Using Predictive and Generative AI to Detect Truthfulness in News Articles

Nicholas Shor nshor@ucsd.edu
Henry Luu hluu@ucsd.edu
Irisa Jin irjin@ucsd.edu
Lohit Geddam lgeddam@ucsd.edu

Dr. Ali Arsanjani arsanjani@google.com

Abstract

The issue of fraudulent and misleading news has been an issue for as long as humans can remember, but today it is more rampant than ever with online news and social media giving everyone the chance to stay up to date with news worldwide. With the development of AI, there is the opportunity to either extenuate the increasing amount of misinformative news or to mitigate it. Previous approaches have brushed the surface, with many fact-checking websites leading the way in debunking non-factual statements made by various sources, but the development of generative AI has opened new doors that have allowed for new techniques to flourish. This project will continue to push the needle towards minimizing misinformation and disinformation from media sources through predictive AI and generative AI machine learning methods. By finding new data that is specific to this task, along with developing models addressing many different factors that go into detecting misinformation, this project will be a culmination of the most important factors contributing to mis/disinformation which is unlike other projects previously. The final product will be an interface that is able to intake a news article, process the contents using predictive and generative AI, and return how true it is by providing text describing the truths/falsities, along with a truthfulness score from 1-100.

Code: https://github.com/hluu01/DSC180A_B09_2

1 Introduction

Throughout the internet, there are countless sources of news that people use every day to keep themselves updated on current events. The most prevalent are news websites and social media apps. These platforms have grown consistently as web applications have continued to develop, making them the primary news sources for people worldwide. The problem is that with this change, it has become easier than ever for misinformative news to spread rapidly without being fact-checked. It is extremely difficult to detect wrong from right when the credibility of a piece of news, among other factors, cannot be easily detected. The goal of this project is to consider as many of the factors present and use these to paint a picture of where a news article may be truthful or misinformative. This will allow the user to take this information and make their best judgment on whether the news article is valid or not. Removing misinformation is crucial for protecting public health, democracy, and social cohesion by preventing the spread of false beliefs that can lead to harmful decisions and damage our trust in institutions. By promoting accurate information, we can empower individuals to make informed choices and create a more resilient and cohesive society.

In this project we developed both Generative and Predictive models that were able to produce either truthfulness scores from 1-100, or return a truthfulness label. The labels we used comes from PolitiFact's Truth-o-meter scale which has a range of classifications which are, True, Mostly-true, Half-true, Barely-true, False, and Pants-on-fire. This gives a much more accurate description of the truth value of an article in comparison to a simple binary classification stating whether something is true or false. The truthfulness of a statement is never simply true or false and is instead a blurred scale where many factors must be accounted for. In the end, our model was able to accurately predict the label of a statement from this six way classification over 82% of the time. Also, a majority of the incorrect predictions were only one label off which is very promising. This is still far from where we would like the final product to be to fully implement a project like this, but in a field where much work still must be done it is a very impressive result.

1.1 Literature Review

Previously there has been a large amount of research and work put into predictive tasks using data and well-known machine learning algorithms to try and detect misinformation. A major focus of our project across the past few months was on the Liar Liar dataset. A novel study was conducted by William Yang Wang in which he created this new benchmark dataset to be used for fake news detection. Wang (2017) drew information from PolitiFact.com to create a dataset with over 12,000 data entries of the statement, speaker, context, and truth-label for some factual/non-factual statement (Wang 2017). The development of large datasets like this emphasizes how early on we still are in this process of quality fake news detection methods. This study also goes on and uses this dataset to try and classify similar statements into their correct truth-value category. Unfortunately, the best results Wang (2017) got were from Hybrid CNN models that used all the data present but still only got 27.4% accuracy (Wang 2017). Beyond this study, there have been adjust-

