University of Toronto

ECE1786

Final Report

HonestEye

Mei Li - 1002358094

Mohammadreza Safavi - 1010510617

Professor Rose

Word count: 1976

Penalty: 0

Introduction

The goal of our project is to detect a lie from 3 sentences, where the other two sentences are truths. Despite the difficulty of deception detection, this project aims to identify lies within the specified constraints. For this project, the statements must be relevant to Canadian cities. In addition to a falsehood, the lies must also have an element of deception. It is important to explore the performance of natural language processing models against lie detection as it may provide interesting insights.

Illustration

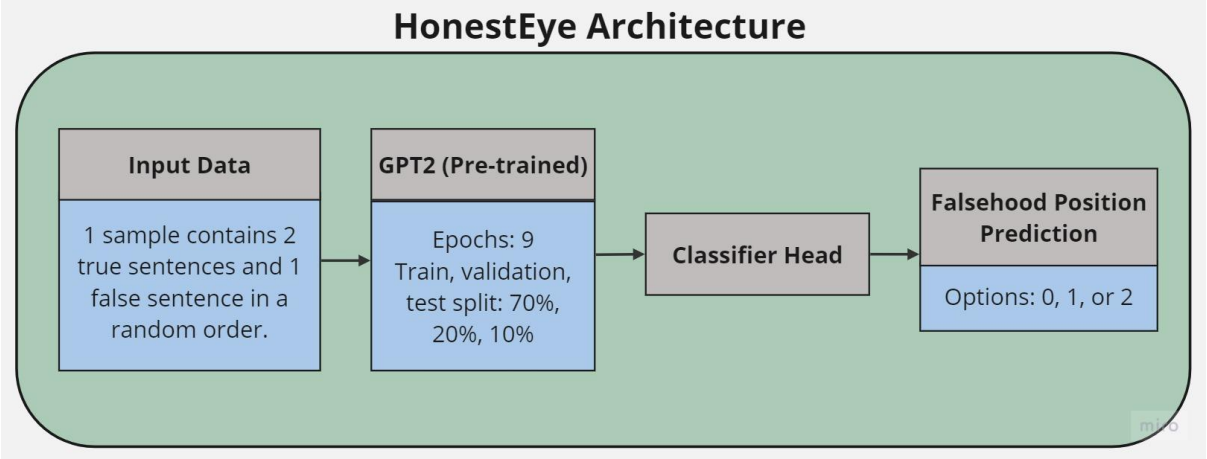

Figure 1. This figure depicts the architecture of the project model.

Background

1) Paper title: Deception Detection for News: Three Types of Fakes
This paper [1] describes 3 different types of fake news that nowadays everybody may be exposed to. These types are: Serious Fabrications, Large-Scale Hoaxes, and Humorous Fakes. It is important to understand how to categorise fake news, as it would help understand what makes people confused and how to feed them convincing misinformation. The paper also discusses the requirements for a deception detection system to be able to detect fake news effectively.

2) Paper title: Deception detection with machine learning: A systematic review and statistical analysis
This survey paper [2] provides insight in the field of deception detection with machine learning approaches. This paper covers a wide variety of methods, monomodal, bimodal and multimodal, with different machine learning architectures such as random forest, SVM, neural networks, etc. The authors detail observations about culture and language in the field which is helpful for creating an accurate lie detection classifier.

Data Processing

The data collection process started by querying Chat GPT4 for a list of wikipedia article names about Canadian cities. Then we developed code to fetch and provide a summary of the articles. GPT4 was used to modify the sentences within the summaries to be more specific so that it directly named cities or areas. For example, “This city” was changed to “Toronto”.

Majority of the summaries were split coherently with the NLTK library, but there were some exception sentences that were not grammatically correct or were not formatted properly. These sentences were removed. Cases where the API call returned an error response were also deleted. Additionally, sentences that did not make sense as a stand-alone were removed. This is because the context was established over multiple sentences.

The samples that were thoroughly cleaned were placed into a ‘truth’ dataset. Then we took 50% of the dataset and used an API call to GPT4 to turn them into lies. GPT4 returned 5 unique lies as we found that the options were more creative, the final lie was randomly selected. Our final system prompt to GPT4 to negate sentences was:

“*Suppose someone is talking to you about a Canadian city and their statement is true. Return 5 different sentences that are lies given the input. Try to make the 5 lies subtle but deceptive and not obvious. Separate the output by lines.*

Input: Kingston is a city in Ontario, Canada, on the northeastern end of Lake Ontario.
Output: Kingston is a city in Ontario, Canada, and is land locked.

Input: The boroughs of Montreal often comprise intimate neighbourhoods.
Output: Unfriendly neighbourhoods are common in Montreal's boroughs.

Input: It takes roughly 6 hours to drive from Toronto to Montreal.
Output: I can drive from Toronto to Montreal in 30 minutes.”

