Matrix:
 [[108  44  54  78  64  16]
 [ 61  73  31  77  83   6]
 [ 88  37  61  83  81  10]
 [ 86  61  53 127 113  10]
 [ 75  58  43 112 110   7]
 [ 28  17  16  31  15  31]]
Classification Report:
              precision    recall  f1-score   support

       FALSE       0.24      0.30      0.27       364
        TRUE       0.25      0.22      0.24       331
 barely-true       0.24      0.17      0.20       360
   half-true       0.25      0.28      0.27       450
 mostly-true       0.24      0.27      0.25       405
  pants-fire       0.39      0.22      0.28       138

    accuracy                           0.25      2048
   macro avg       0.27      0.24      0.25      2048
weighted avg       0.25      0.25      0.25      2048
```

Figure 3. Metrics for baseline using logistic regression.

Methods and Algorithms

This report utilized five different models for a comprehensive performance analysis:

BERT:

Bidirectional Encoder Representations from Transformers. This method was chosen for multiple reasons. Using its multiple layers of transformers, it can encode meaning and context from both left and right directions. It is also able to capture the semantic and syntactic features of the text. I liked this quote someone gave about it: “BERT can understand the difference between "I love this movie" and "I love this movie, not" by taking into account the word order, punctuation, and negation.”

BERT works by being trained on large amounts of text data. During this training, it learns to generate word representations by bidirectionally considering contexts. After this initial training, fake news detection can be implemented by finetuning the model to fit the smaller “LIAR” Dataset. BERT tokenizes the text representation into high-dimensional embeddings that are contextually aware. It utilizes a transformer architecture that consists of self-attention layers. Torch was used to create a DataLoader, and AdamW was used to create an optimizer for this model.

RoBERTa:

Robustly Optimized BERT Approach. This method was chosen because it boasts improved performance over BERT by being trained on 10 times the data: 160 GB instead of just 16 GB. It has been

previously used with data such as CC-News articles and OpenWebText datasets, so it seemed similar to the data that this project was using.

RoBERTa is different from the earlier mentioned BERT model in the way that it is trained with a larger set of data and has more training steps. The tokenization for this model is more fine-grained, which means it is more effective than BERT with words that are outside its vocabulary.

DistilBERT:

This method was chosen because compared to BERT, it can be fine-tuned with less data. Given our limited amount of data, this seemed like a good option. It also boasts faster performance and simpler coding.

Distilled BERT is a smaller and faster version of BERT, and overall works very similar to the BERT model. The only major difference is that it uses layer distillation to compress knowledge into fewer layers compared to the larger models such as BERT.

LSTM:

Long Short-Term Memory. This method was chosen because of its great understanding of contextual information in a sentence. Its performance on small datasets is also of a high standard. It can also handle different sequence lengths while using lower computations and memory while training. Bi-LSTM was also used (bidirectional LSTM) to capture patterns from both directions of the data sequence and was tested separately from LSTM. Implementation was done using TensorFlow and Keras, along with NLTK for text processing. Dropout layers were used to help with regularization.

Long Short-Term Memory is a model that is designed to address the long-term dependencies in many sequential data. This model is able to utilize memory cells and gating mechanisms to capture and retain information over an extended period. For the LIAR dataset, each sentence is converted to tokenized sequence data which is then used as the input for this model. The model itself is built with an embedding layer that embeds the input data. This is then followed by the LSTM layer with a 32-unit memory cell and a dropout rate of 0.2. The data is then flattened and sent to subsequent dense layers which consist of a ReLU layer followed by the output Sigmoid layer. A similar structure was used for the Bidirectional LSTM. However, both models performed similarly even with hyperparameter tuning.

CNN:

Convolutional Neural Network. This method was chosen because recent progress has been made in sentiment analysis using CNNs leveraging their ability to do pattern detection without extensive feature engineering. They are also good at capturing context for short phrases, which closely match the structure of our data. They are less sensitive to noise in the data and can handle variable input sizes. This was also implemented using TensorFlow with Keras, and NLTK for data processing.

