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

A. Discussion of prior work

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 chatbox, allowing interactive dialogue to transparently present detection results.

B. Description of LIAR-PLUS Dataset

The initial 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.

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.

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 side is a veracity score that is generated based on the predicted probability of each label and applying weights ranging from 0 to 1 to the values. The label "pants on fire" gets the lowest weight and goes up by a 0.2 increment as it gets closer to being completely true, which is assigned 1 as the weight. By multiplying the probability and the weight, we get a veracity score between 0 and 1.

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. 1. 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.

We provide the model with an article divided into three chunks for analysis. Figs.2 and 3 summarized the truthfulness scores generated by the GenAI model using normal prompts and FCoT prompts that clearly illustrate the effectiveness of FCoT. When analyzed with normal prompts, the truthfulness scores varied significantly, ranging from 0.7 to 0.4, a gap of 0.3. In contrast, the FCoT approach produced more consistent truthfulness scores, ranging from 0.51 to 0.62. This consistency better aligns with human detection patterns, highlighting FCoT's capability to improve the model's reliability and cohesiveness in analyzing content.

Comparison of Truthfulness Scores: Normal Prompt vs. Fractal Chain of Thought (FCoT)

	chunk1	chunk2	chunk3
Nomal Promt	0.65	0.4	0.7
FCoT Prompt	0.51	0.58	0.62

Fig. 2. Comparison of Truthfulness Scores

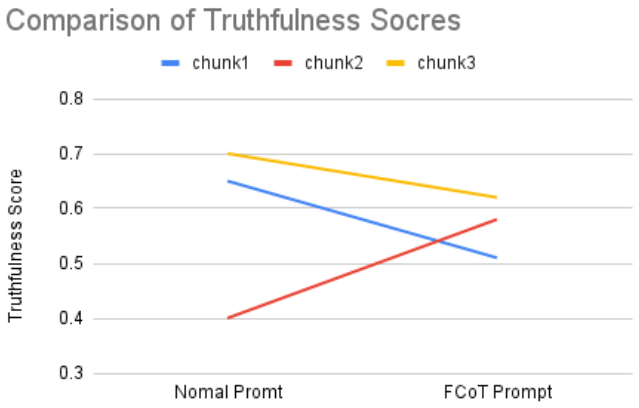

Fig. 3. Comparison of Truthfulness Socres Chart

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. The predictive model uses feature engineering techniques to analyze various linguistic attributes, such as sentiment, readability, and named entity recognition. Meanwhile, the generative AI model iteratively refines its evaluation to produce a robust and nuanced analysis.This approach ensures a comprehensive analysis of text by leveraging the unique strengths of each model type.

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 make this advanced technology accessible to a broader audience, the Mesop UI has been integrated into the model. This frontend interface simplifies the process, enabling users to upload their files, view detailed analyses of extracted text, and receive comprehensive reports summarizing the findings. The reports provides key insights, such as potential misinformation, linguistic patterns, and structural anomalies, making it easier for users to understand and act on the results.

VI. Future Directions

For the remainder of this quarter, we will focus on refining and enhancing our misinformation detection model by addressing existing limitations and improving overall performance. Our priority is to finalize the integration of all 12 factuality factors, ensuring a more robust and comprehensive evaluation system. To optimize data retrieval, we will continue refining the ranking mechanism in WeaviateDB and explore alternative indexing strategies to improve the accuracy and relevance of retrieved information.

Additionally, we will enhance the predictive model, evaluating the effectiveness of neural networks compared to random forests and implementing the best-performing approach. Improvements to function calling logic will further refine trigger precision in our model. On the UI/UX side, we aim to complete the front-end interface, ensuring a seamless user experience with well-integrated factuality factor analysis and intuitive interactions. Testing and debugging efforts will focus on optimizing responsiveness across different devices.

We will also conduct structured A/B testing between Serper.dev API and SerpAPI to determine which provides better search results for misinformation detection. If results indicate a significant improvement, we will transition to the superior API. Lastly, we will perform comprehensive system evaluations, refining our scoring mechanisms and ensuring fair weight distribution between predictive and generative components. These efforts aim to deliver a more accurate, efficient, and scalable misinformation detection system by the end of the quarter.

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.