Evidence-Augmented LLMs For Misinformation Detection

Oren Ciolli
ociolli@ucsd.edu

Zhixing Jiang
zhj003@ucsd.edu

Jun Linwu
julinwu@ucsd.edu

Yiling Cao
yic055@ucsd.edu

Dr. Ali Arsanjani
dr.arsanjani@gmail.com

1 Abstract

In an era marked by the rapid dissemination of misinformation and disinformation, the need for effective fact-checking methodologies has never been more pressing. Traditional fact-checking approaches, while valuable, suffer from limitations such as time-consuming manual processes, and previous automated fact-checkers haven't been practical due to their lack of justification and context regarding their classifications. This paper proposes a novel approach to fact-checking which leverages Large Language Models (LLMs) within a multi-model pipeline to provide both veracity labels and informative explanations for claims. Building upon previous research, we integrate various predictive AI models and the retrieval of external evidence from reliable sources to enhance the accuracy of our predictions.

2 Introduction

It was once said that "a lie gets halfway around the world before the truth has a chance to get its pants on." This is more true today than ever, as misinformation and disinformation are becoming increasingly pervasive and destructive, and have permanently altered the political landscape. Fact check sites such as PolitiFact and Snopes have led the charge in combating this misinformation thus far, but they unfortunately have their own shortcomings. Due to the fact that claims must be manually labeled and explanations for those labels must be typed up and reviewed by real people, the process of generating fact-checks is highly time consuming. This lends credence to our motivating quote, as it's much easier for bad actors (or misinformed internet users) to quickly generate and spread misinformation and disinformation than it is for fact-checkers to correct it.

Automated fact-checkers have the potential to allevaite this issue, but the major shortcoming of previous automated fact-checking implementations is that they don't provide any context or justification for their classification, and the only thing that users have access to is a single true/false label. To address this shortcoming, we aim to utilize a LLM to translate our predicted labels or values of the factors into informative, human-readable explanations which inform the user exactly where the claim is potentially biased, misleading, or incorrect.

Additionally, many earlier studies primarily rely exclusively on textual cues and linguistic features for misinformation detection, which often fall short of accurately determining intentionally misleading content and providing information on other factors which are important for users to understand the nature of the claims being made in a wider political landscape. We incorporate more external information in our predictive modeling through the retrieval of external evidence from trustworthy sources such as Wikipedia and various reliable news sources (such as CNN), which allow us to more appropriately contextualize claims and create informative, accurate responses.

2.1 Related Work

Early datasets for this task were limited, with Vlachos and Riedel (2014) compiling just a few hundred labeled examples. Wang (2017) made progress with the LIAR dataset containing over 12,000 labeled short statements, collected from PolitiFact.com. These claims are labeled according to PolitFact's "truth-o-meter", allowing further predictions of 6 different labels for varying degrees of truthfulness. However, LIAR doesn't provide any background context to support the labels. Alhindi, Petridis and Muresan (2018) propose an extended version of LIAR that includes evidence sentences extracted from verification reports. This benchmark dataset enables the modeling of justification evidence for improved performance.

Numerous studies have addressed the critical task of fake news detection using a variety of methodologies, each with its own strengths and limitations. Abdullah-All-Tanvir et al. (2019) and Choudhary and Arora (2021) tackle the challenge of discerning misinformation by respectively leveraging deep learning techniques and linguistic features. While deep learning models excel at identifying patterns within textual and visual content, linguistic analysis provides valuable insights into the semantic nuances of language. However, these approaches may struggle to provide transparent explanations for their classification decisions, limiting their interpretability. Sarrouti et al. (2021) also presented a deep learning approach on their own created dataset, HEALTHVER, which allows others to study the validity of real-world claims compared to ground-truth evidence based on the relationship between two statements (e.g. Refute/Support). One thing that might be critical is the process of manually labeling the relationship between two statements, which may also involve inaccuracy and bias when detecting fake news.

Popat et al. (2018) explores the integration of external evidence from credible sources to enhance the reliability of fake news detection models. While this approach strengthens the veracity of predictions, it may still fall short in providing comprehensive explanations for classification decisions. Additionally, B, Kumar and Chacko (2022) discusses the use of explainable artificial intelligence (XAI) techniques for fake news detection, particularly on social media platforms. However, it framed the problem as a binary classification task, which may not fully capture the complexities of political news.

Our project amalgamates these diverse methodologies to address the limitations inherent in existing approaches. By focusing on multi-class classification, our project provides more nuanced explanations by justifying predictions for complex cases that fall into grey areas between strictly true or false news. By integrating external evidence from reputable sources and employing generative AI models, we aim to provide veracity labels along with informative explanations for claims. This approach promises to enhance the accuracy and interpretability of fake news detection.