An example of a lie generated from GPT4:

“*From Toronto, Montreal is an easy one hour drive away.*”

Finally, 2 samples from the ‘truths’ dataset were randomly combined with a sample from the ‘lie’ dataset to form the final dataset, which contained 432 examples. The training, validation, and test datasets were split by 70%, 20%, and 10% respectively. The final dataset was split almost evenly of each class – 30%, 37%, and 33% for 0, 1, and 2 respectively. In this case, the example’s label is 0 (first sentence is a lie):

“I had to take a ferry to reach Barrie as it's located on an island. Collingwood is a town in Simcoe County, Ontario, Canada. Merritt has dozens of bronzed hand prints of country music stars who have been in Merritt for the annual Merritt Mountain Music Festival displayed throughout town.”

Architecture

An illustration of the model architecture can be seen in Figure 1. The final architecture centres around HuggingFace's GPT2 pre-trained model. This model is connected to a classifier head that predicts the position of the lie (0, 1 or 2). The input data contains 3 sentences, one of which is a lie while the other 2 are truthful statements. The model was trained on 302 data samples, validated on 86 data samples, and tested on 44 samples (70%, 20%, and 10% split respectively). Training was performed over several epochs, but the best performance was at 9 epochs, as seen in Figure 3.

Comparison

Rather than comparing our project with a baseline model, we decided to compare the performance with 37%. This is because our dataset was split into 30%, 37%, and 33% representing each prediction class for the position of the lie (0, 1, 2, respectively).

Quantitative Results

First, we fine-tuned GPT2 using our initial dataset with 70% training split and 30% validation (containing 302 samples).

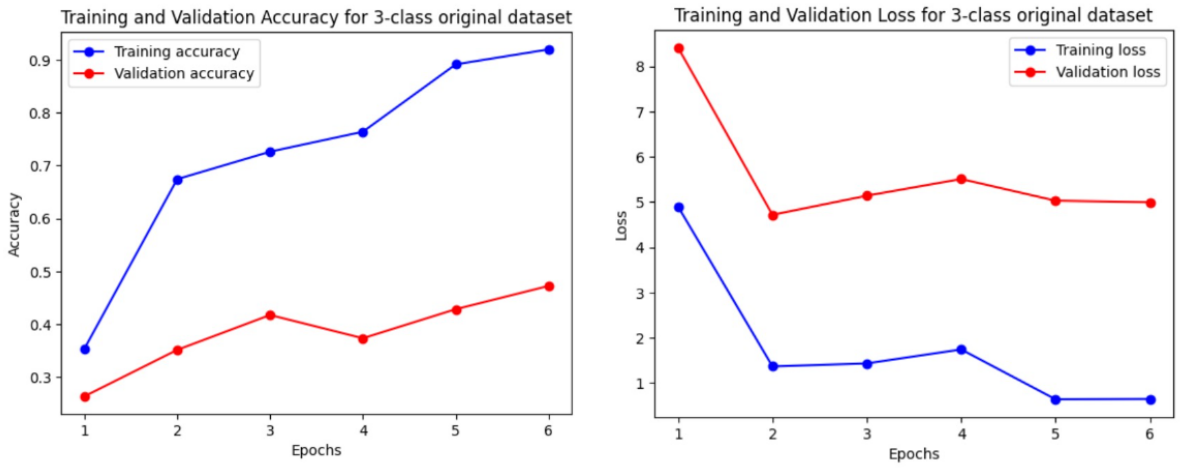

Figure 2. Accuracy and loss curves for initial training and validation.

Figure 2 shows the training accuracy continuously increasing, while the validation accuracy increases and decreases over the epochs. Furthermore, there is a large gap between the training and validation accuracy which implies overfitting (~90% vs. 45% respectively). As discussed in the Data Processing section, we further cleaned and improved our dataset and retrained the model.

Figure 3. Accuracy and loss curves for final training and validation.

Figure 3 shows the gap in accuracies between training and validation have reduced dramatically. The new model achieved ~100%, 81.3%, and 70.5% accuracies for training, validation, and test respectively. Epoch 9 provided the best validation accuracy despite the near perfect training accuracy. These results from our fine-tuned model, especially the 70.5% accuracy for the hold-out test set, indicates the success of our model, compared to the 33% accuracy.

Qualitative Results

The following is an example from the dataset, where the model successfully classifies the third sentence as a lie:

“The metropolitan population in 2022 was 171,608, making Moncton the fastest growing CMA in Canada for the year with a growth rate of 5.3%. Portage la Prairie is a small city in the Central Plains Region of Manitoba, Canada. Victoria, being the smallest, occupies less space than any other city in British Columbia.”

Output: The third sentence is a lie!

