Chenly Insights: The Misinformation and Veracity Engine

Calvin Nguyen
can005@ucsd.edu

Samantha Lin
yul186@ucsd.edu

Dr. Ali Arsanjani
email@ucsd.edu

Abstract

The objective of our product, Chenly Insights, is to help users combat misinformation using factual factors and microfactors. As misinformation gets easier to spread due to deep fakes and large social networks, this project empowers users by giving them a veracity (truthness) for the news they receive through a Mesop interface. A user will link and upload a news article of their choice. Then, two different types of AI, generative and predictive, judge the article on appropriate factuality factors. The scores are accumulated and then shown to the user, along with other visualizations related to the article. In addition, these scores, along with metadata related to the article, will be uploaded to a vector database that generative AI can use. This vector database is constantly updated with new information from fact check and news websites. Users can also converse with the AI and ask questions about each article and why a particular score was given. We also found the best prompting types with adjustments to the prompts that gives the highest accuracy.

Website: TBA
Website Design: https://shorturl.at/HtUD6
Website Flow: https://shorturl.at/XnonU
Code: https://github.com/Neniflight/Q1-B01-Team-CS

1 Introduction

Modern technological breakthroughs in large language models have leveraged the non-negligible issue of creating and publicizing misinformation, spreading false information without intent, and disinformation, spreading false information intent to mislead people, to a new level. LLMs like ChatGPT can generate context with a given prompt in an extremely speedy fashion with very little cost, and this aids disinformation spreaders in efficiently generating and disseminating human-like text that makes it hard for LLMs and humans to distinguish between true and false statements (. This can lead to major issues within our society including intensifying polarized stances between people, masking the truth of an event, and more. With this in mind, the goal of our project is to provide a scoring system that breaks down the article and justifies them with multiple factors to help users identify the truthfulness of a given text.

The issue of misinformation and disinformation is not a new topic facing the community. Researchers have approached this issue with different methods, from tuning RoBERTa models to training LLMs to detect more advanced disinformation, and the biggest breakthrough is the use of deep learning to evaluate the content of the text. In the paper "Disinformation Detection: An Evolving Challenge in the Age of LLMs", Jiang et al. (2024) discovered that a fine-tuned RoBERTa model was able to detect simple LLM-generated disinformation created by LLMs, but it was unable to detect complex false context such as combining human-written false information and LLM generated context and LLM context generated by using the chain of thought prompting technique. However, the research group has discovered that guiding LLMs with a chain of thought prompts allows LLMs to detect disinformation at a much higher level of accuracy. In summation, the paper suggests that existing detection techniques still struggle to detect disinformation, but some directions we can take to combat misinformation and disinformation in the future are to guide LLMs with chain-of-thought prompts and focus on highlighting the significance of meaning and contextual elements.

Therefore, the goal of our project is to build an advanced system that utilizes specialized AI and models to detect, rank, and address misinformation automatically. The system will use over 15 comprehensive factuality factors to analyze information. We will make the process for checking for misinformation as smooth as possible for users and help people access trustworthy and accurate content. Using this tool, users will get a more nuanced, balanced, and credible news. We can help improve public discourse, support democratic processes, and enhance our public health and safety. Additionally, this system, if successful, can be implemented into many different social media platforms, which allow these platforms to shutdown false information effectively. This is an addition to our progress from Quarter 1's project.

2 Methods

To build Chenly Insights, we received guidance from our mentor to combine predictive and generative AI onto a Mesop interface. These models will work in tandem to compile multiple different scores of factuality factors into a veracity score. Our group was assigned 4 factuality factors to work on: Naive Realism, Poltiical Stance, Social Credibility, Sensationalism. He recommended that the more information and context you give a generative AI model, the better the results become. Therefore, we also implemented function calling, the SERP API, and a vector database. Each subsection will discuss the methods used for each model type and the adjustment added.