2.2 Data

2.2.1 LIAR-Plus

The LIAR-Plus dataset, presented in Alhindi, Petridis and Muresan (2018), represents an expanded dataset derived from the foundational LIAR dataset, encompassing a comprehensive array of 16 distinct features. Among these features are notable elements such as the speaker's name, their previous fact-checks, their party affiliation, the claim's subject matter, justification for the label, and others. Comprising a training set with 10,242 instances, a validation set containing 1,284 instances, and a test set comprising 1,267 instances, the LIAR-Plus dataset presents a well-structured resource for training, validating, and testing. Comparing to the original LIAR dataset, the LIAR Plus dataset contains more features which serves as a plus in building model. In the project, this dataset serves as the foundation for training multiple models for the factors, and is used in part to evaluate the performance of the generative AI pipeline.

2.2.2 Data From Politifact.com

We scraped data from PolitiFact.com, which contains a short statement and a truth-o-meter rating, which labels the truthfulness of the information. The truth-o-meter has 6 classes, which are listed in descending order of veracity below:

1. TRUE – The statement is accurate and there's nothing significant missing.
2. MOSTLY TRUE – The statement is accurate but needs clarification or additional information.
3. HALF TRUE – The statement is partially accurate but leaves out important details or takes things out of context.
4. MOSTLY FALSE – The statement contains an element of truth but ignores critical facts that would give a different impression.
5. FALSE – The statement is not accurate.
6. PANTS ON FIRE – The statement is not accurate and makes a ridiculous claim.

The collected dataset contains 25,615 elements and ten attributes including the speaker's name, the speaker's past fact-checks, the date of the claim, a summary of relevant context, and other features. This dataset is used in building our predictive models for credibility, spam, source reliability, etc. We also use this dataset to evaluate the full pipeline, in conjunction with LIAR-plus.

2.2.3 Scraped News Data for Evidence Database

Using a Selenium webdriver, we also scraped a selection of news articles from NPR's archive (articles from between early-2022 to early-2023) and CNN (late 2023), which we divided into one to two sentence chunks and used in our vector database.

2.2.4 Clickbait Data From Kaggle.com

The clickbait dataset encompasses headlines from diverse news sites, including WikiNews, New York Times, The Guardian, The Hindu, and etc. This dataset consists of 32,000 rows with two columns: the first containing headlines and the second featuring numerical labels indicating clickbait status (1 for clickbait, 0 for non-clickbait). The dataset is evenly split, with 50% classified as clickbait and the remaining 50% as non-clickbait. We employ this dataset to train our clickbait predictive model.

2.2.5 POLUSA Dataset

Gebhard and Hamborg (2020) provide a large dataset of news articles which we used for evidence retrieval. This dataset contains approximately 0.9M articles covering political topics published between Jan. 2017 and Aug. 2019 by 18 news outlets. We filtered out unreliable news sources (such as Breitbart) which were present in the dataset, and experimented with this dataset as one of our sources of evidence.

2.2.6 Entity-Manipulated Text Dataset

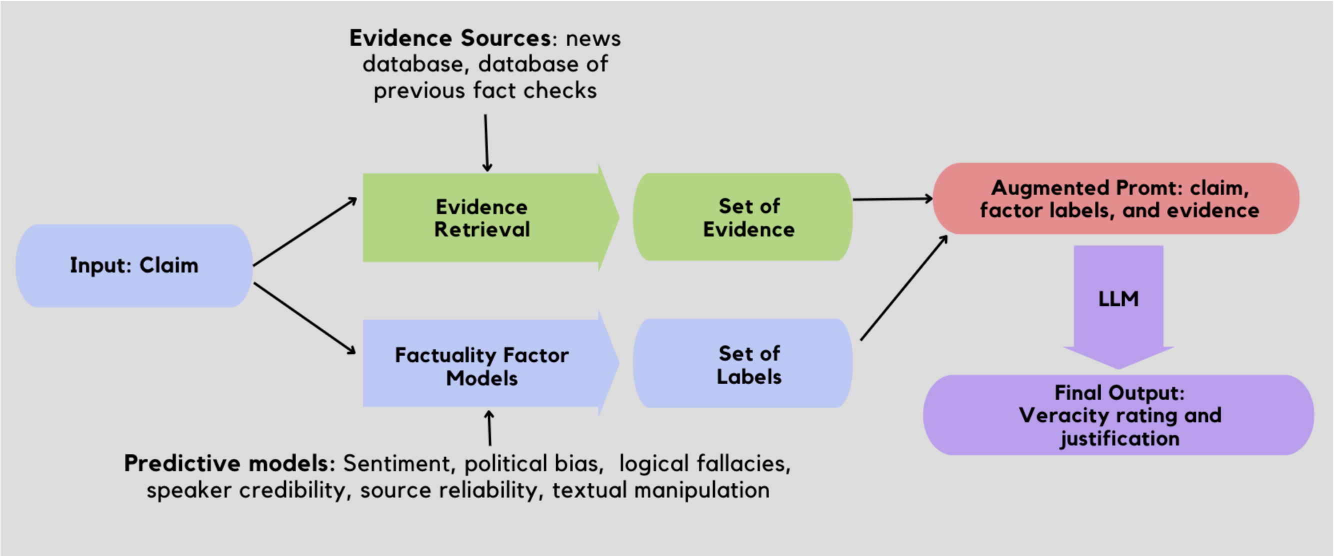
Jawahar, Abdul-Mageed and Lakshmanan (2022) provides a large dataset to allow us predict text manipulation within the context consisting of training, validating, and testing subsets. Text and label are the two main features that we apply in training our style (text manipulation) model.

