Hybrid Approaches for Disinformation Detection: Integrating Predictive and Generative AI Models

Yiheng Yuan
Halıcıoğlu Data Science Institute
University of California, San Diego
San Diego, USA
yiy159@ucsd.edu

Luran Zhang
Halıcıoğlu Data Science Institute
University of California, San Diego
San Diego, USA
luz010@ucsd.edu

Jade Zhou
Halıcıoğlu Data Science Institute
University of California, San Diego
San Diego, USA
gzhou@ucsd.edu

Dr. Ali Arsanjani
Google Cloud, Applied AI Engineering
Google
Mountain View, USA
arsanjani@google.com

***Abstract*—Misinformation and disinformation threaten public trust and decision-making. This project develops an AI-powered system combining predictive and generative models to detect, rank, and mitigate false content in news articles. Using the LIAR-PLUS dataset and six initial Factuality Factors (expanding to twelve), the predictive model generates veracity scores through feature extraction and machine learning, while the generative model applies Fractal Chain of Thought (FCoT) prompting for improved contextual analysis.**

Key innovations include asynchronous web scraping, vector database storage, graph database integration, and real-time context enrichment. FCoT prompting enhances consistency and accuracy in truthfulness evaluation. This work establishes a scalable framework for misinformation detection, with future improvements focusing on expanded factors, refined scoring, and efficient data management.

***Index Terms*—generative AI, disinformation, misinformation**

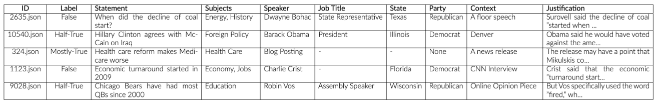
I. Introduction

Misinformation and disinformation have become pressing issues in the digital age, significantly affecting public trust and decision-making. While misinformation refers to the unintentional spread of false information, disinformation is intentionally misleading, crafted to deceive audiences. Both phenomena distort public understanding on critical issues, often exacerbating confusion and societal polarization. To address this challenge, we have developed a combined predictive and generative AI system, which is designed to detect and curb the spread of misleading content. By leveraging the capabilities of generative AI, the model can detect and mitigate the spread of false or misleading content. While challenges around accuracy and ethical considerations remain, our approach strives to balance these factors in creating a responsible solution.

Our project begins by scraping relevant articles, which are then divided into manageable segments, either sentences or paragraphs, then stored in a vector database along with key metadata, such as author and publisher details. To enhance detection accuracy, we utilize 6 initial factuality factors (with plans to expand to 12) that are broken down into mini-factors. These factors help guide the predictive model's focus on different aspects of misinformation detection. The predictive model is then integrated with a generative AI model and a meso UI page.

A. Data: Liar-Plus Dataset

The predictive model is trained on the LIAR-PLUS dataset, which includes sufficient labeled statements from Politifact reports covering topics like politics, economics, and social issues. Each statement is assigned a credibility label, from ”pants-on-fire” to ”true,” along with metadata such as subject, speaker name, job title, state information, party affiliation, total credit history count, context, and extracted justification. The extensive metadata makes LIAR-PLUS a valuable dataset for training models capable of detecting misinformation and disinformation.



ID	Label	Statement	Subjects	Speaker	Job Title	State	Party	Context	Justification
2635.json	False	When did the decline of coal start?	Energy, History	Dwayne Bohac	State Representative	Texas	Republican	A floor speech	Surovell said the decline of coal ”started when ...
10540.json	Half-True	Hillary Clinton agrees with McCain on Iraq	Foreign Policy	Barack Obama	President	Illinois	Democrat	Denver	Obama said he would have voted against the ame...
324.json	Mostly-True	Health care reform makes Medicare worse	Health Care	Blog Posting	-	-	None	A news release	The release may have a point that Mikulskis co...
1123.json	False	Economic turnaround started in 2009	Economy, Jobs	Charlie Crist	-	Florida	Democrat	CNN Interview	Crist said that the economic ”turnaround start...
9028.json	Half-True	Chicago Bears have had most QBs since 2000	Education	Robin Vos	Assembly Speaker	Wisconsin	Republican	Online Opinion Piece	But Vos specifically used the word ”fired,” wh...

