THE ROLE OF LLMS IN CURTAILING THE SPREAD OF DISINFORMATION

by

Diksha Saxena

August 2024

A thesis submitted to the
Faculty of the Graduate School of
the University at Buffalo, State University of New York
in partial fulfilment of the requirements for the
degree of

Master of Science

Department of Computer Science and Engineering

Acknowledgments

I would like to express my deepest gratitude to Professor David Doermann for his invaluable guidance, support, and encouragement throughout this research. His expertise and insights were instrumental in shaping this thesis, and his mentorship has been a source of inspiration.

I am profoundly grateful to Professor Rohini Srihari for her continuous support, insightful feedback, and for providing me the opportunity to work closely with her esteemed research lab. The collaborative environment and the innovative spirit of her lab significantly contributed to the development and success of this project. I extend my heartfelt thanks to all the members of Professor Srihari's research lab for their assistance, constructive discussions, and unwavering support throughout this journey.

This work would not have been possible without the combined efforts and support of Professor Doermann, Professor Srihari, and her research team. Thank you all for your encouragement and for helping me achieve this milestone.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis is about understanding the artistry of digital deception. It seeks to discern the distinctions between Recontextualized Social Media Posts, Consistent Human Commentary and News Agency posts, particularly in cases devoid of image manipulations. If the former, then pinpoint the nature and locus of such recontextualization. The focus is on political propaganda surrounding the Russia-Ukraine War Conflict, examining factors beyond factual misquotations. It is essential to analyze the intentions and narratives behind the content posted in connection with an article, as they are integral aspects considered in the classification process. This research aims to contribute a formal and comprehensive analysis of the mechanisms influencing the classification of social media content within the context of political conflicts.

Chapter 1

Introduction

1.1 Digital Disinformation

In the present landscape, misinformation—defined as false or misleading information shared on social media platforms, spreads more rapidly than ever, fueled by the viral nature of social media and algorithm-driven content feeds that prioritize user engagement over factual accuracy.

1.1.1 Digital Misinformation vs Digital Disinformation

Digital disinformation and digital misinformation represent two distinct challenges in the realm of online information. Digital disinformation refers to false or misleading information deliberately created and spread with malicious intent to deceive or manipulate the public. This includes fake news stories, doctored images, and fabricated statistics designed to sway opinions, cause confusion, or influence outcomes such as elections or public health. In contrast, digital misinformation involves the spread of incorrect information without the intent to deceive. This often results from misunderstandings, errors, or ignorance. Addressing digital disinformation typically requires targeted counter-disinformation efforts, robust fact-checking, and educational initiatives to improve digital literacy. On the other hand,

combating digital misinformation focuses on correcting inaccuracies and raising public awareness. Understanding these distinctions is crucial for developing effective strategies to manage and mitigate the impact of false information online.

1.2 The Spread of Digital Disinformation

Digital Disinformation has become a critical issue in today's globally connected society, exacerbated by the rapid expansion of social media platforms and the ease with which content can be shared online.

This rampant spread is not just a product of technology, but also of human psychology. Cognitive biases, such as confirmation bias, and the tendency for individuals to consume and share information that aligns with their pre-existing beliefs, contribute significantly to the spread of misinformation. Echo chambers created by social media algorithms further reinforce these biases, exposing users predominantly to information that affirms their views and isolating them from opposing perspectives.

1.2.1 Disinformation in Political Propaganda

In addition to these general factors, digital disinformation has become a powerful tool in the arena of political propaganda. Political actors, state-sponsored entities, and other interest groups exploit the rapid dissemination of information on digital platforms to spread propaganda that can shape public opinion, discredit opponents, and influence electoral outcomes. This type of misinformation is often strategically crafted to be emotionally compelling, making it more likely to be shared and believed, thereby deepening its impact.

In the realm of political propaganda, misinformation is deliberately deployed to manipulate perceptions, often with far-reaching consequences. State actors and political organizations use misinformation campaigns to create and sustain narratives that serve their strategic interests. These campaigns can range from subtle misinformation, which slightly

distorts the truth, to blatant disinformation, where entirely fabricated content is presented as fact.

The consequences of this politically motivated misinformation are profound. It can erode public trust in institutions, polarize societies, and destabilize democratic processes. The recent conflict between Russia and Ukraine is a poignant example, where both sides have engaged in information warfare on digital platforms, each seeking to control the narrative and sway global opinion in their favor.