3 Methods

In this project, we attempt to create a system which streamlines the fact-checking process by emulating the activity of a human fact checker: evaluating claims by searching for external evidence, analyzing the claims within the current political landscape, and writing a response to summarize its findings and correct and contextualize the claims.

Our project involves a three-stage pipeline to evaluate the veracity of the claims, using predictive models and a generative model augmented by an evidence retrieval module. The outputs of the predictive models and the evidence retrieved by our retriever are used to augment the prompt for the LLM, which then uses this information to evaluate the veracity of the claim and generate a textual response correcting any misinformation. With this approach, we're able to leverage the reasoning and summarization ability of large language models in conjunction with lightweight, task-specific models which provide the LLM with relevant context.

The overall architecture of our pipeline is given in figure 1 below:


Evidence Sources: news database, database of previous fact checks
Input: Claim
Evidence Retrieval
Set of Evidence
Augmented Promt: claim, factor labels, and evidence
LLM
Factuality Factor Models
Set of Labels
Final Output: Veracity rating and justification
Predictive models: Sentiment, political bias, logical fallacies, speaker credibility, source reliability, textual manipulation

3.1 Predictive Models

Previous work from other researchers have shown extensive potential in predictive models for misinformation detection, and we have used similar techniques to emulate their success. Our project aims to extend their work and combine a series of models pertaining to different factuality factors to improve overall performance. These factuality factors are meant to provide the model with more information pertaining to a claim which may help it determine the veracity. For example, if the model is told that the claim was taken from a source which is typically biased towards certain viewpoints, it may pay more attention to misinformation associated with these viewpoints.

For each factuality factor, we have one model which accepts the claim as an input and outputs a label (or continuous value) describing the claim's classification (or score) according to that factuality factor. The factuality factors we evaluated are as follows:

1. authenticity 1 (comparison to other fact-checked claims)
   - Approximate nearest neighbor search (using Weaviate's built in similarity search functionality) across a vector database containing previously fact-checked claims and their veracity labels. The 10 most similar claims are retrieved.
2. authenticity 2 (comparison to other news sources)
   - Approximate nearest neighbor search (using Weaviate's built in similarity search functionality) across a vector database containing 2-3 sentence chunks of news articles from reliable sources (NPR, CNN).
3. sentiment analysis
   - Ternary classification of the claim's sentiment. Inputs are classified as positive, neutral, or negative. The model we used was a distilBERT-base model tuned for sentiment analysis.
4. logical fallacy detection
   - Binary classification of claims into two classes: those containing logical falla-

cies and those with no fallacious logic. The model employed was an AdaBoost classifier trained on Google's Big Bench dataset.