Fig. 1. Sample Dataset from LIAR-PLUS

Five columns are not shown in the sample table. They represent five labels used to assess statement accuracy:

The ”Barely True” includes statements with some truth but are largely misleading or lacking key facts. The ”False” applies to statements that are entirely inaccurate, misrepresenting facts or fabricating information. The ”Half True” refers to statements that are partially correct but omit crucial details, altering their overall meaning. The ”Mostly True” covers statements that are largely accurate but may contain minor errors or missing context. The ”Pants on Fire” is for statements that are not only false but also outrageously misleading or deceptive.

B. Data: Politifac

We use data from the Politifact fact-checking website (https://www.politifact.com/) as a reference for our analysis. Specifically, we rely on the human labeling provided by Politifact to categorize statements based on their accuracy. These labels serve as a benchmark, allowing us to compare our results against expert fact-checking assessments.

II. Scraping Articles and Chunking Content for Vector Database Storage

In this project, the process of collecting and managing text data is essential to creating a factuality assessment model capable of handling diverse content efficiently. We utilized advanced scraping methods, iterative processing strategies, and structured storage techniques to prepare data for analysis.

A. Web Scraping with Enhanced Querying

To gather article data, we employed web scraping combined with external query APIs to enrich the factuality assessment process. We developed a Python-based scraper using the `requests` library to pull article content from online sources, such as Politifact, a fact-checking website. In addition, we integrated SerpApi to perform external validation of extracted claims. This API enables querying search engines directly, providing top results for claims and facilitating external cross-checks against reliable sources.

The scraping pipeline identifies relevant portions of each article, such as the claim, author, date, and factuality rating, capturing structured data critical for model training. SerpApi queries supplement this process by verifying claims' presence and context in search engine results. To minimize server load and avoid blocking, we implemented rate-limiting mechanisms and batch processing for API calls. These enhancements yielded a collection of diverse claims, enriched with external validation data, enabling the model to analyze statements across broader contexts.

B. Chunking Text for Targeted Analysis

The data collected through scraping and querying is cleaned and processed for analysis by predictive and generative models. To handle the size and complexity of articles, we divided each article into smaller segments or "chunks," typically containing up to 2,000 words. This chunking strategy ensures that each segment remains cohesive and meaningful, enabling specific parts of the article to be analyzed in isolation while maintaining contextual accuracy.

In this project, chunking serves an additional purpose: it aligns the text data with downstream processes, such as Fractal Chain of Thought (FCoT) prompting, a method for iterative analysis. By segmenting the text, each chunk undergoes independent analysis and iterative refinement, ensuring a focused and accurate evaluation.

C. Storing and Embedding Chunks in a Vector Database

To efficiently store and retrieve these chunks for further analysis, we utilized Weaviate, a vector database optimized for handling text embeddings and similarity searches. Each chunk was stored as a vector embedding using the `text2vec-google` module, ensuring the retention of its semantic representation. Storing text in this format supports quick access for similarity comparisons and scoring, key components in the factuality analysis pipeline.

Each chunk is assigned a unique identifier and stored in the database alongside metadata such as claim content, source, and validation scores derived from SerpApi queries. This structure facilitates seamless integration with hybrid analysis models, allowing targeted retrieval for predictive and generative AI processes.

III. Predictive Model For Factuality Factors

As a group, we had six factuality factors to start with. We divide by the number of factors in half and evenly divide them between predictive models and generative AI models. We each created a predictive model for one factor that we thought is more suitable as they make more sense to create numerical features. For each of our functions, we use the text statement from the LiarPlus dataset as our input and return a feature containing the sub-scores for each of the mini-factors for out factuality factor.

The three factuality factors used in the predictive model are authenticity check, content statistics, and linguistics and toxicity.

- **Authenticity Check**: Evaluates whether a statement aligns with known facts and sources. It involves cross-referencing with reliable datasets and verifying the credibility of sources cited within the text.
- **Content Statistics**: Focuses on structural and numerical aspects of the statement, such as word count, sentence complexity, and the presence of numerical claims. These statistical features help assess the consistency and reliability of the content.
- **Linguistics & Toxicity**: Analyzes the language style, sentiment, and toxicity level of the statement. It examines linguistic patterns, emotional tone, and potentially misleading or inflammatory language that could indicate biased or deceptive content.