ments to the Liar Liar dataset which have added context to the statements provided. The Liar Liar Plus dataset is an extension that has a justification added to the dataset for each entry, providing more context on the matter. This dataset created by Alhindi, Petridis and Muresan (2018) took the Liar Liar dataset and added the justification for the claim that was provided in the fact-checking article associated with the claim. This article explained why the statement some person made was true, false, or somewhere in between. To convert the textual data to usable features for prediction, the team ended up using unigram representations (Alhindi, Petridis and Muresan 2018). The deep learning model of choice was a Bi-directional Long Short-term Memory architecture that has previously proven to be successful with NLP-related tasks (Alhindi, Petridis and Muresan 2018). By using the extracted justification along with additional metadata, the team achieved 37% accuracy on the test set, a significant improvement from Wang's 27% accuracy. This is a huge increase and emphasizes the importance in providing additional context during model building. Combining these ML predictions with Generative AI is something that is yet to be done in this field since the usage of Large Language Models (LLMs) for developers has taken off so suddenly. This is the space we intended to fill with our project.

1.2 Data

The data we used was a combination of readily available data from previous studies, as well as data we were able to scrape and collect especially for usage on our own project. The dataset section is split up into predictive and generative AI data as we used specific datasets for these different approaches.

	Json_File_ID	Truth_Label	Statement	Subject	Speaker	Speakers_Job	State	Party	Context_Venue_Location	Justification
0	2635.json	false	says the annies list political group supports ...	abortion	dwayne-bohac	State representative	Texas	republican	a mailer	That's a premise that he fails to back up. Ann...
1	10540.json	half-true	when did the decline of coal start it started ...	energy,history,job-accomplishments	scott-surovell	State delegate	Virginia	democrat	a floor speech.	"Surovell said the decline of coal ""started w...
2	324.json	mostly-true	hillary clinton agrees with john mccain by vot...	foreign-policy	barack-obama	President	Illinois	democrat	Denver	"Obama said he would have voted against the am...
3	1123.json	false	health care reform legislation is likely to ma...	health-care	blog-posting	NaN	NaN	none	a news release	"The release may have a point that Mikulskis c...
4	9028.json	half-true	the economic turnaround started at the end of ...	economy,jobs	charlie-crist	NaN	Florida	democrat	an interview on CNN	"Crist said that the economic ""turnaround sta...

Figure 1: LiarLiarPlus Dataframe

1.2.1 Predictive AI Data

The first dataset we used was the Liar Liar Plus dataset (Alhindi, Petridis and Muresan 2018). This dataset had a statement made by an individual or post online, the associated truth label, subject, speaker, speaker's job, political party, location of the statement, and additional justification. It also had the counts of previous truth ratings of individuals. This was the baseline dataset we were working with and was used to train our autoML, and BERT full-text embedding model (Figure 1).

	media	when/where	content	label	speaker	documented_time	percentages	check_nums	summaries	article
0	Instagram posts	stated on October 28, 2023 in a screenshot sha...	"Haaretz investigation reveals discrepancies i...	FALSE	Madison Czopek	October 31, 2023	['0%' '0%' '2%' '7%' '67%' '21%']	[5 3 16 54 473 152]	['Haaretz, an Israeli newspaper, said on X tha...	A viral Oct. 28 social media post claimed that...
1	Scott Walker	stated on May 30, 2023 in Interview:	"Wisconsin has historically ... and I think larg...	barely-true	Laura Schulte	October 31, 2023	['12%' '21%' '18%' '19%' '21%' '5%']	[26 45 39 41 44 11]	['Although Wisconsin has voted for more Democr...	In 2016, Wisconsin helped to swing the preside...
2	Instagram posts	stated on October 27, 2023 in a post:	"The airport in Salzburg, Austria, has a count...	FALSE	Ciara O'Rourke	October 30, 2023	['0%' '0%' '2%' '7%' '67%' '21%']	[5 3 16 54 473 152]	[]	A social media post poised to encourage people...
3	Viral image	stated on October 27, 2023 in an Instagram post:	Video shows Palestinians pretending to be corp...	FALSE	Ciara O'Rourke	October 30, 2023	['0%' '1%' '2%' '4%' '62%' '28%']	[4 13 35 53 745 336]	['This video is 10 years old and shows student...	The Gaza Health Ministry has said the Palestin...
4	Facebook posts	stated on September 25, 2023 in a Facebook post:	The life span of a wind tower generator lasts ...	FALSE	Loreben Tuquero	October 30, 2023	['0%' '1%' '4%' '9%' '59%' '23%']	[24 50 108 247 1519 594]	['A study by energy industry experts showed th...	Let's clear the air. Do wind turbine component...