Addressing the spread of digital misinformation, particularly in the context of political propaganda, requires comprehensive strategies. Technological tools must be developed and refined to detect and counteract false narratives quickly. Moreover, there is a need for enhanced digital literacy programs that educate the public on how to critically evaluate the information they encounter online. Regulatory frameworks must also evolve to hold platforms accountable for the content they amplify, ensuring that the public discourse remains informed by facts rather than falsehoods.

This study delves into the current dynamics of digital misinformation, with a particular focus on its role in political propaganda, exploring the societal impacts and outlining potential strategies for mitigation in an age where information flows more freely and quickly than ever before.

1.3 Political Disinformation

The spread of misinformation on social media platforms has become a key tool in political propaganda, with significant implications for democratic processes and public trust in governance. Political misinformation, often disseminated deliberately, is designed to manipulate public opinion, discredit opponents, and influence election outcomes. The ability of such content to reach vast audiences quickly, coupled with the emotionally charged nature of political discourse, makes social media an effective but dangerous vehicle for

propaganda.

1.3.1 Mechanisms

Political propaganda on social media is spread through various mechanisms, each designed to manipulate public perception and influence political outcomes:

- **Disinformation Campaigns:** These are coordinated efforts aimed at disseminating false or misleading information to deceive the public. State actors, political groups, and independent operatives use social media to craft and promote narratives that serve their interests. Often, these campaigns employ bots, trolls, and fake accounts to amplify their messages, creating a false sense of widespread support or opposition.

- **Deepfakes and Synthetic Media:** Technological advancements have made it easier to produce highly realistic but entirely fabricated images, videos, and audio clips. Deepfakes—videos manipulated to make it appear as though someone said or did something they did not—are increasingly used in political propaganda to discredit or smear public figures and candidates.

- **Echo Chambers and Filter Bubbles:** Social media algorithms are designed to present users with content that aligns with their existing beliefs and interests. This leads to the formation of echo chambers, where individuals are exposed only to information that reinforces their views, allowing misinformation to spread unchecked. In these environments, users are less likely to encounter opposing viewpoints or fact-check the content they consume.

- **Memes and Simplified Messages:** Memes, which are easily shareable and often humorous, have become a potent tool for spreading political misinformation. By reducing complex issues to simple, bite-sized messages, memes can distort facts and propagate misleading narratives that quickly gain traction and are widely shared by users.

1.3.2 Detection Imperative

The rapid proliferation of political propaganda on digital social media platforms poses a significant threat to global stability and the integrity of democratic processes. The ability to quickly and effectively detect such propaganda is crucial for maintaining informed public discourse, preventing the spread of harmful misinformation, and protecting the integrity of elections and public opinion. As social media continues to be a primary source of information for millions of people worldwide, the need to implement robust detection mechanisms for political propaganda has never been more urgent.

1.4 Russia-Ukraine War

The ongoing Russia-Ukraine war has underscored the critical importance of detecting and combating political propaganda on social media. Since the conflict began, both state and non-state actors have used social media platforms to spread misinformation, disinformation, and propaganda to influence public perception, both domestically and internationally.

Russian Propaganda Campaigns

Russia has been particularly adept at using social media to advance its geopolitical agenda. Through a combination of state-sponsored disinformation campaigns, the use of bots and trolls, and the manipulation of algorithms, Russia has been able to spread narratives that support its actions in Ukraine, undermine international support for Ukraine, and sow discord among Western nations. The spread of fake news, doctored videos, and misleading narratives has had a profound impact on global public opinion, making it difficult to discern the truth in the fog of war.

The Role of Ukrainian Counter

Ukraine, in response, has also engaged in its own counter-propaganda efforts. Leveraging social media, the Ukrainian government and its allies have sought to counteract Russian narratives by promoting pro-Ukrainian content, sharing real-time updates from the front-lines, and highlighting Russian atrocities. While these efforts have been crucial in rallying international support, they also contribute to the complex information landscape where propaganda and factual reporting are intertwined.

Impact on Global Perception

The propaganda war between Russia and Ukraine has had significant implications for global perception of the conflict. In regions where access to independent media is limited, social media often serves as the primary source of information. This makes the detection of propaganda on these platforms essential to ensure that global audiences are not misled by false narratives. Failure to detect and mitigate the spread of propaganda can lead to a distorted understanding of the conflict, influencing foreign policy decisions, humanitarian responses, and public opinion.