The following is an example of the model failing to classify the first sentence as a lie, as it predicts the second sentence as a lie:
“In the early 20th century, Lethbridge was known for its booming oil business. Hamilton has a population of 569,353, and Hamilton's census metropolitan area, which encompasses Burlington and Grimsby, has a population of 785,184. Brantford is known as the "Telephone City" as Brantford's famous resident, Alexander Graham Bell, invented the first telephone at his father's homestead, Melville House, now the Bell Homestead, located on Tutela Heights south of Brantford.”

All the incorrect predictions were collated to a file for further investigation. We suspect that the model was unable to detect lies that were “very confident and formal”. This is an example where it is a false statement but is written very confidently:
“In the early 20th century, Lethbridge was known for its booming oil business.”

The model thought this sentence was false:

“Hamilton has a population of 569,353, and Hamilton's census metropolitan area, which encompasses Burlington and Grimsby, has a population of 785,184.”

To further investigate our hypothesis, we rephrased the previous truthful sentence to be more confident using ChatGPT4:
“Hamilton boasts a population of 569,353, while its census metropolitan area, inclusive of Burlington and Grimsby, is home to 785,184 residents.”

The modified sentence was fed into the model:

“Hamilton boasts a population of 569,353, while its census metropolitan area, inclusive of Burlington and Grimsby, is home to 785,184 residents.”

The model correctly suggests that the first sentence is a lie. This method was applied to several incorrectly labelled examples, resulting in a similar trend. This brings us to the idea that the model does not necessarily pay attention to the factually correctness of sentences, but also to the confidence level of the inputs. This is analogous for humans, if a person lies confidently,others are more likely to believe them rather than if they were nervous. to better analyse the model, the dataset could be expanded with diverse sources for training.

Discussion and Learnings

The ~35% increase in validation accuracy after revising the training dataset shows the significance in the quality of the dataset. In the initial dataset most of the lies were direct opposites of the truths.Thus, the model was not very capable of detecting lies about data that it was not trained on. But as the dataset was modified, and used more creative lies, the model performed better

In the second iteration, we see another pattern in the predictions of the model, which was discussed in the Results section. By analysing some input-output pairs, we suspect that although the model achieves around 70% test accuracy, the reason why it cannot reach higher accuracies is that it could not identify lies that are said “very confidently”.

To tackle the issue about confidence level, we need to increase the training set and use a wider range of lies to help the model understand what confident sentences are indeed lies, and what are truths. Retrospectively, we would rethink how the dataset is generated, and try to use different prompts to negate the sentences (instead of using one prompt for all sentences). For example, for 50% of our dataset, we ask GPT4 to negate the sentences as before, but for the other portion, we tell it to make the lies very convincing, but with different “deceptive” levels. This change would help the model see a wider range of lies and perhaps reach higher accuracies.

A more diverse dataset could help the performance by sourcing data from different subjects. For example, not only using Wikipedia, but also using GPT4 to tell us 1000 true sentences

(about different subjects) and then reading all of them manually (or some of them) to make sure they are actually true.

Individual Contributions

Mei

I developed the code for the data collection and processing pipeline, “data_gen_pipeline.ipynb”. I investigated the dataset and manually cleaned samples as shown in the prefixed “cleaned-*-dataset.csv” datasets. I collected the “cities.csv”, “dataset-cleaned.csv”, “groupDataset.csv”, and datasets prefixed with ‘cleaned-*-dataset.csv’. I also engineered the prompt to negate the original samples through GPT4 and wrote the gradio implementation, “frontend.ipynb”. I wrote the proposal slides, proposal document (except the architecture section), progress report, final presentation slides, and the final report (introduction, illustration, data processing, and architecture).

Mohammadreza

I collected and verified wikipedia articles from GPT4 and provided the initial prompt for negating sentences. I also wrote the code to use GPT4 and wikipedia API, which later Mei developed the full data generation pipeline using them. I was also responsible for training the pre-trained GPT2 model and trained the model for different datasets (with 2 classes and 3 classes), for both initial and final datasets and provided the learning curve figures as provided in the Git repository. The notebook for training the model is provided in “trained.ipynb” file. I also wrote functions to complete the backend word and predict the label given inputs from gradio and finalised the “frontend.ipynb” file. I also tried to investigate what the model is learning and created the “wrongLabeles.csv” file that contains model failures.

References:

[1] V. L. Rubin, Y. Chen, and N. K. Conroy, “Deception detection for news: Three types of fakes,” *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015. doi:10.1002/pra2.2015.145052010083

[2] A. S. Constâncio, D. F. Tsunoda, H. de Silva, J. M. Silveira, and D. R. Carvalho, “Deception detection with machine learning: A systematic review and statistical analysis,” *PLOS ONE*, vol. 18, no. 2, 2023. doi:10.1371/journal.pone.0281323

Permissions

Permissions	Mei Li	Mohammadreza Safavi
Post video	Wait to see video	Wait to see video
Post final report	Yes	Yes
Post source code	Yes	Yes