5. clickbait
   - Binary classification of headlines into clickbait vs non-clickbait headlines. The training dataset came from Kaggle, and the features encompass aspects such as the number of words, parts of speech, use of punctuation, the presence of hashtags, average word length, the ratio of uppercase characters, etc. We used a simple feedforward neural network as the classifier for this model.
6. source reliability
   - A prediction of the source's overall reliability, based on their previous publications which have been reviewed by PolitiFact. Inputs are vectorized using TF-IDF and passed into a logistic regression model for multiclass classification, where the target classes are the same truth-o-meter classes used for individual claims.
7. credibility
   - A random forest regressor used to evaluate the credibility of the speaker based on previous claims they've made that have been fact checked.
8. context veracity
   - This model is applied at the level of an entire article, rather than an individual claim. It predicts whether there is a drift in the sentiment, topics, or named entities contained within an article across multiple chunks. In other words, whether the article presents inconsistent narratives. The model is a weighted sum of the shift in sentiment score, frequency of topics (derived from the latent Dirichlet allocation), and the named entities extracted using an NER model. The exact score is determined by:

$$0.4 \cdot \text{topic\_drift} + 0.4 \cdot \text{sentiment\_drift} + 0.1 \cdot \text{ner\_drift} + 0.1$$

9. political affiliation
   - The predicted political affiliation of the speaker, one of: left, neutral, and right. We used a tuned BERT model for classification.
10. style (text manipulation)
    - This is a binary classification task to predict whether the text has been computationally manipulated (such as text generated by language models). The model we used was introduced in Jawahar, Abdul-Mageed and Lakshmanan (2022), and consists of a RoBERTa model combined with a graph convolutional network designed to reason across a knowledge graph.
11. spam
    - Predicting whether or not a claim is likely to be spam. The training dataset is based on a list of spam trigger words commonly used in spam filters, the embeddings are computed using a bag-of-words method with a count vectorizer. The model used for classification is a K-Nearest Neighbors classifier with K = 3.

Some of these models (clickbait, context veracity, and spam) involve comparing different parts of a news article to one another, and as such can't be applied to the sentence-level claims that we evaluate our model on in this paper. Instead, we use these models in our interactive website, the second deliverable of our project.

3.1.1 Enrichment

We employed a series of data enrichment tasks on our dataset using BERT models to bolster the performance of our predictive models. Firstly, we conducted keyword extraction to identify and extract the most pertinent words or phrases encapsulating the main topics within each article. Following this, we utilized article summarization techniques to condense the full article into a concise version without losing too much information. Lastly, topic modeling was employed to uncover latent topics present across the entire dataset. These enrichment tasks were performed through a combination of text preprocessing, tokenization, scoring, and selection procedures tailored to each specific task.

3.2 Evidence Retrieval

In addition to factors such as sentiment and political affiliation, it's important for the LLM to have references containing the ground truth. After all, without knowing what's true, it's impossible to detect falsities. While LLMs have exhibited a considerable amount of knowledge regarding the world, they are inherently limited by the scope of their training data. To get around this limitation, we employed retrieval augmented generation, a technique in which we query knowledge databases for relevant evidence, retrieve said evidence, and pass them in to the LLM in addition to the claim.

The knowledge database was built using news articles scraped from CNN and NPR. While these sources are known to exhibit slight center-left bias, their reporting is generally considered factual and bias is minimal compared to other alternatives. They also had the most accessible data for us to scrape, since many other sites (such as the New York Times) hide content behind paywalls. We scraped the articles from each of these sources, separated them into chunks (one to three sentence subsections), and computed BERT embeddings for each sentence which served as the entries in a vector database.

We also looked into adding Wikipedia as an additional source, since it's even less biased than the two sources previously mentioned, and has more extensive information on a wider variety of topics. However, due to the size of the Wikipedia dumps, we ran into trouble with both vectorization (which is computationally expensive) and storage of this data, and as such we weren't able to implement this component.

We also tried a different approach to get evidence from Wikipedia using a combination of Named entity Recognition and Wikipedia's API. In this approach, we conducted NER on the input claim to extract the "keywords", which we then queried for using Wikipedia's API. If any article was returned, we vectorized it and selected the k most similar sentences to use as evidence. This didn't improve our performance at all, and the evidence was generally of low quality, since many claims pertained to complex topics that couldn't be answered just based on information contained at the level of single sentences. As such, we decided to exclude Wikipedia from our knowledge database, although we still believe it's a promising source which should be explored with more time and computational power.

While we evaluated many approaches to the retrieval of evidence, including many projects regarding the FEVER task (such as BEVERS DeHaven and Scott (2023)) and other adjacent projects (such as MUSER Liao et al. (2023), we ultimately opted to use a simple similarity search (based on cosine similarity) over our vector database due to the high computational requirements for the other implementations. We believe this is the largest shortcoming of our project, and it's an area that we intend to concentrate on with future work.

The retrieved evidence set, along with the outputs from the factuality factor models, is then added to a prompt containing the claim and an explanation of the 0-5 classification scale we're using. Finally, this prompt is passed in to a LLM for final classification and justification generation.