Figure 2: Sean Jiangs Scraped PolitiFact Dataset

We also used a dataset that was scraped by our classmate, Jiang (2023), that contained all the truth ratings of all PolitiFact truthfulness speakers in a column called check_nums. The order of truthfulness was true, mostly true, half true, barely true, false, and pants on fire. This was very useful in the training of our n-gram, autoML, sentiment, quality of writing, and sensationalism models as it provided additional information that the Liar Liar Plus dataset didn't have (Figure 2).

The last dataset we used for our predictive models was a Clickbait dataset that we got from Kaggle. This dataset provided the headline of an article and then either a 1 if that articles title was clickbait, or a 0 if it wasn't. This was used to train our clickbait model.

Table 1: Kaggle Clickbait Data (32000 rows)

headline	clickbait
Should I Get Bings...	1
Which TV Female Friend Group Do You...	1

1.2.2 Generative AI Data

We used five datasets of fact-checked and reliably sourced news article information from various providers to use in our Generative AI process. The first of those providers was

Politifact.com, a former Pulitzer Prize winner. From PolitiFact, we web-scraped data to get two datasets of information. The first consisted of just the title and text from news articles PolitiFact constructed (Table 2). The second consisted of statements made by individuals and the corresponding truth value of those statements that PolitiFact has provided after expert fact-checking. Along with this statement and rating, we also scraped the claimer of the statement and the text that explains why this statement received the rating it did (Table 3). This data assisted in providing additional context and fact-checked information for our Generative AI model to reference before creating its output.

Table 2: Webscraped PolitiFact Articles (3131 rows)

Title	Text
The deficit has fallen under Joe Bid...	President Joe Biden has often touted...
The Biden-versus-Trump economy: Who ...	Get ready: In the 2024 presidential ...

Table 3: Webscraped PolitiFact Statements (8811 rows)

Claimer	Statement	Truth_value	Text
Brian Kemp	The ”left” is blatantly ...	false	With Republican presidential...
Bloggers	Greta Thunberg “goin...	false	The headline of a story on a...

The next set of datasets we used came from FactCheck.org, another highly touted fact-checking organization that has received a handful of awards for their service. We also web-scraped information from their website to receive their fact-checking articles' text, title, date, and additional list data that some articles provided (Table 4). This provided more context and information for our Gen AI model to leverage. The next dataset we used came from SciCheck.org, which is just a branch of FactCheck.org that focuses exclusively on false and misleading scientific claims that are made by partisans to influence public policy. From here we scraped the fact-checking articles text, title, and date (Table 5). There was no list data available in these articles.

Table 4: FactCheck.org Article Data (2051 rows)

Text	List_data	Title_and_Date
The Supreme Court ruled that states...		Role of Illinois Circuit Court...
People vaccinated with an authorized...		Blood Donations from COVID-19...

The final dataset we used in our Generative process was a dataset we obtained from web-scraping Science.Feedback.org. Another highly-rated fact checking organization that focuses on confirming or denying scientific claims that are made in the community. We used this data to provide our Generative model with more scientific fact-checked claims to reference. The data we scraped contained a claim made and the supporting label for that claim (Table 6). These five datasets contained all the data we provided the Generative

Table 5: SciCheck.org Article Data (580 rows)

Text	Title_and_Date
The Supreme Court ruled that states...	Role of Illinois Circuit Court Judge...
People vaccinated with an authorized...	Blood Donations from COVID-19 Vaccin...

AI model to gather additional context and information from so that it could appropriately score a piece of texts truthfulness.