1.5 DARPA Disinformation Curtail Program

1.5.1 Semantic Forensics

SemaFor, a program sponsored by DARPA, developed various tasks aimed at tackling digital disinformation spread. Semantic Forensics (**SemaFor**), [16] led by Dr. Wil Corvey, addresses the challenges of detecting manipulated media as traditional statistical methods become inadequate. As media generation technologies advance, they often produce semantic inconsistencies—such as mismatched earrings in GAN-generated images—that can be exploited to identify falsified content.

SemaFor aims to develop advanced semantic detection technologies to:

- Detect if media has been generated or altered.
- Attribute media to specific sources.
- Characterize the intent behind media manipulation.

DARPA supports this through:

- **Analytic Catalog**: An open-source repository of SemaFor-developed resources for ongoing use and development.
- **AI FORCE**: A research challenge to create models that distinguish between real, manipulated, and fully synthetic AI-generated images.

1.5.2 Task 4.1.2

In Task 4.1.2 - Social Media Image Recontextualization, performers are tasked with detecting the type of recontextualization that occurred in a social media probe. For recontextualized images, these will be sourced from a Reference News Article, with the associated social media post classifying, labeling, or stating an inaccurate representation of the article's content, tailored to a specific narrative. The task was to determine whether the social media post represents a recontextualization of the image/information or is consistent with the provided Reference News Article.

In this thesis, evaluation submissions made to SemaFor for Task 4.1.2, which focused on detecting, localizing, and classifying recontextualized social media content derived from news articles are listed under experiments. The technical solutions were established to address the specific problem statement of this task. This solution is adaptable to broader scenarios and can be extended beyond its initial scope.

Table 1.1 provides an overview of the experimental and ablation tasks for this evaluation period as represented in the Gym.

Table 1.1: Overview of Experimental and Ablation Tasks

Task	Description
4.1.2	Social Media Image Recontextualization
4.1.2a	Recontextualized Detection
4.1.2b	Recontextualization Localization
4.1.2c	Recontextualization Type

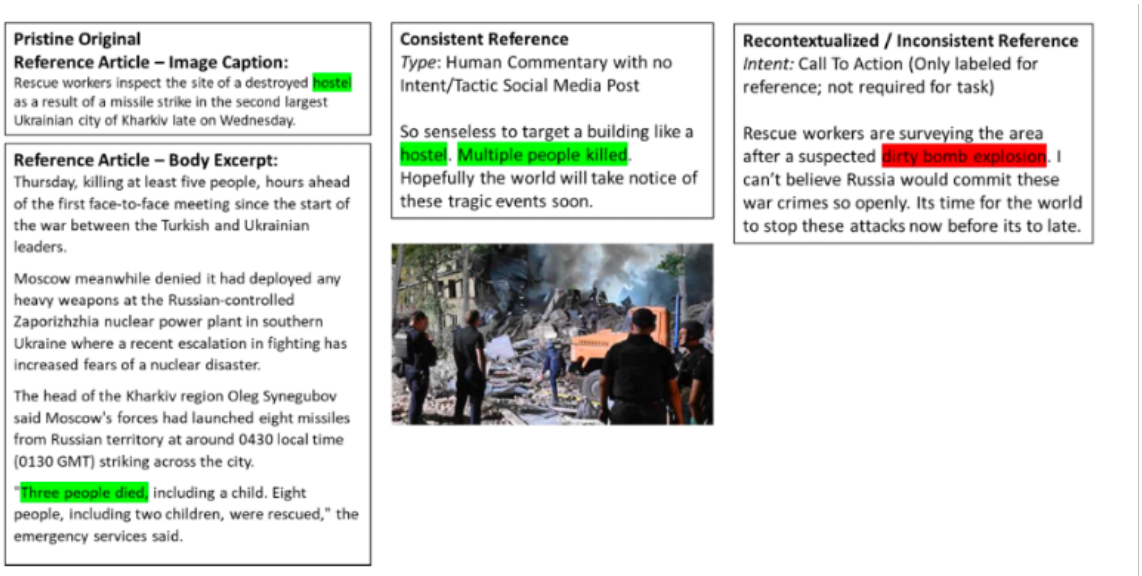


Figure 1.1: An example of Article, Consistent and Inconsistent Post

Task A - Recontextualization Detection

In Task 4.1.2.a, performers are required to determine if a related social media post is recontextualizing the original source media's image and/or information or if it is consistent with the provided Reference News Article.

Evaluating both the social media post and the Reference News Article is vital. In some cases, the evaluation might be possible by examining the social media post alone. However, in other cases, it may be necessary to analyze the Reference News Article to ascertain if the social media post recontextualizes the original content.