2.1 Predictive AI

For our predictive AI model, we applied traditional machine learning techniques to the Liar Plus dataset Tariq60 (2018). This dataset consists of a statement, a truthfulness label, and metadata surrounding that statement. The goal was to predict the truthfulness label of an article (from a scale of 1-6) based on microfactors extracted from each factuality factor. For example, one key factuality factor was ”Naive Realism”. The microfactors related to this factor were perspective analysis, dissenting view checks, and isolation analysis. This was operationalized by creating confidence, subjectivity, and reputation columns respectively, created from using HuggingFace models and textblob. If a statement exhibited high subjectivity and confidence, it is likely to have been viewed as the correct perspective while dismissing other points of view. From there, we test out multiple different machine learning models of varying complexities and evaluate each model based on accuracy. Each factuality factor has its own machine learning model and its own derived microfactors. This process ensures each factuality factor has an accurate model associated with it.

When we incorporate each model into the Mesop interface, we extract the appropriate metadata to create scores for each microfactor. The article’s title is used as the ”statement” like from the Liar Plus dataset.

2.2 Vector Databases

To help supply our generative AI model with more context, we utilized a vector database, as its optimal for similarity searches. We employed ChromaDB, an open-source vector database, to store and manage high-dimensional word embedding extracted from article headers and the Liar Plus dataset. Within each collection, we store the documents (statements) and metadata (label, speaker, justification). This database is hosted within a persistent Docker container to ensure data retention. To ensure up-to-date information, we periodically update the vector database by scraping recent statements and promises from Politifact, Washington Post Fact Checker, and Snopes. Using specific HTML parsing for each website, we extract the relevant metadat to input into the database.

When implementing the vector database into Mesop, we extract the title from the user-inputted article. Then, we perform a vector database query to find the top 3 similar statements. The similar statements and its metadata are provided to the generative AI model as a prompt.

2.3 Serp API

Another way we provide context for the generative AI model is through the Serp (Google Search) API. After extracting the title from the article, we search for related articles about the topic using the Serp API. The retrieved articles are processed using Newspaper3k to extract metadata (title, author, summary, publication date). The articles and its metadata are feed as the ”ground truth” to the generative AI in relation to a topic to refine its evaluations.

2.4 Function Calling

Function calling enables generative AI to execute predefined tasks beyond its typical capabilities. In our system, we useed function calling to integrate predictive AI models for factuality factor analysis. Certain microfactors, such as emotion analysis, are better suited to predictive Ai due to the availability of specialized models like HuggingFace emotion analyzers. These functions, defined in separate Python files, were registered within the AI model declaration, allowing seamless integration between generative and predictive AI components.

However, function calling has proved to be difficult, as it is inconsistent. Sometimes, it would not call the function at all or the generative AI will ignore the function call result all together. We could transition to using agents or phasing out function calling entirely.

2.5 Generative AI

As recommended by our mentor, we utilized Gemini 1.5 Pro 002. We use generative AI in two ways:

1. Extracting key attributes from user-uploaded PDFs (e.g., speaker, context, title) to support predictive AI modeling.
2. Evaluating sensationalism and other factuality factors via structured prompts.

There are three different prompting techniques we could potentially use to evaluate facuality factors: normal, Chain of Thought (CoT), and Fractal Chain of Thought (FCoT). Each prompt consists of grade objective functions from 1-6, defined by microfactors of the factuality factors, and asking the model to grade an article based on the factuality factor from a scale of 1-6. CoT prompting involves asking the model to explain its reasoning, causing it to think more about its response and getting more accurate response. FCoT prompting asks the model to go through three iterations to explain its reasoning. After each iteration, we ask the model what it missed on its previous iterations. With each prompting type, we could also add combinations of querying the vector database, calling the Serp API, or performing a function call.

There are three different types of prompting with 8 different adjustment combinations of vector database, Serp, and function calling. For our Mesop interface, we want to offer the best option for the prompting type and adjustment combination. To determine the best options, we selected four different types of articles and asked four different people to grade the article based on a factuality factor. The article's true factuality factor score is based on the mean of the scores given by each grader. Then, each prompting type and adjustment combination is used on each article. Whichever article has the lowest mean absolute error is declared the best.

2.6 Website

For our website, we wanted our designs to look seamless, simple, trustworthy, and good. Additionally, we, not only wanted users to detect misinformation, but learn more about the pipeline, about this project, and to test out their own prompts as well. To design all of this properly, we utilized Figma to organize our ideas and design the website. We decided on the name Chenly Insights, as it combines the words for truth in Vietnamese and Chinese. The blue/green color theme communicates truth and accuracy.