The flowchart outlines the veracity assessment process for predictive models. The process begins with an input text statement, which undergoes preprocessing using NLP techniques to clean and structure the data. Next, feature extraction is performed based on the three factuality factors—authenticity check, content statistics, and linguistics and toxicity. These extracted features are then fed into model training, where predictive algorithms analyze patterns and correlations. Finally, the trained model outputs a veracity assessment result, providing a score or classification for the statement's factual accuracy.

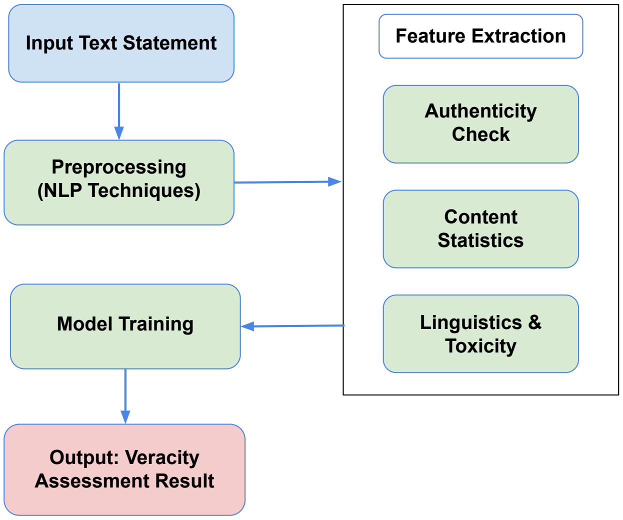


Fig. 2. Predictive Model Flow Chart

A. Feature Extraction and Classification Model

The analysis focuses on developing models to assess the factuality of statements based on various content-based indicators. The approach employs machine learning techniques to classify text by factual accuracy, incorporating natural language processing (NLP) methods to quantify elements that suggest factual reliability. Using feature extraction methods such as `CountVectorizer` and `TfidfVectorizer`, the model captures essential textual patterns, while sentiment analysis with `VaderSentiment` adds contextual nuance by examining sentiment tones that may correlate with reliability. A `RandomForestClassifier` is then used to analyze these characteristics, aiming to effectively differentiate between factual and nonfactual statements. This framework leverages both structural and linguistic factors to build a model that not only predicts factuality, but also provides insight into the attributes that enhance or diminish credibility.

B. Model Output

The final output from the predictive model is a veracity score, calculated based on the predicted probabilities of each label and their assigned weights, ranging from 1 to 6. The ”Pants on Fire” label receives the lowest weight (1), while statements classified as completely true are assigned the highest weight (6). As the classification moves toward higher factual accuracy, the assigned weight increases accordingly. The veracity score is then obtained by multiplying each label's probability by its corresponding weight, resulting in a final score between 1 and 6.

IV. Generative AI model

The generative AI component of our model is designed to bring nuanced language understanding to factuality assessment. For this purpose, we used Google Gemini, a large language model capable of evaluating statements on multiple dimensions of truthfulness.

A. Model Configuration

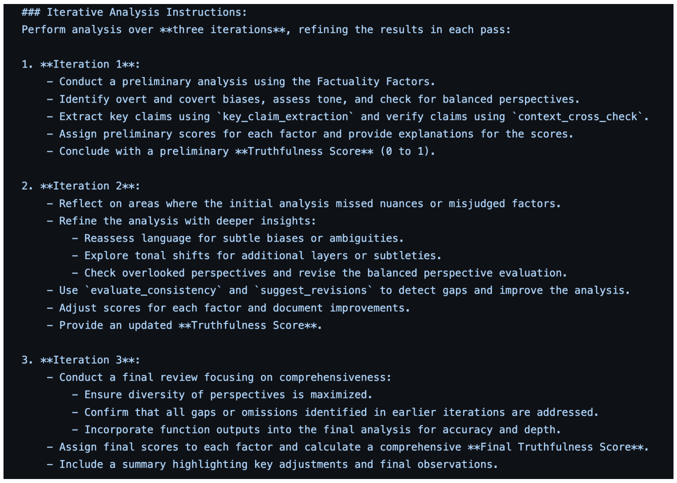
We configured the generative AI model using Google Gemini's API, setting parameters such as `temperature`, `top_p`, and `max_output_tokens` to fine-tune its responses. These parameters control the diversity and length of the output, allowing us to balance between generating informative responses and maintaining relevance. We prompt the model with specific questions about each chunk's content, designed to evaluate factors such as language bias, tonal bias, and perspective balance.