Table 6: Science.Feedback.Org Statement Data (580 rows)

claim	label
Chemical found in Cheerios, Quaker Oats may cause ...	Unsupported
Researchers have “already perfected the ability to...	Misleading

2 Methods

The biggest difference between previous approaches tackling fake news and current approaches is the recent development of LLMs for developers to leverage for their own projects. While the Generative AI model approach does have similarities to the Predictive model approach, some differences will be discussed here.

2.1 Predictive Models

2.1.1 Google AutoML

Our Predictive Models consisted of various different types of models. We started with Logistic Regression to setup a baseline model, and later explored training models using Multinomial Naive Bayes, a model known for doing well with sentiment analysis. Ultimately, we decided to once again use the Google Cloud console in order to train the best possible model possible. We uploaded two datasets, the LiarLiarPlus dataset and the Scraped PolitiFact dataset from Sean Jiang. We then trained both models using Google’s AutoML feature, specifying that we were doing multi-class tabular classification on the truthfulness of the article. Google’s AutoML then ingested and prepared the data, before automating the model selection and hyper-parameter tuning. While AutoML is very costly and inefficient in the long run due to scalability concerns, it provided us with a very high floor to build from. The results from Google’s AutoML models and how they were used will be discussed in the Results section.

2.1.2 Readability Metric

To evaluate the readability of the inputted text we used a common metric called the Flesch-Kincaid Grade Level score which takes the text and evaluates how easy it is for the text to be understood in English by providing the comprehension level associated with that text. The readibility of a text is very important because it has to do with the credibility of the author. A good author should know that the average American is high school educated, and that they should aim for a reading grade level between 8-12 based on good journalism practice. Ignoring this hinders reliability unless they work for a specialized news source that caters towards a more educated audience. If this is true though, the other scoring metrics should make up for the reduction from this metric. The python package we used to implement this metric can be found here.

2.1.3 Clickbait

To evaluate the clickbait levels of a news article we trained a clickbait detection model on a clickbait dataset found on Kaggle. We first preprocessed the text and then used a CountVectorizer() function to convert the text to data usable for modeling. The best performing algorithm was a Support Vector Classifier (SVC) and we used this to detect whether an article was clickbait or not. At evaluation time, we used the confidence score that a certain piece of text was clickbait to convert this classification into a multi-class prediction from the following labels, [“pants-fire”, “false”, “barely-true”, “half-true”, “mostly-true”, “true”]. This was then combined with our other predictive model evaluations and then further included in our overall evaluation.

2.1.4 Context Veracity

To delve into contextual veracity, we utilized BERT embeddings extracted from the Liar Plus dataset employing the pre-trained ’bert base-uncased’ model available in the transformers library. Leveraging the GPU provided by the UCSD Data Science Machine Learning Platform (DSMLP), we extracted these embeddings. Subsequently, we employed logistic regression, decision tree, and random forest classifiers to analyze the embeddings, with the random forest classifier demonstrating superior performance among the three.

2.1.5 Quality of Writing, Sensationalism and Sentiment

To assess the quality of writing, we used the Type Token Ratio (TTR), which is determined by dividing the number of different words by the total number of words. Sensationalism was rated based on the number of adjectives over the total number of words. Finally, we added a sentiment score ranging from -1 to 1. Based on these scores, a label was predicted and provided back to the user. Additionally, this label was then converted to a number from 1 to 100 to be combined in our overall rating evaluation.

For further sentiment analysis, we used TF-IDF vectors. However, use to TF-IDF's inability to capture context, we decided to supplement it with N-Gram Analysis, where text was broken up into groups of N words. We used these features to train both Multinomial Naive Bayes and Logistic Regression models.