3 Results

.

4 Conclusion

5 Appendix

5.1 Introduction

5.1.1 Broad Problem Statement

Misinformation, false or misleading information, spreads faster and farther today than ever before, due to social media and the internet. According to **?**, a research expert at Statista, 38.2% of Americans accidentally shared fake news. This means one of you could have unknowingly spread false information. Though misinformation may seem harmless at first, it has serious consequences for our society.

It can mislead people, causing them to make poor decisions based on falsehoods. A real-world example of this was during the COVID-19 pandemic. **?** from NIH said that misinformation regarding to incidence rate and spread of the virus ”contributed significantly” towards the complacent attitude of people, causing more fatalities than needed. Misinformation can be also used to divide communities. Oftentimes, politicians can create ”us vs. them” scenarios, deepening divisions within our society. Especially during election cycles, misinformation can shape public opinion on candidates and sway votes against certain candidates.

Though there are existing tools to fight against misinformation, such as Politifact and Snopes sites, they do not contain all types of misinformation people can spread. After all, there are only so many people working on these websites and working to debunk statements. Additionally, many people are not likely to look through these fact-checker sites to determine whether the news they share contains misinformation.

Therefore, the goal of our project is to build an advanced system that utilizes specialized AI and models to detect, rank, and address misinformation automatically. The system will use over 10 comprehensive factuality factors to analyze information. We will make the process for checking for misinformation as smooth as possible for users and help people determine whether the sources and their information are nuanced, balanced, and credible.

With this system, we believe we can help improve public discourse, support democratic processes, and enhance our public health and safety.

5.1.2 Narrow Problem Statement

Previous methods to fight against misinformation often focused on specific forms of misinformation. From our readings from our mentor, these are ways people are using to counter against misinformation:

- SNIFFER: Multimodal Large Language Model for Explainable Out-of-Context Misinformation Detection
- Disinformation Detection: An Evolving Challenge in the Age of LLMs
- PolitiFact

For example, **?** created SNIFFER, a multimodal large language model for **out-of-context misinformation** detection, to fight against instances where a caption does not match the context of an image. The model is able to utilize up-to-date information through external checking and explain the reasoning behind its judgment. However, images and captions are only one way an article can use misinformation. Misinformation can be manipulated content through AI or image editing, clickbait through exaggerated titles, completely fabricated news stories, and more. Existing methods are too narrow to capture all types of misinformation, so a user has to use multiple different models to determine whether different aspects of the article are misinformation. People like Jiang et al. (2024) have fine-tuned RoBERTa models to detect more advanced disinformation that is more general to text. However, it still cannot detect complex false contexts that are generated by combining human information and LLM-generated content.

From past work, we've concluded that we need to focus on combining multiple aspects to see improvements in the detection of misinformation. That being said, we need to capture all types of misinformation (in different types of media) by using factuality factors and veracity vectors, collect and utilize up-to-date information, and design specific prompting to apply it to our generative model to detect complex false contexts.

Currently, Our misinformation detective system can analyze text with four factuality factor checkers that use both Predictive AI (deep neural networks and transformers) and Generative AI (Google Gemini). Additionally, we have also created a chromadb vector database to store our web scraped information from the Politifact Website, and we've also implemented real-time Google searches with SerpAPI to ensure we're also utilizing up-to-date information when analyzing the text. Lastly, After testing the mean absolute error of different prompting techniques and different additional information combinations, we've narrowed down to the conclusion that the best combination that will be selected as our default prompt to feed into the Generative AI is the combination of normal prompting with the top three most similar item from our chromadb vector database and serpAPI search results. However, we plan to keep our system flexible to users by keeping all other options open to the users to select by themselves through toggle buttons and dropdown menus so they can select their preferred way of evaluating the text they want to check. Additionally, we're also working on completing a final page with a table containing factuality factors, their score, the reasoning for the score assigned, and the sources used for a more detailed description for the users to evaluate. Finally, we've also completed a final UX design for our webpage and will be looking to implement it over the next week.