B. Evaluating Factuality Factors

The generative model assesses three specific factuality factors: **Biases Factuality Factor**, **Context Veracity Factor**, and **Information Utility Factor**. Each factor contains subfactors evaluated by the model, such as language analysis, tonal analysis, consistency checks, and content value. By using a structured prompt with clear instructions for each factor, we obtain a `Final Truthfulness Score` for each chunk, which is a numerical representation between 0 and 1. This score reflects the model's assessment of the chunk's factuality, with 1 indicating complete truthfulness.

C. Fractal Chain Of Thought

Fractal Chain of Thought (FCoT)is an advanced approach that introduces a layered reasoning process for GenAI model, enabling it to analyze problems across multiple dimensions. We use the prompt to guide the model through three iterations, refining its response with each successive iteration. An example prompt is provided in Fig. 1. This method mimics the way humans detect misinformation by breaking down complex tasks into smaller, interconnected components, allowing for a deeper and more structured understanding. Compared to traditional prompts, FCoT provides a more nuanced and consistent analysis.



```
### Iterative Analysis Instructions:
Perform analysis over **three iterations**, refining the results in each pass:

1. **Iteration 1**:
    - Conduct a preliminary analysis using the Factuality Factors.
    - Identify overt and covert biases, assess tone, and check for balanced perspectives.
    - Extract key claims using `key_claim_extraction` and verify claims using `context_cross_check`.
    - Assign preliminary scores for each factor and provide explanations for the scores.
    - Conclude with a preliminary **Truthfulness Score** (0 to 1).

2. **Iteration 2**:
    - Reflect on areas where the initial analysis missed nuances or misjudged factors.
    - Refine the analysis with deeper insights:
        - Reassess language for subtle biases or ambiguities.
        - Explore tonal shifts for additional layers or subtleties.
        - Check overlooked perspectives and revise the balanced perspective evaluation.
    - Use `evaluate_consistency` and `suggest_revisions` to detect gaps and improve the analysis.
    - Adjust scores for each factor and document improvements.
    - Provide an updated **Truthfulness Score**.

3. **Iteration 3**:
    - Conduct a final review focusing on comprehensiveness:
        - Ensure diversity of perspectives is maximized.
        - Confirm that all gaps or omissions identified in earlier iterations are addressed.
        - Incorporate function outputs into the final analysis for accuracy and depth.
    - Assign final scores to each factor and calculate a comprehensive **Final Truthfulness Score**.
    - Include a summary highlighting key adjustments and final observations.
```

Fig. 3. FCoT Prompt

In practice, a GenAI model processes an article by dividing it into smaller chunks and analyzing each chunk individually.

By comparing the results generated for each chunk using FCoT and normal prompts, we observe that FCoT delivers more consistent findings across all chunks. This aligns with expectations, as all chunks originate from the same article and should logically produce coherent results. The consistency offered by FCoT highlights its reliability and superiority in maintaining alignment between different segments of a text, making it an invaluable tool for misinformation detection.

D. *Function Calling*

To effectively leverage the concept of Fractal Chain-of-Thought (FCoT) prompt engineering, it is essential to define clear and measurable objective functions that serve as evaluation metrics for each iteration within the Fractal CoT framework. For our project, these objective functions were derived from the feature extraction processes integrated into our predictive models. These features form the basis for evaluating the generative model's outputs in a structured and quantifiable manner. When constructing prompts for the generative AI, the model is tasked with assessing each iteration based on pre-defined metrics tied to key factors relevant to misinformation detection. These metrics are designed to ensure iterative improvement and alignment with the objectives of the task. We passed these objective functions as `tools` objects, a declarative mechanism within the function-calling paradigm. This approach enables the model to interpret and utilize the functions effectively, identifying when and how a particular function can assist in generating more accurate, contextually relevant responses. By systematically integrating these functions into the iterative prompt design, we ensured a rigorous and adaptive process to evaluate and refine outputs at every stage.