Task B - Recontextualization Localization

In Task 4.1.2.b, performers must identify the specific line of text within a recontextualized social media post that represents the manipulated recontextualization, with reference to the provided article.

Performers are required to pinpoint which sentence in the social media post contains the recontextualized information. This determination is made based on the information in the Reference News Article and the provided input evidence graph. Only one sentence within the social media post will contain the recontextualization.

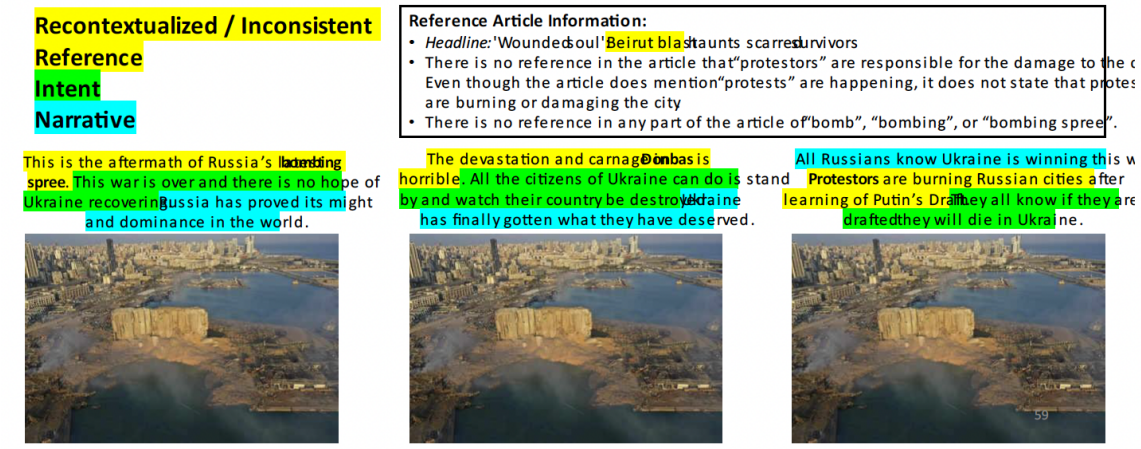


Figure 11. Example of Task 4.1.2.b with (in)consistency (yellow), intent (green),and narrative (blue) localizations annotated.

Figure 1.2: Example of a social media post deconstructed

Task C - Recontextualization Classification

In Task 4.1.2.c, performers must determine the type of recontextualization in a given social media post relative to a reference article. Performers should identify which type of recontextualization occurred based on the three specified types: Event, Location, and Individual/Group.

The three recontextualization types are:

- **Event**: Mischaracterization of an event such as a bombing, explosion, riot, or protest.
- **Location**: Mischaracterization of a specific area or place, such as a country, city, or a specific location within a city (e.g., a hospital).
- **Individual/Group**: Inaccurate representation of individuals or groups, such as politicians, protestors, or civilians.

Each social media post will contain only a single type of recontextualization. Performers will select one of the three provided types based on the content of the post. Figure 1.3 below illustrates examples of each recontextualization type.

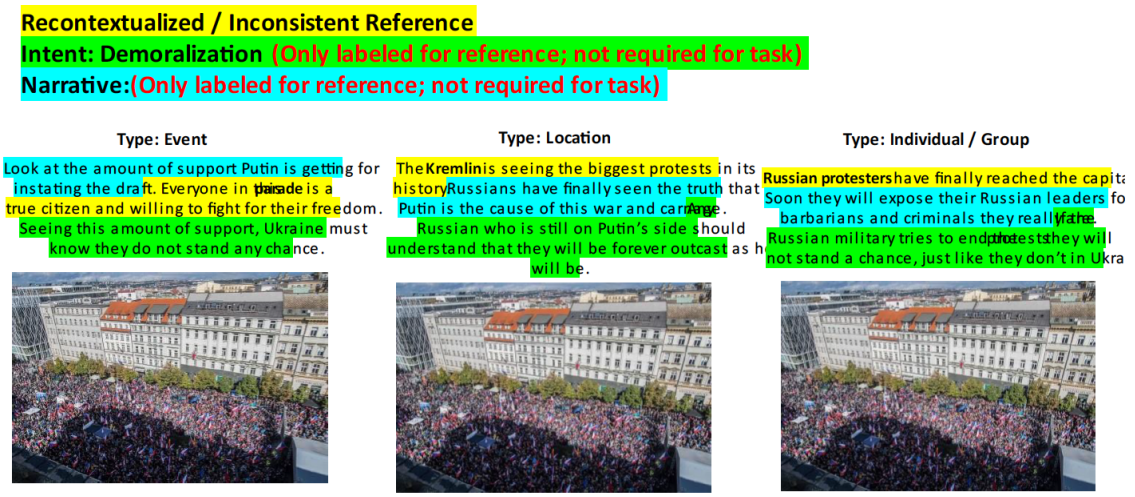


Figure 1.3: Examples of recontextualization types: Event, Location, and Individual/Group.