5.2 Proposed Methodology

The following tools will be used in our system. All of these technologies will be stored in one repository shared within our whole capstone group. A docker image will be used to ensure our technology is replicable, as maintaining an environment yml file has been difficult. Over the past couple of weeks, we have worked on the following:

Table 1: Technologies Used in this Project

Category	**Technologies**
Languages, Frameworks	Python, Mesop, CSS
Machine Learning	Google Gemini, Transformers, Deep Neural Networks, Decision Trees,Pytorch
Prompting	Function Calling, Fractal Chain of Thought
Databases	ChromaDB, Docker
Data Collection	BeautifulSoup, SerpAPI
Data Sources	Liar Plus Dataset, PolitiFact, Snopes
Evaluation	Accuracy, Precision, Recall, Mean Absolute Error
Factuality Factors	Stance Detection, Social Credibility, Sensationalism, Naive Realism, Neural Misinformation, NodeRank, Stance Features, Title vs Body

- Tested all combinations of prompting and information for Generative AI, and concluded that normal prompting with the top three most similar item from our chromadb vector database and serpAPI search results gives the smallest error.
- UI/UX design of our final system webpage with Figma

For the upcoming weeks, we will focus on implementing the following:

- Implementing the UI/UX design and add quality of life (QOL) features to our Mesop interface
- completing the table showing the scores and evaluation from Gemini in our final page
- combining work across multiple groups in our capstone group

We've taken a detour from our original plan to add more factuality factors and automate our project, and as a capstone group, we've decided to work on testing our prompts with evaluation criteria and organize the results, adjust the prompts to get better results, and also focus more on the UI/UX part of our final product. Although the detour wasn't originally planned during the last quarter, after talking with our mentor, we are all clear on the direction we're taking and what to expect to complete for our final product. We believe that we will be able to complete all adjustments to prompts, the implementation of UI/UX of our website, and combine every group's work into a repository within the next couple of weeks.

5.3 Project Outputs and Deliverables

We will create a poster, a report, and a website (will be non-locally hosted if time permits).

The poster will consist of an introduction to the topic: misinformation, the workflow diagram of our project on how we arrive at the final veracity score assigned to user-inputted text, a method section including information and evaluation (accuracy and how we got there) for our predictive AI and Generative AI models, a data section including the datasets we've used, the websites we've scrapped, and explanation of Serp API search results, a screenshot of our system showing the user interface, and lastly, a table showing the accuracy of detecting misinformation, scores of factuality factors produced by our system, evaluation (response) from gemini, and the sources that were used in each evaluation.

The report will include sections: Introduction, Methodology, Results, Conclusion, and References. In the introduction section, we will introduce our topic, what has been done, why is our topic important, and what data we will be using throughout the project. In the Methodology section, we will introduce our approach to resolving the problem in greater detail including sections to introduce our models for the Predictive AI portion of the project, the prompting we used as default, and other prompting techniques we've used throughout when we were working through the project, and how we combine the two along with other functions such as scraping, SERP API, and function calling that helps our system generate a final veracity score. In the results section, we will include a table of the accuracy of the detection of misinformation and tables that include the scores generated by different prompting methods, evaluations, and sources. In the conclusion section, we will summarize our paper and discuss possible approaches or improvements for future work on detecting misinformation.

The website will be the primary output, and it consists of multiple sections. There will be a homepage where users land and learn about our project, an about page where we discuss the technologies we've used and our research paper, and a pipeline page to explain our project with a flow chart. To get started with the misinformation detection system, the user will have to click on the "try it out" button on the home page, then they will be navigated to a page where they select their desired prompting (or default) techniques, upload a pdf or a link of their chosen article. Lastly, after the system runs, a score will be presented, and the user will have the option to click on a "Deep Analysis" button to check out the basic information of the inputted text and a table with factuality factors, their scores, and the evaluation to show how the score is decided, and at the bottom, there will be a chat box that allows users to chat with Google Gemini.

5.4 Data Justification

Some of the data we will be using includes: Liar Plus dataset, Politifact website, Snopes website, and Serp API search results.