E. *Search Engine Scrapping*

To enhance the generative AI model's ability to identify misinformation within a corpus of news articles, it is crucial to provide a robust contextual understanding of the topic at hand. To achieve this, we incorporated search engine scraping as a method to gather supplementary context information from recent and relevant web posts. This strategy enables the model to process and analyze current narratives and perspectives surrounding a given topic. Specifically, we utilized the **SerpAPI** to scrape Google search engine results. This API allowed us to efficiently extract the most pertinent search results. Then we extracted the title and snippet of the top 5 search results, which were then integrated into the prompt as an additional layer of context. By incorporating real-time, topically relevant information into the generative AI's input, we enhanced its ability to detect discrepancies, corroborate claims, and evaluate the credibility of the information presented in the news articles. This approach bridges the gap between static data and dynamic, real-world content, providing the AI model with a richer informational foundation to identify misinformation effectively.

F. *Score Interpretation and Output*

The model's output provides a breakdown for each subfactor, detailing how language and context influence the statement's perceived truthfulness. This nuanced analysis captures aspects of the text that might not be immediately evident, such as subtle biases in tone or shifts in context that could affect meaning.

V. Combining GenAI with Predictive Model

A. *Dual-Model Approach*

The final misinformation detection model uses a dual-model approach, combining a GenAI model (Google Gemini) with a pre-trained predictive model, which were both discussed previously in the report.

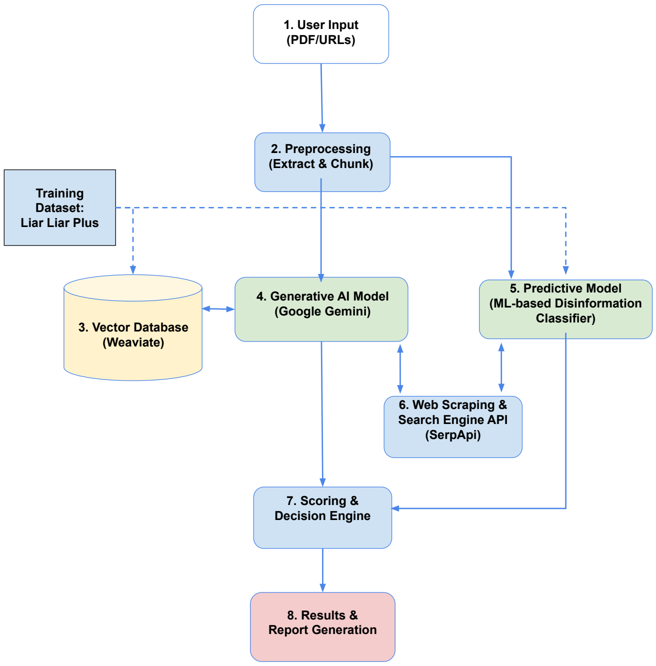


Fig. 4. Dual-Model Approach Flow Chart

The misinformation detection model follows a structured pipeline to ensure accurate and comprehensive evaluation of text. The process begins with user input, where users provide documents or URLs for analysis. Next, the input undergoes preprocessing, which involves extracting and chunking the text to prepare it for further processing.

The system then stores and retrieves relevant data using a vector database (Weaviate), allowing for efficient similarity searches and comparisons with previously analyzed statements. The processed data is then analyzed by two key components: a Generative AI model (Google Gemini) and a Predictive Model (ML-based Disinformation Classifier). The Generative AI model refines the evaluation iteratively, while the predictive model applies machine learning techniques to classify disinformation based on linguistic features and factual consistency.

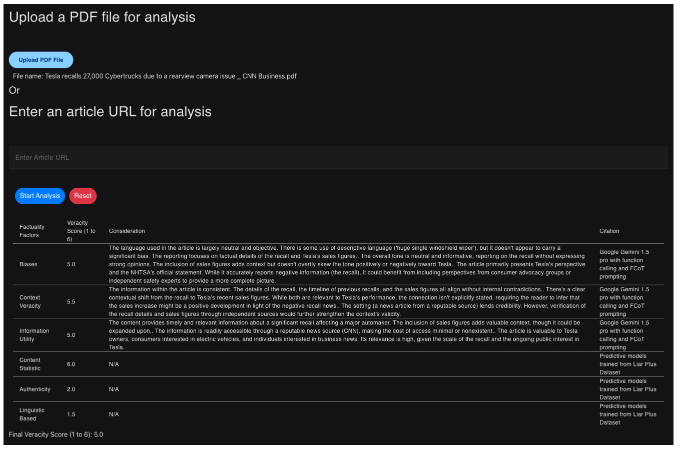
To enhance accuracy, the system integrates web scraping and a search engine API (SerpApi), which gathers additional evidence and contextual information from external sources. These insights are fed into the scoring and decision engine, which aggregates results from both models and assigns a final veracity score. The process concludes with results and report generation, providing users with a comprehensive misinformation assessment based on a combination of predictive and generative AI methodologies.