3.2.1 Previous Fact Checks

In addition to retrieving this evidence from news articles, we also decided to make a separate database which contains approximately 20,000 fact-checked claims scraped from politifact. The claims, along with metadata such as their veracity labels, the speaker, date of the claim, and justification for the label, are stored in a vector database. At evaluation, we employ the same similarity search across the vector database to query for the 10 most similar claims, which we pass in to the LLM (along with their metadata) to offer it additional context.

3.3 Generative Model

LLMs have demonstrated their ability to summarize documents effectively and generate quality textual outputs, and they've also exhibited impressive reasoning ability, as explained in Huang and Chang (2023). Through the retrieval of evidence and the results of our predictive models, we attempt to provide the LLM with sufficient information to create an accurate, informative response.

Specifically, we chose to use Google's Gemini 1.0 pro large language model for this paper. In our prompt, we first explain the rating system used to classify the claims, and then we augment the prompt to our LLM with the results of our predictive models and the retrieved evidence.

We also experimented with Google's Gemma 2 billion parameter model, which we fine-tuned using LoRA (Hu et al. (2021)).

4 Results

4.1 Evaluation of Predictive Models on Various Subtasks

Task	Architecture	Performance
Political Bias Prediction	Fine-Tuned BERT	0.69 (3-way accuracy)
Sentiment	distilBERT	N/A
Credibility	Random Forest	8.175 (MSE)
Textual Manipulation	Hybrid RoBERTa + GNNs	0.824 (binary accuracy)
Source Reliability	Logistic Regression	0.55 (6-way accuracy)
Logical Fallacies	Adaboost Classifier	0.71 (binary accuracy)

4.2 Pipeline Evaluation

We tried a number of different pipelines involving various combinations of the aforementioned components. In all the models, we used the LLM as our primary reasoning module, and passed in the other components (such as evidence, related fact-checks, and the results of our predictive models) as part of the prompt.

To evaluate our pipelines, we used two different datasets of fact-checked claims from PolitiFact, rated according to PolitiFact's truth-o-meter, whose classes we have again listed in descending order of veracity below:

1. TRUE – The statement is accurate and there's nothing significant missing.
2. MOSTLY TRUE – The statement is accurate but needs clarification or additional information.
3. HALF TRUE – The statement is partially accurate but leaves out important details or takes things out of context.
4. MOSTLY FALSE – The statement contains an element of truth but ignores critical facts that would give a different impression.
5. FALSE – The statement is not accurate.
6. PANTS ON FIRE – The statement is not accurate and makes a ridiculous claim.

The first dataset is LIAR-plus (Alhindi, Petridis and Muresan (2018)). This dataset contains approximately 12,000 claims from PolitiFact from before 2018, each falling into one of the categories outlined above. The table below contains our pipeline's results on this dataset.

LIAR-plus

Trial	Accuracy	Macro-F1
Simple prompt (Gemini)	0.17	0.17
In-context learning (Gemini)	0.23	0.22
ICL + Evidence (Gemini)	0.29	0.20
ICL + evidence + predictive models (Gemini)	0.23	0.22
ICL + similar fact checks (Gemini)	0.62 ±5	0.56
LORA-tuned Gemma 2B	0.19	0.14

Our pipeline received moderate improvements over the baseline when we included in-context learning (explanations of the classifications, their differences, and examples of each), evidence, and the outputs of the predictive models, but the factor that provided the most significant boost in performance was the inclusion of similar fact-checks. By passing in similar claims which had already been fact-checked, our accuracy rose to 62%. It is worth noting that while including predictive models gave a boost over the baseline, this implementation performed worse than any other non-baseline approach that we took.

Including single sentences as evidence showed some improvement in some cases, but overall didn't have the intended effect, even decreasing the accuracy overall when combined with our most effective model. To address this, we tried to improve the quality and extent of the justifications by using MUSER evidence retriever (Liao et al. (2023)) to retrieve relevant sentences, before passing these sentences into Gemini for summarization. Then, we produced a re-ranking of how relevant the evidence is to the claim, which we also included in the prompt to provide insight into the relatinoship between the claim and evidence. However, due to computational constraints, we could only run MUSER on a database of 20,000 news articles, which we don't believe was sufficient to retrieve insightful evidence in most cases. As a result, it did not improve the quality of retrieved evidence.

We also evaluated the possibility of fine-tuning the 2 billion parameter Gemma instruct model, but we quickly ran into difficulties due to the low quality of the justifications present in our training data, many of which displayed circular reasoning such as "this claim is true because it is true". As a result of this, the explanations generated by the fine-tuned model were often nonsensical, and showed no signs of improving through further training. As a result of this and hardware limitations, we decided to move away from Gemma for the remainder of the project, although we would like to emphasize that we do think that with better training data, this could be a theoretically viable approach based on our review of related literature and the results of other models (which we will examine in section 5.1).