The convolutional neural network model that this project uses is for binary classification, so the labels in the data were changed. The text representation of the statements was converted into a numerical representation by a tokenizer. This data was then passed through various layers to detect patterns and features. These layers consisted of ReLU and sigmoid for this project's implementation. An optimizer was also used after it went through these layers, and then the model was fit and evaluated. Hyperparameter tuning was done on the convolutional neural network and the best hyperparameter settings were a learning rate of 0.1, a batch size of 32, and a dropout rate of 0.1.

Results

Below you can see the different implemented models and their respective accuracy scores.

Model	BERT	RoBERTa	DistilBERT	LSTM	CNN
Accuracy (Multi-label)	29%	28%	28%	-	-
Accuracy (Binary)	69%	67%	65%	54%	64%

For more detailed metrics and confusion matrices, please look at the ipynb notebooks.

An underlying theme for our comparative study was that due to the complex nature of the dataset, multiclass classification was yielding poor results. We then decided to simplify the labels and group them so as to make them binary. (True={true, mostly-true}, False={ barely-true, false, half-true, pants on fire}). This, as can be seen in the table above, gave us significantly improved results. We came up with several intuitive reasons which could explain this:

1. **Simplifying the Problem:** Binary classification simplifies the task by reducing the number of classes. This can make the problem more manageable and less complex for the model to learn.
2. **Imbalance in Classes:** Multilabel classification often requires handling imbalanced classes, where some labels may have significantly fewer samples than others. This can make it harder for the model to learn patterns effectively across all classes, leading to lower performance.
3. **Difficulty in Discrimination:** The distinctions between the class labels might be subtle and challenging for the model to accurately differentiate. Grouping these classes into binary categories could help by providing clearer boundaries for classification.
4. **Feature Representation:** The features in the dataset might be more naturally suited for binary classification. Certain features or combinations of features may exhibit strong associations with the simplified binary labels, leading to higher accuracy.

Finally, it was observed that BERT and RoBERTa outperformed all other models, and was a significant improvement on our baseline benchmark set by the logistic regression model.

Conclusion

Much of what has been covered underscores the complexities and challenges in detecting fake news. Even with lacking accuracy applying various models to the problem revealed much. So far no model is a silver-bullet solution to this problem and this not referring to the limited attempts above. No, fake news continues to impact our society. One limitation in today's scientific community to try and work on this is the limitation of not having large datasets with verified truth/false labels. The liar dataset makes a step forward in this regard, but we need more.

Future work for this project includes looking at other datasets outside of the “LIAR” dataset since the statements used are relatively short compared to others. The models may perform better when implemented with a dataset that includes more context. Another thing that could be worked on in the future is connecting multiple datasets together to create a graph neural network.

References

1. Shu, Kai, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. "Fake news detection on social media: A data mining perspective." *ACM SIGKDD explorations newsletter* 19, no. 1 (2017): 22-36.
2. Zhou, Xinyi, and Reza Zafarani. "A survey of fake news: Fundamental theories, detection methods, and opportunities." *ACM Computing Surveys (CSUR)* 53, no. 5 (2020): 1-40.
3. Kaliyar, Rohit Kumar, Anurag Goswami, and Pratik Narang. "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach." *Multimedia tools and applications* 80, no. 8 (2021): 11765-11788.
4. Kaliyar, Rohit Kumar, Anurag Goswami, Pratik Narang, and Soumendu Sinha. "FNDNet–a deep convolutional neural network for fake news detection." *Cognitive Systems Research* 61 (2020): 32-44.
5. Bahad, Pritika, Preeti Saxena, and Raj Kamal. "Fake news detection using bi-directional LSTM-recurrent neural network." *Procedia Computer Science* 165 (2019): 74-82.
6. Pavlov, Tashko, and Georgina Mirceva. "Covid-19 fake news detection by using BERT and RoBERTa models." In *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 312-316. IEEE, 2022.
7. Dentith, Matthew RX. "The problem of fake news." (2016).
8. Ahmed, Alim Al Ayub, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. "Detecting fake news using machine learning: A systematic literature review." *arXiv preprint arXiv:2102.04458* (2021).
9. Wang, William Yang. "" liar, liar pants on fire": A new benchmark dataset for fake news detection." *arXiv preprint arXiv:1705.00648* (2017).
10. Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).