B. Addressing Limitations

The combination of two models addresses the limitations of using either model independently. Predictive models are effective at evaluating text using predefined features but are limited by the patterns they learned during training. This makes them less effective when encountering complex or novel language. They also lack the ability to adapt dynamically to new contexts. On the other hand, generative models are more flexible and context-aware, but they can sometimes produce unreliable or biased outputs due to inconsistencies in their training data or a lack of grounding in factual information. This combined method is particularly beneficial when detecting misinformation which requires both precise analysis and contextual understanding. By integrating two models, the system compensates for their individual weaknesses, ensuring both a rigorous and adaptive analysis of text.

C. Frontend Interface

To enhance accessibility and usability, the Mesop UI has been integrated into the model, providing a seamless interface for misinformation detection. The UI allows users to upload PDF files or enter article URLs for analysis. Once processed, the interface displays a detailed breakdown of factuality factors alongside veracity scores, considerations, and citations.



Upload a PDF file for analysis

File name: Tesla recalls 27,000 Cybertrucks due to a rearview camera issue _ CNN Business.pdf

Or

Enter an article URL for analysis

Factuality Factors	Veracity Score (1 to 6)	Consideration	Citation
Biases	5.0	The language used in the article is largely neutral and objective. There is some use of descriptive language ('huge single windshield wiper'), but it doesn't appear to carry a significant bias. The reporting focuses on factual details of the recall and Tesla's sales figures.. The overall tone is neutral and informative, reporting on the recall without expressing strong opinions. The inclusion of sales figures adds context but doesn't overtly skew the tone positively or negatively toward Tesla.. The article primarily presents Tesla's perspective and the NHTSA's official statement. While it accurately reports negative information (the recall), it could benefit from including perspectives from consumer advocacy groups or independent safety experts to provide a more complete picture.	Google Gemini 1.5 pro with function calling and FCoT prompting
Context Veracity	5.5	The information within the article is consistent. The details of the recall, the timeline of previous recalls, and the sales figures all align without internal contradictions.. There's a clear contextual shift from the recall to Tesla's recent sales figures. While both are relevant to Tesla's performance, the connection isn't explicitly stated, requiring the reader to infer that the sales increase might be a positive development in light of the negative recall news.. The setting (a news article from a reputable source) lends credibility. However, verification of the recall details and sales figures through independent sources would further strengthen the context's validity.	Google Gemini 1.5 pro with function calling and FCoT prompting
Information Utility	5.0	The content provides timely and relevant information about a significant recall affecting a major automaker. The inclusion of sales figures adds valuable context, though it could be expanded upon.. The information is readily accessible through a reputable news source (CNN), making the cost of access minimal or nonexistent.. The article is valuable to Tesla owners, consumers interested in electric vehicles, and individuals interested in business news. Its relevance is high, given the scale of the recall and the ongoing public interest in Tesla.	Google Gemini 1.5 pro with function calling and FCoT prompting
Content Statistic	6.0	N/A	Predictive models trained from Liar Plus Dataset
Authenticity	2.0	N/A	Predictive models trained from Liar Plus Dataset
Linguistic Based	1.5	N/A	Predictive models trained from Liar Plus Dataset

Final Veracity Score (1 to 6): 5.0

Fig. 5. Frontend Interface Demo

The results include key factuality factors such as biases, context veracity, information utility, content statistics, authenticity, and linguistic-based features. Each factor is assigned a veracity score (ranging from 1 to 6), along with a textual explanation highlighting the strengths and limitations of the analyzed content. Additionally, the citation column provides references to either the Google Gemini model or predictive models trained on the LiarPlus dataset, ensuring transparency in the analysis. The final veracity score offers a comprehensive assessment of the text's reliability, making it easier for users to interpret and act upon the results.