2.2 Generative Model

2.2.1 Generative AI Model Instantiation

The first step in our Generative AI process was to select an LLM we could use for our usage. We chose to work with Google's Gemini Pro model which is accessible through the Google Cloud Vertex AI console tool. By working through the Google Cloud console we were able to freely configure our LLM and customize it for our specific task. The first step was to apply the appropriate configurations for our model. We specified the max number of output tokens to be 2048 which is approximately 1,500 words, however, all of our responses are much shorter than this and usually don't exceed 200 words. We also had to set up the safety configurations so that our model does not potentially output harmful or dangerous data to users. To do this we used the Perspective API by Jigsaw which uses machine learning methods to identify toxicity in text online. This helped us ensure that all output text included no toxic or harmful information. It is also critical to note that in Geminis model instantiation, there is a parameter called "temperature" that ranges from 0-1. This controls how much creativity and diversity the model puts into its responses. For our specific task, we chose to use a temperature of 0. This means that the model will provide much more factual and predictable outputs, something that is necessary for our model to run consistently.

2.2.2 Retrieval Augmented Generation

Retrieval Augmented Generation, or RAG, is a technique that has become very popular in the Generative AI world. RAG uses a vector database where large amounts of information are stored as embeddings. These embeddings can be any set of documents that you would like, but their main purpose is to supply additional context for the LLM to refer to in its question-answering process. To appropriately implement RAG you must first chunk your documents into smaller pieces, usually single sentences or short paragraphs. You then need to embed these documents using a vector embedding so they can be stored in your vector database of choice. This vector database can then be prompted to search for certain pieces of information and then retrieve this information using a semantic search algorithm known as an Approximate Nearest Neighbors Search (ANN). This is a critical step in the Generative AI process because it can reduce the amount of hallucination the Gen AI models are prone to do on occasion by increasing the amount of context the model has about the specific topic (Arsanjani, 2023). The catch of course is that if your vector database does not have information related to the prompt, then the vector database will not be able to provide relevant information which could harm your model. This is a common issue in the use of

RAG implementations and a solution that has been developed recently is the use of RAG re-ranking systems. What the re-ranker does is sifts through the responses provided by the vector database and re-ranks them based on relevance to the prompt (Solanki 2023). This ensures we only get the most relevant documents for our model. We used FlagEmbeddings free re-ranking package for our implementation. For our Vector Database implementation we decided to work with the Chroma Vector Database implementation using its embeddings. The data we uploaded to the database was fact-checked statements and articles made by Politifact, FactCheck.org, Science.Feedback.org, and SciCheck.org.

2.2.3 Our RAG implementation

In our implementation, we decided to use two separate vector databases, one for contextual information obtained from reliably sourced news articles, and the other for fact-checked statements that had the appropriate truth label that we put in the entries metadata. This allowed us to supply the model with contextual information on the topic as well as fact-checked statements so it could make the most informed decision in its ratings and explanations. To ensure that we only get the most relevant information, we chunked the evaluated news article at runtime so that each evaluated statement is the same size as the chunks loaded into the vector database. At runtime, when the model is processing an individual chunk from the article, RAG searches these two vector databases for the most relevant information to the chunk provided. We then fed this retrieved information into a RAG re-ranking system called FlagReranker that is created by FlagEmbedding. This allowed us to feed our RAG output into a re-ranking mechanism that compares the context/fact-checks we received to our inputted chunk and returns the top-k most relevant pieces of information. We then take these top-k pieces and feed them into the model so that we are only supplying the most relevant and useful information on the statement being evaluated.

2.2.4 Prompt Engineering

Another critical part of our Generative AI model development process was prompt engineering. To reach our final prompt, many iterative steps were taken in order to get there. Thanks to our incredible mentor Dr. Ali Arsanjani, we were given lots of insight and advice on the best approaches to take. The first step was to be very specific in our prompting. This means we need to craft a prompt that is very clear and detailed so that the LLM knows exactly what we want from it. To tackle this we made sure we told it specifically that we wanted the statement provided to be rated on truthfulness form 1-100, that we wanted it to follow a specific output format, and that we wanted it to provide a short explanation behind the rating it provided as well. We also partitioned the different parts of our prompt clearly, meaning that we included all our contextual information in one section and the question we wanted to be answered in another section. The next step was to implement some form of Chain-of-Thought reasoning, which for us just entailed telling the model to think about this problem step-by-step and provide an explanation to ensure it provided a logical and reliable response (Arsanjani 2023). The next step was to format the additional information