While the previous dataset contained data from 2018 and earlier, we also wanted to evaluate our pipeline on more current claims. For this purpose, we used our dataset scraped from PolitiFact's fact checks and evaluated our model on all claims from 2022 and 2023, with the results presented below.

PolitiFact 2022-2023

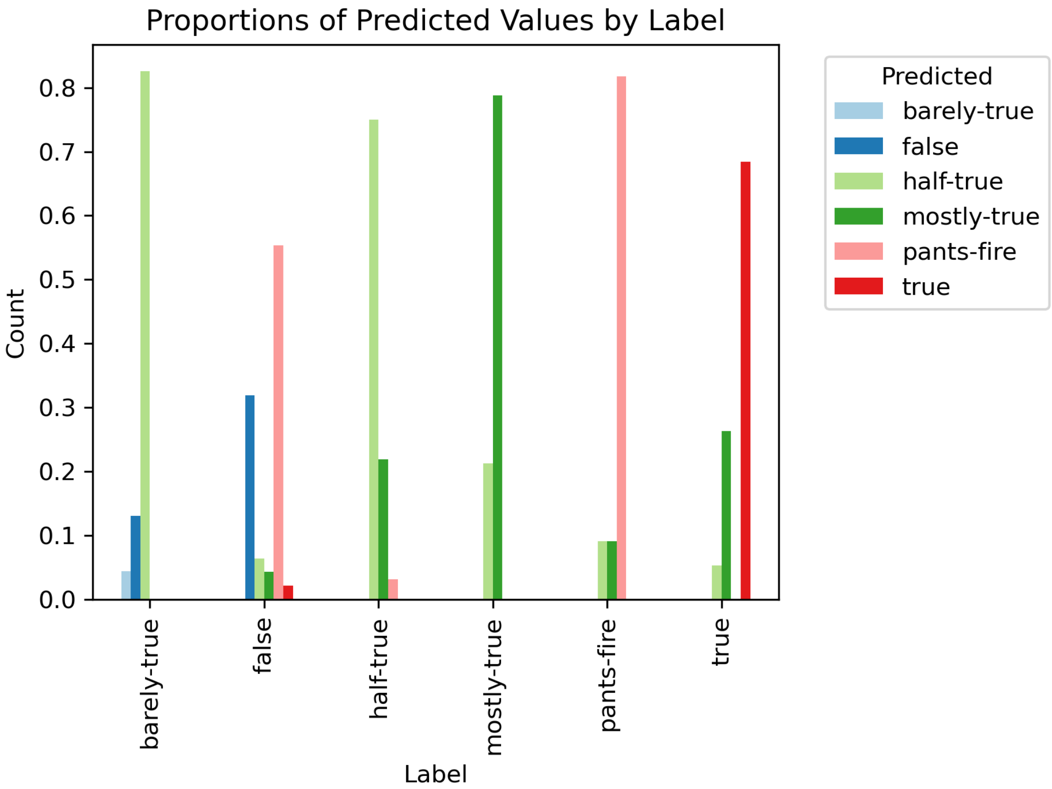
Trial	Accuracy	Macro-F1
Simple prompt (Gemini)	0.23	0.18
In-context learning (Gemini)	0.27	0.23
ICL + Evidence (Gemini)	0.28	0.20
ICL + similar fact checks (Gemini)	0.40 ±10	0.56 ±10
ICL + fact checks + predictive (Gemini)	0.30	0.15

Due to the fact that some, but not all, of the claims in LIAR-plus were present in the knowledge database, we believe that the pipeline's performance on the claims from 2022-2023 is more reflective of its generalizability and true performance on unseen data (since none of these claims were present). Below, we've shown the top performing pipeline's label-wise accuracy on this dataset, where the accuracy is evaluated on a subset of the dataset containing each the true label.

Label-Wise Accuracy Breakdown

Label	Accuracy
true	0.684
mostly-true	0.788
half-true	0.75
barely-true	0.043
false	0.319
pants-on-fire	0.818

Interestingly, the model performs decently when it comes to most labels, with the exception of false and barely-true. We can gain some insight on this from the figure below, where we have plotted the proportion of each class's predictions that fall into each respective class:



Many classes display fairly good accuracy (such as pants-on-fire and mostly-true which saw around 80% of their instances predicted correctly). The important aspect of this graph is related to **how** the model mispredicted on the labels for which it performed poorly. Among claims which were barely-true, which only reported around 4% accuracy, the model predicted half-true over 80% of the time. This is important because half-true and barely-true are adjacent labels, and arguably have quite similar criteria. Similarly, among claims that fell into the false label (the class with the next lowest accuracy), we see that approximately 30% are predicted correctly, but nearly 60% are predicted as pants-on-fire, which is yet another label which has similar meaning to the correct one.