The first data we will be using is the Liar Plus Dataset from Tariq60 (2018). This dataset is obtained by "extracting articles written by journalists from the PolitiFact website", a "Pulitzer Prize-winning site" (Tariq, 2018) & (University of California Berkeley Library, 2024). That being said, we're confident to say that this website is reliable and the data we've obtained is of great quality. Furthermore, the author of the dataset has published a paper using this dataset, proving the quality of the data to be great!

Looking into the data, this data consists of both training and testing data in json format that can be loaded as a pandas data frame. There are a total of 14 columns in this dataset, and the descriptions are as follows:

- Column 1: the ID of the statement
- Column 2: the label.
- Column 3: the statement.
- Column 4: the subject(s).
- Column 5: the speaker.
- Column 6: the speaker's job title.
- Column 7: the state info.
- Column 8: the party affiliation.
- Columns 9-13: the total credit history count, including the current statement.
  - 9: barely true counts.
  - 10: false counts.
  - 11: half true counts.
  - 12: mostly true counts.
  - 13: pants on fire counts.
- Column 14: the context (venue / location of the speech or statement).
- Column 15: the extracted justification

$Tariq, 2018$

This data includes different attributes that could affect a text's truthfulness as well as the label column to categorize the truthfulness of the statement (col.3). We will be utilizing the attributes in this data to build our predictive AI models as well as using the labels to calculate the accuracy of our models. Furthermore, we will also be using the labels to determine the accuracy of our system overall by inserting the statement into our system and evaluating the overall validation score it generates.

The Second data we will be using is the information we obtain from the Politifact fact check Website and Politifact truth-o-meter website by web Scraping. The Politifact website is the same source as the Liar Plus dataset, therefore, we believe the data we scrape from this website will also be of great quality. We have successfully scraped and obtained data such as speaker, statement, label, and context from both websites, and this information has also been successfully added to our docker container and utilized in the generative AI prompting portion of our project. We will be continuously updating the scraped information by running the script manually.

The third data we will be using is the information from the snopes fact check website. This is a website that was claimed to be ”The definitive Internet reference source for urban legends, folklore, myths, rumors, and misinformation” by the University of California Berkeley Library. Therefore, we believe that this website also contains reliable and great-quality data. This website contains similar information as the PolitiFact website including: author, statement, and labels. We will be scrapping this information from the Snopes website and utilizing it the same way as we utilize the information we scraped from the Politifact website.

The last data we will be using is the SERP API search results. This will be data we obtain by using SERP API web-searching in our project. We have been able to successfully implement the function and utilize it in our quarter 1 project as one of our prompting techniques. To go into details, this function takes in the user’s inputted text and uses keywords to retrieve the next 10 search results from the web. Then from the search results, we were able to get the title, author, summary, full text, publish date, and source in a list with dictionaries for each search result. However, web searching doesn’t necessarily always give us good-quality data, so we will be testing if it helps improve our system’s performance and we will be tuning the amount of information we will be using in our prompting if needed.

6 Contributions

Samantha

- Work on the initial modification to our UI/UX design for our mesop interface
- Read and ask one person to read articles to completing human scoring articles in order to evaluate our prompting combinations against human’s judgment
- Started working on creating the table on our final page with factuality factors, scores, evaluations, and sources but ran into an API issue limit issue, so wasn’t able to continue with the coding portion until the middle of week 5
- Work with Calvin together on lo-fi and hi-fi designs of our final product on Figma

- Work with Calvin together to build the pages for our final mesop interface
  - About Page, More Info Page, Scoring page, loading page before scoring page

Calvin

- Worked with Samantha on lo-fi and hi-fi designs of Chenly Insights on Figma
- Read and ask one person to read articles to completing human scoring articles in order to evaluate our prompting combinations against human's judgment
- Determine the best prompting type with proper adjustments through analysis
- Building UI for home page, pipeline explanation, and customization pages.

References

Jiang, Bohan, Zhen Tan, Ayushi Nirmal, and Huan Liu. 2024. "Disinformation Detection: An Evolving Challenge in the Age of LLMs." *arXiv preprint arXiv:2309.15847*. [Link]

Tariq60. 2018. "LIAR-PLUS." Oct. [Link]