we were providing it through RAG in the easiest way possible for the LLM to understand. This entailed providing specific meta-data about the truth value of the retrieved statement, along with other details such as the title and date of the chunked article context. The final step we took was to provide the model with the previous and following chunk of information from the news article being examined to ensure that the model had all the context necessary to make its evaluation. Oftentimes, sentences in isolation can be confusing due to the use of pronouns or other non-specific phrases. Adding the single previous and following chunk should supply sufficient context so that the model knows exactly where, what, and who the statement is discussing. Following all these steps brought us to our refined final model prompt.

2.3 Final Deployment

To successfully deploy our model we needed to combine all of these methods mentioned previously and also find a location to host our model that had a user-friendly interface. The first step was chunking down the news article into smaller pieces so that when used to search in the vector database it was able to find relevant results. We then fed each chunk of the news article through the Gemini Pro model and told it to rate the following statements' truthfulness based on the context we provided from the vector database, and also any knowledge that it already had. Additionally, we fed in the news articles' previous chunk, and following chunk to give the model any contextual information that may have been missing from the single chunk it is rating. This gives the model as much context as possible to rate each chunk in the article accurately. The output for each chunk is a truthfulness score from 0-100, along with a short explanation of why that score was given. All of these scores and explanations, along with the scores provided by our predictive models, are then parsed and processed so that we can provide the average of the scores and sufficient explanation for the user. It is important to note that our final model was weighted so that 80% of the score comes from our Generative AI scoring and the other 20% is made up of our Predictive AI scoring. Once the model was functioning in its entirety, the next step was to use Gradio, a platform designed to make it easy to share machine learning apps in a friendly web interface. We used Gradio to set up a simple website where a user can take a news article and corresponding headline, input it into the Generative AI interface, and then receive an evaluation of how true the inputted information is. At the same time, the predictive scores are being calculated for the content and then provided to the user. Finally, a short overall synopsis of the articles truthfulness is provided to the user. There are also a few pre-loaded examples of up-to-date news articles from some very popular news providers that have been evaluated by our model on the website for users.

3 Results

3.1 Predictive Model Results

Table 7: Results from Different Models on Politifact data

Model	Precision	Recall	F1-score	Support
Full Text BERT Embedding Predictions				
pants-fire	1.00	0.01	0.02	84
false	0.32	0.34	0.33	297
barely-true	0.24	0.04	0.08	200
half-true	0.23	0.41	0.30	273
mostly-true	0.28	0.45	0.34	268
true	0.34	0.12	0.18	227
Overall Accuracy	0.33	0.27	0.24	1349
Full Text N-Gram Analysis				
pants-fire	0.89	0.52	0.66	673
false	0.60	0.85	0.70	1438
barely-true	0.78	0.57	0.66	805
half-true	0.72	0.78	0.74	830
mostly-true	0.60	0.61	0.60	757
true	0.64	0.41	0.50	555
Overall Accuracy	0.69	0.67	0.66	5058
Full Text Google AutoML				
pants-fire	0.95	0.89	0.92	279
false	0.91	0.90	0.91	737
barely-true	0.91	0.88	0.90	375
half-true	0.95	0.93	0.94	397
mostly-true	0.92	0.90	0.91	357
true	0.85	0.83	0.84	262
Overall Accuracy	0.92	0.89	0.90	2407

3.1.1 Clickbait Model

Our clickbait model was different from our other predictors in the sense that it made binary predictions of 0 if the title wasn't clickbait and 1 if the title was clickbait. This is why we saw an accuracy score of 95.67%, which was much higher than the other predictive models performances.

3.1.2 Multi-Class Model Results

Our random forest classifier trained on full-text BERT embeddings achieved a classification accuracy of 33% on the test dataset, which comprised of 1349 instances. The best multi-class logistic regression model was one that was trained using TF-IDF vectors supplemented with N-Grams. It achieved an overall accuracy of 66.7%.