5 Discussion

5.1 Analysis of our Best Model

Arguably the most important and encouraging result from our investigation comes from our analysis of the model's missed classifications, and what exactly it predicted instead of the correct label. In the case of both the poorly performing classes, the model mistook the claims for instances of very similar classes, which arguably lack a clear delineation between them.

This could be evidence towards the model's ability to identify the general veracity of a claim, even if it struggles to place them into these six (somewhat subjectively-defined) classes. All the instances of barely-true claims, for example, were classified as either false, barely-true or half-true. This is highly encouraging considering the fact that barely-true is meant to lie between half-true and false, though in our opinion its definition bears more resemblance to half-true claims than false ones. Similarly, among false claims, well over 80% of the claims are classified as either false or pants-on-fire, which are two other classes whose delineation is not particularly clear in many cases. As such, we believe that the model has the potential to be quite performant if it's able to learn to distinguish between these "nearby" labels through fine-tuning, further clarification in the prompt, or provision of additional evidence and knowledge.

It is worth noting that the model had its fair share of claims which were badly misclassified, such as the 20% of pants on fire claims which it classified as mostly or half true. These are fairly serious misclassifications which would need to be rectified before the pipeline could be used in any practical environment, but overall we find these results highly encouraging.

5.2 Temporal Differences in Performance

There are two interesting results related to the temporal differences in the datasets that we'd like to emphasize here. First, we see that the baseline models tended to do worse on the data from prior to 2018 (LIAR-plus), but marginally better on the data from 2022 and onwards. Our speculation leads us to believe that this is partially due to the fact that the underlying LLM (Gemini 1.0) is tuned to optimize question answering on more recent data, but may not be as reliable when it comes to evaluating older claims like those in LIAR-plus, some of which date back to 2007. The second interesting result is the effect of including evidence and previous fact checks in the pipeline. Including similar fact checks increased the accuracy on the pre-2018 data by much more than the post-2022 data. We attribute this to the fact that patterns of misinformation could tend to remain similar throughout a given time periods, and that all of the fact checks in our knowledge database were from before 2022 (to avoid overlap with our evaluation dataset). As a result, many of the talking points and common points of deception present in data after 2022 won't appear in the fact checks that our model has access to, making them less helpful for these later claims. On the other hand, there are likely many claims in the knowledge database which are quite similar to those present in LIAR (since the time periods these claims were taken from do overlap). These highly similar claims seem to provide the model with important information, based on the massive increase in both accuracy and F1 score between the ICL-only prompt and the prompt which has access to similar fact checks.

We also saw an interesting result when it came to integrating the predictive models. The model seemed to become "distracted" by the inputs of the predictive models, as can be seen in the massive drop in performance. Upon evaluating the responses generated from this prompt, we saw that virtually all of them focused solely on the results of the predictive models, with virtually no mention of relevant events derived from the model's own knowledge or the fact checks passed in. We tried tuning the prompt by instructing the model to weigh the predictive results and its own knowledge differently, but the results were at best equivalent to those obtained without the predictive models. We do believe that there is potential for improvement with this technique, but we weren't able to adjust the prompt accordingly to achieve such a boost in performance.

5.3 Inconsistency of LLM outputs

Perhaps our most serious point of concern was the high variability between responses when given identical prompts, which resulted in some drastically different results across trials. While many of the responses didn't vary too widely (from outputting "True" to "Mostly True", for example), sometimes the model would output "True" on one trial and "False" on the next. This issue presented itself especially when we incorporated more inputs (such as the predictive models, evidence, and previous fact checks). In fact, in our trials with in-context learning and previous fact-checked claims on the 2022 data, the accuracy ranged from 35% to 53%! Interesting, the accuracy of this same pipeline on the data from before 2018 was much more consistent, ranging from 60% to 69%.

We suspect that this may be a combination of the model being too "confident" based on the information provided, which sometimes leads it astray, and the fact that the evidence database is not sufficiently large. Scraping a year's worth of news articles is likely insufficient to fill all the gaps in the LLM's knowledge regarding various subjects, which is especially apparent in this task since the claims cover such a wide variety of subjects. We believe this is one of the largest areas for improvement, as we've seen that when the LLM has access to relevant information (such as with the previous fact-checks, which related more closely to the claims in the LIAR dataset), the pipeline was actually fairly consistent and somewhat reliable.