VI. Results

A. Evaluation of Normal CoT vs FCoT

This table compares two evaluation methods, Normal CoT prompting and FCoT (Function Calling + Chain of Thought), in assessing the factuality of an article across different factors. The factors evaluated include biases, context veracity, and information utility, each rated on a scale from 1 to 6.

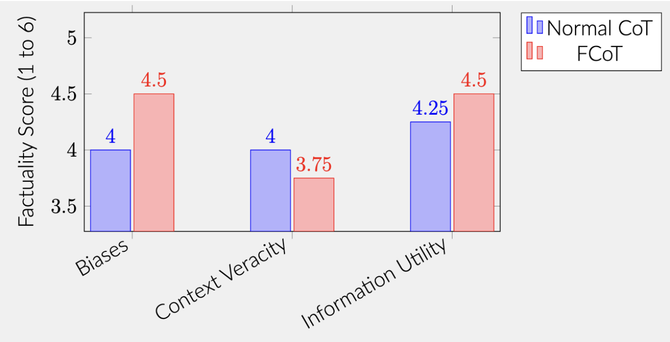


Fig. 6. FCoT vs Normal CoT Experiment Results

For the biases factor, the Normal CoT method assigned a score of 4, while FCoT provided a slightly higher score of 4.5. This indicates that FCoT was more sensitive to subtle biases and missing counterarguments, making it a more critical evaluator in detecting implicit biases within the text.

Regarding context veracity, Normal CoT gave a score of 4, whereas FCoT rated it 3.75. The lower score from FCoT suggests that it penalized unverified claims more strictly and enforced a stricter standard on context consistency. This implies that FCoT places greater emphasis on validating factual accuracy and detecting shifts in contextual integrity.

For information utility, Normal CoT provided a 4.25, while FCoT rated it slightly higher at 4.5. This suggests that FCoT values completeness, particularly considering historical context—such as past recalls—when assessing the usefulness of an article. This makes FCoT a more comprehensive evaluator in determining whether an article provides well-rounded and relevant information.

Overall, the average score across all factors was 4.08 for Normal CoT and 4.25 for FCoT, indicating that both methods found the article to be largely factual. However, FCoT demonstrated greater precision by being more critical of biases, contextual inconsistencies, and missing historical context, making it a more rigorous tool for misinformation detection.

B. Evaluation of Model Result

The evaluation of our misinformation detection model against PolitiFact's expert fact-checking rankings shows a strong overall alignment but with some room for improvement. The model's Mean Absolute Error (MAE) is 1.00, meaning that, on average, its predictions deviate by one ranking point from the ground truth. This suggests that the model is reasonably accurate but could be further refined to improve precision in classifying factuality.

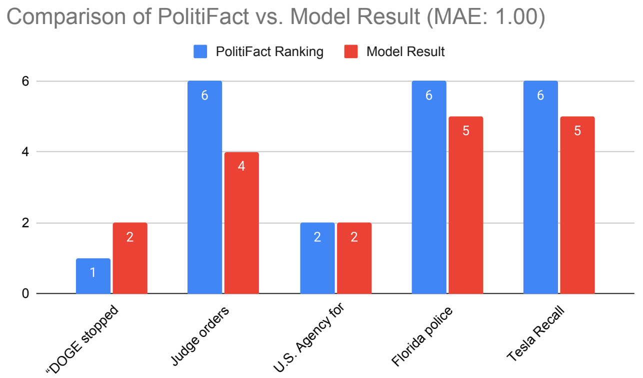


Fig. 7. Comparison of PolitiFact vs. Model Result

The bar chart provides a visual comparison between PolitiFact's rankings (blue bars) and the model's predictions (red bars). A smaller difference between the two bars indicates higher accuracy.

In most cases, the model was closely aligned with PolitiFact, with only minor deviations. For example, in the ”Tesla Recall” and ”Florida Police Arrest” cases, the model slightly underestimated the factuality, assigning a score one point lower than PolitiFact. In the ”DOGE Payment” case, the model overestimated its factuality by assigning a higher ranking than PolitiFact, suggesting it may need stricter penalties for misinformation. The largest discrepancy was observed in ”Judge Orders”, where PolitiFact assigned a ranking of 6, but the model predicted 4. This suggests a need for improved context verification, particularly in policy-related claims.

Overall, the model demonstrates promising performance but requires further refinement in specific areas to enhance its accuracy and reduce its MAE below 1.00.