3.1.3 Google Vertex AI AutoML

Google Vertex AI's AutoML was used to train two separate multi-class tabular classification models. The first model was trained on the Liar Liar Plus dataset and was able to achieve an overall accuracy of 50.1% and precision of 81.5% on the test data. The features that were given the most importance were the credibility scores, which took into account the truthfulness of that author's past articles. The model did well in identifying articles that were either false or had little truth to it but struggled in identifying articles that were pant-fire, false, and true. The second model was trained on Jiang's dataset that included the entire article as a feature. This AutoML model was trained solely using the article and achieved an impressive accuracy of 95.4% based off of the area under the precision-recall curve and a precision of 91.6% on the test data. This model excelled across all labels, but showed weakness in identifying the "true" label correctly as it scored an accuracy of 88.7% for that label.

3.2 Generative Model Results

Our experiment investigated the effectiveness of Generative and Predictive AI methods in detecting truthfulness and deceptiveness in news articles. In this section, we will be discussing how we tested our Generative AI model. Since we are working with a Generative LLM here, the process of testing the model is a bit different than traditional Machine Learning methods. What we did was tested the model by varying our prompt, and the information/context that we provided as input. This allowed us to see an improvement in performance as we made changes. To test performance we evaluated our various approaches on a randomly selected set of 500 entries from the Liar Liar dataset that were not included in our vector databases. We did this by asking the model to process one of the statements from the Liar Liar dataset and then return one of the six possible labels (pants-fire, false, barely-true, half-true, mostly-true, true). We then compared the predicted labels to the actual labels we had in the dataset to test accuracy.

The first method we tested was simply feeding the statements directly into the Gemini Pro LLM. This served as our baseline, achieving an accuracy of 21%. The next step was to implement well-known Generative AI techniques to improve this performance. By simply adding Retrieval-Augmented Generation (RAG) alone, we saw a significant increase in accuracy to 59.8%. This provided the model with additional contextual information on the statement and also similar fact-checked statements on the topic. Further improvement was achieved through additional Prompt Engineering as mentioned in the Methods section. This

provided the model with more guidance towards truth detection and resulted in an accuracy of 71%. Our final approach using just Generative methods was the most successful, combining RAG, our re-ranking mechanism, and prompt engineering, reaching an accuracy of 78.2%.

To further improve our model's overall performance, we also explored combining our Generative AI model with scores from our separate Predictive Models. During implementation, we saw some interesting results. When our Predictive and Generative scores were weighted equally, the accuracy dropped to 34.6%. However, by weighting the Generative AI model's score more heavily (80/20) we saw the highest overall accuracy of 82.4% (Table 8). These results suggest that Generative AI methods, particularly when combined with prompt engineering and re-ranking, are promising tools for detecting truthfulness in news articles. Furthermore, incorporating scores from external models can offer additional improvements, but optimal performance relies on careful weighting of these scores and thorough testing.

Table 8: Overall Model Performance (Tested on 500 random Liar Liar dataset entries)

Model Description	Score (%)
Baseline (Feeding straight into Gemini Pro)	21
RAG	59.8
RAG + Prompt Engineering	71
RAG + Re-ranking + Prompt Engineering	78.2
RAG + Re-ranking + Prompt Engineering + Pred Modeling Weighted 50/50	34.6
RAG + Re-ranking + Prompt Engineering + Pred Modeling Weighted 80/20	82.4

4 Discussion

4.1 Interpretation of the Results & Other Works

As we look back to the Liar Liar Dataset study and additionally the Liar Liar Plus study, we see that they achieved a 27% and 37% accuracy score respectively. It is clear that this is far too low to serve as a reliable truthfulness detector. What we found from our predictive results is that while we were able to generate higher accuracy scores than these studies, it was extremely hard to generalize our results to different news domains. Having limited training data meant our models performed very well on specific, familiar types of data and not as well on new and foreign ones. With our collection of new data though we have provided new and more in-depth data for others to utilize in their projects to hopefully improve performance across these news domains. We also have achieved some notably high accuracy scores of 66% in our N-gram and 90% in our Google AutoML models.