5.4 Limitations and Future Work

As mentioned before, we believe that retrieving more relevant information for a wider variety of claims would make an immediate impact on the pipeline's performance. As such, we would focus future work on incorporating a more advanced evidence retrieval module (as opposed to our simple nearest-neighbor search), with access to a much larger knowledge database containing information on a wider set of subjects.

Furthermore, we believe that training more powerful predictive models would provide a boost in performance as well. In theory, providing the LLM with additional context regarding potential biases contained in the claim may help it more clearly identify its flaws, but if these potential biases are misidentified by the predictive models, it could just lead the LLM astray, as it appeared to do in our case.

Finally, we believe that there is a good deal of potential in fine-tuning the LLM, as this may help it learn the minutiae necessary to differentiate between nearby labels. In our trials, we found that (depending on the trial) roughly 70% to 80% of the claims were within one label of the correct one, but it seemed that the model struggled to understand the criteria that separate these nearby labels. We think that fine-tuning through LoRA (Hu et al. (2021)) may help the LLM learn to distinguish between these labels, and potentially learn other patterns of misinformation.

6 Conclusion

Our exploration of the augmentation of large language models (LLMs) for misinformation classification and correction showed some promising results, and also shed light on the potential pitfalls of this approach. Through the retrieval of evidence and the results of our predictive models, we attempted to provide a LLM with sufficient information to create an accurate, informative response to a potentially misinformative claim.

While we didn't get spectacular results, we did improve on the baseline presented in the original LIAR paper (with our best-performing model reaching up to 53% label accuracy in one trial, around 40% more consistently), and we have conditional evidence that the augmentation of the LLM with external knowledge and select predictive models could result in improved performance.

While our implementation lacks generalizability, we were able to show that LLMs can be quite performant on the task of veracity classification under the right circumstances (as was the case with the LIAR-plus dataset, where we achieved over 60% label accuracy thanks to the dataset's close relationship to the knowledge in our database). We also see that even when the knowledge-augmentation fails to provide relevant context to the model, it still performs reasonably well and seems to make predictions which are close to the true labels. These results indicate that there is potential in the use of LLMs for automated fact-checking, and that they could prove to be a viable step forward in this field.

References

Abdullah-All-Tanvir, Ehesas Mia Mahir, Saima Akhter, and Mohammad Rezwanul Huq. 2019. “Detecting Fake News using Machine Learning and Deep Learning Algorithms.” In *2019 7th International Conference on Smart Computing Communications (ICSCC)*. [Link]

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)*.

B, Athira A, S D Madhu Kumar, and Anu Mary Chacko. 2022. “Towards Smart Fake News Detection Through Explainable AI.”

Choudhary, Anshika, and Anuja Arora. 2021. “Linguistic feature based learning model for fake news detection and classification.” *Expert Systems with Applications* 169, p. 114171. [Link]

DeHaven, Mitchell, and Stephen Scott. 2023. “BEVERS: A General, Simple, and Performant Framework for Automatic Fact Verification.”

Gebhard, Lukas, and Felix Hamborg. 2020. “The POLUSA Dataset: 0.9M Political News Articles Balanced by Time and Outlet Popularity.” In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. ACM. [Link]

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. “LoRA: Low-Rank Adaptation of Large Language Models.”

Huang, Jie, and Kevin Chen-Chuan Chang. 2023. “Towards Reasoning in Large Language Models: A Survey.” In *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada Association for Computational Linguistics. [Link]

Jawahar, Ganesh, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2022. “Automatic Detection of Entity-Manipulated Text using Factual Knowledge.”

Liao, Hao, Jiahao Peng, Zhanyi Huang, Wei Zhang, Guanghua Li, Kai Shu, and Xing Xie. 2023. “MUSER: A MUlti-Step Evidence Retrieval Enhancement Framework for Fake News Detection.” In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. [Link]

Popat, Kashyap, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. “DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning.”

Sarrouti, Mourad, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. “Evidence-based Fact-Checking of Health-related Claims.” In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic Association for Computational Linguistics. [Link]

Vlachos, Andreas, and Sebastian Riedel. 2014. “Fact Checking: Task definition and dataset construction.” In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA Association for Computational Linguistics. [Link]

Wang, William Yang. 2017. ““Liar, Liar Pants on Fire”: A New Benchmark Dataset for

Fake News Detection." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada Association for Computational Linguistics. [Link]