VII. Future Directions

To further enhance the accuracy and robustness of our misinformation detection model, several key areas for improvement have been identified. These future directions aim to refine the system's capabilities, improve data reliability, and expand its applicability across different platforms.

One significant improvement is the expansion of factuality factors, increasing from the current six factors to twelve. This will allow for a more granular and detailed veracity assessment, capturing a broader range of linguistic, contextual, and credibility-related indicators to enhance the precision of misinformation detection.

Another focus area is the improvement of data retrieval by enhancing the WeaviateDB ranking system. By refining how retrieved information is prioritized and matched to user-provided content, the model can achieve more accurate and contextually relevant comparisons, ensuring better fact-checking performance.

Additionally, we plan to integrate real-time sources by incorporating live fact-checking databases and authoritative news APIs. This will enable the system to cross-check statements against the latest verified information, improving its responsiveness to rapidly evolving misinformation trends.

To further optimize performance, Fractal Chain-of-Thought (FCoT) prompting will be refined to increase model consistency and reliability. Enhancing FCoT will help mitigate inconsistencies in reasoning, leading to more stable and trustworthy veracity assessments.

Finally, efforts will be made to scale the model for deployment by optimizing performance for integration into social media platforms and fact-checking systems. This will allow the misinformation detection system to function at scale, providing real-time insights in digital environments where misinformation spreads most rapidly.

By addressing these areas, the model will continue to evolve into a more accurate, efficient, and scalable misinformation detection tool, contributing to a more informed and responsible information ecosystem.

Appendix A: Proposal

Project Proposal from Fall 2024

Appendix B: Contribution

Yiheng Yuan contributed to documenting the Predictive Model Factuality Factor, focusing on authenticity, model training, and the RAG process for the group report. His research on RIG and RDB helped lay the groundwork for future implementation, while improvements to function calling logic enhanced accuracy. In addition to assisting with presentation slides, he explored methods to improve model reliability and initiated front-end design work. Enhancements to the RAG workflow optimized information retrieval from WeaviateDB. He also conducted an A/B test, comparing Serper.dev API and SerpAPI for search performance while refining WeaviateDB's ranking system.

Luran Zhang played a key role in developing the Predictive Factuality Factor model and supported UI/UX development. Debugging issues in factuality factor functions ensured smoother execution, while contributions to front-end development with Mesop improved overall usability. Mentor feedback was incorporated by enabling individual factuality factor score displays. Further refinements to the predictive model involved restructuring calculations and exploring neural networks as a potential alternative to random forests. To improve accuracy, she collaborated with other groups for testing and evaluated

Streamlit as a possible replacement for Mesop in front-end integration.

Jade Zhou focused on refining factuality factors, particularly in toxicity and linguistic-based scoring. Research on hill climbing contributed to optimization efforts, while analysis of the LIAR-PLUS dataset informed in-context learning. In UI/UX design, she selected key components to enhance user experience and developed the factuality factor analysis page for seamless model integration. Resolving responsiveness issues ensured smooth interactions across devices. The UI integration was finalized through iterative improvements to user flows, design enhancements based on feedback, and cross-platform testing.

Acknowledgment

The authors would like to thank Dr. Ali Arsanjani, Director of Applied AI Engineering at Google Cloud, for his mentorship and guidance throughout this project.

References

[1] Alhindi, T., Petridis, S., Muresan, S. (2018). Proceedings of the First Workshop on Fact Extraction and Verification (Fever), 85–90.

[2] Arsanjani, A. (2023, October 5). Implementation strategies for factuality factors, veracity vectors, and truth tensors. Alternus Vera.

[3] Jiang, B., Tan, Z., Nirmal, A., Liu, H. (2024). Disinformation detection: An evolving challenge in the age of llms. Proceedings of the 2024 SIAM International Conference on Data Mining (SDM), 427–435.

[4] Liu, H., Wang, W., Li, H. (2023). Interpretable multimodal misinformation detection with Logic Reasoning. Findings of the Association for Computational Linguistics: ACL 2023, 9781–9796.

[5] Pastor-Galindo, J., Nespoli, P., Ruipérez-Valiente, J. A. (2024). Large-language-model-powered agent-based framework for misinformation and disinformation research: Opportunities and open challenges. IEEE Security amp; Privacy, 22(3), 24–36.