On the Generative AI side, we saw some very promising results as well. The most difficult part of this aspect of the project was the lack of research previously done with

Generative AI and fact-checking. Since LLMs and their open usage with developers has only recently exploded, there is still a huge gap to be filled in this field. This meant we had to pave our own way in this field and due to this, we met very many roadblocks. With that being said, obtaining an accuracy score of 82.4% for statement label prediction is quite impressive. With this being a widely unexplored field, the room for improvement is immense, and having a solid starting ground to work from is very promising. As Generative AI methods and workflows continue to improve, along with an increase in quality and up-to-date fact-checked information, models that attempt to tackle disinformation will only continue to improve. We hope the data we've collected can also be used in the development of other projects and spur motivation in others to attack this worldwide issue as well.

4.2 Impact & Limitations

Accurate information is paramount for making well-informed decisions. This project contributes a valuable tool in combating misinformation and disinformation online by providing a more efficient and streamlined approach to verifying online sources. Since readers can come from various educational backgrounds, this model offers an improvement in media and digital literacy, which would lead to increased technological accessibility and trust in journalism.

This project was limited by multiple different factors, both on the Predictive AI and the Generative AI sides of things. A limiting factor for the Predictive AI was the amount of data that could be processed. Data for training these sorts of models is not readily available and needs to be scraped and processed. These large amounts of data then need to be fed into a model for training. Both of these processes are extremely time-consuming and in our project, we were limited in terms of computing power. Having access to more computing power would have allowed us to process and train more data, which would have led to more comprehensive models. Additionally, we could have trained our models on more domains. Currently, the bulk of the articles used to train our models have been either political or from the healthcare industry. Our predictive AI models perform poorly on articles from other domains, so one of the next steps for this project would be to train models on more data from other domains. This of course would be easier with more computing power.

However, Predictive AI is still limited by one major factor: truthfulness. Regardless of the domains and amounts of data used to train the predictive models, these models are incapable of verifying statements. They can look for patterns in text and run various types of sentiment analysis, but these predictive models have no way of knowing about new, developing stories. This is where the importance of Generative AI comes in. Generative AI allows us to augment our Predictive AI models with fact-checking capabilities. While is important to keep in mind that the truth is not always definitive, these fact-checking capabilities vastly improve Predictive AI models. As seen earlier, our best model combines both Generative AI and Predictive AI in order to produce a complete model. We believe that this is the future for this sort of work and hope this project serves as a baseline.

5 Conclusion

In conclusion, our project represents a significant step forward in the development of AI-driven solutions for detecting truthfulness in news articles. Through the integration of predictive and generative AI techniques, coupled with innovative prompt engineering strategies, we have achieved promising results in assessing the credibility of news content. The mixture of our various predictive models and Generative AI combined with RAG yeiled an accuracy of 82.4%. While there are still limitations to address and areas for improvement, our project lays a solid foundation for future research in this critical area. Ultimately, by providing users with access to tools that evaluate the credibility of news content, we enable individuals to take an active role in verifying information and making informed decisions about what they consume and share.

References

Alhindi, Tariq, Savvas Petridis, and Smaranda Muresan. 2018. "Where is your evidence: Improving fact-checking by justification modeling." In *Proceedings of the first workshop on fact extraction and verification (FEVER)*. [Link]

Arsanjani, Dr. Ali. 2023. "Generative AI Lifecycle Patterns." [Link]

Jiang, Zhixing. 2023. "politifact data combined."

Solanki, Shivam. 2023. "Improving RAG (Retrieval-Augmented Generation) Answer Quality with Re-Ranker." [Link]

Wang, William Yang. 2017. ""liar, liar pants on fire": A new benchmark dataset for fake news detection." *arXiv preprint arXiv:1705.00648*. [Link]