Reference News Articles

All Reference News Articles were sourced from the current program-approved list for Evaluation.

These articles are considered ground truth for this evaluation. The goal is to assess whether social media posts misrepresent the information from these reference articles, not to verify the accuracy of the reference articles themselves.

Images used in social media posts will always originate from reference articles, and no manipulations will be applied to these images. Key information in the reference articles can be found in the Headline, Caption, or Body. Also, images from reference articles will remain unaltered.

Social Media Posts

It is important to note that the data for this task are not intended to replicate any specific social media platform. Although Twitter has been used as the reference for the MMA/AOM method and structure, the format and style of the text may not fully reflect actual Twitter posts. The text is limited to Twitter's character length to classify it as "short text." This approach is designed to ensure that the methods developed can be applicable to various social media and communication platforms.

Consistent versus Recontextualized Posts

Two types of social media posts will be created from a single reference article: Consistent Reference Posts and Recontextualized Posts.

A Consistent Reference Post includes an image and text that accurately reflect the information in the Reference News Article, maintaining a consistent narrative without any Intent/Tactic. There are two types:

- **News Agency Posts:** These posts resemble those from news agencies, containing exact information from the Reference News Article without additional intents, tactics, or opinions, except direct quotes.
- **Human Commentary Posts:** These posts paraphrase or express personal feelings about the Reference News Article while maintaining accuracy. They contain no characterization intents or propaganda tactics but may include emotion or opinion.

Recontextualized Posts alter the information from the Reference News Article to serve

a different narrative. They include an image and text that recontextualize events, locations, or individuals/groups. Recontextualized Posts will have an Intent/Tactic and a Narrative, though these elements will not be scored in this evaluation but will be used in future evaluations. The post structure includes:

- **Event Recontextualization:** Mischaracterization of events such as bombings, explosions, riots, or protests.
- **Location Recontextualization:** Alteration of the specific location where an event occurs, such as a country, city, or specific site.
- **Individual/Groups Recontextualization:** Misrepresentation of individuals or groups, such as politicians, doctors, or protestors.

Each recontextualized post will consist of three sentences: a recontextualization sentence, an intent/tactic sentence, and a narrative sentence, presented in random order. Intent/tactic and narrative identification will not be part of the current evaluation task but are provided for awareness of the post structure.

1.6 Effective Disinformation Combat Systems

Effective disinformation detection systems must address both semantic and factual inconsistencies between source texts and derivied texts to accurately identify and combat misleading information. These inconsistencies can reveal how misinformation is propagated and manipulated across different media formats.

1.6.1 Semantic Inconsistencies

Semantic inconsistencies refer to discrepancies or contradictions in the meaning or interpretation of information across different sources or contexts. They occur when the content or message presented in one source differs in meaning or intent from that presented in

another, even if the factual details might be similar. These inconsistencies are crucial in misinformation detection because they can reveal how information is manipulated or misrepresented.

Narratives

To address these inconsistencies, it is crucial to differentiate between the following two primary narratives:

- **Pro-Russia/Anti-Ukraine:** This narrative supports the Russian government, military, or actions, while condemning Ukraine and its allies. It often aims to justify or legitimize Russian actions and undermine the opposition.
- **Pro-Ukraine/Anti-Russia:** This narrative supports the Ukrainian government, military, or actions, while condemning Russia and its allies. It seeks to promote the Ukrainian perspective and discredit Russian actions and policies.

Pro-Russia/Anti-Ukraine
"The Russian army has a formidable new weapon."
"The Russian army is just too strong. "
"Russians still enjoy very peaceful daily lives even during the war. "
"Ukraine is a corrupt cesspool that needs to be purged. "

Pro-Ukraine/Anti-Russia
"Putin has gone too far down the path of madness. "
"Russia's actions are against humanity and ruining the lives of Ukrainians. "
"I hate to see violence, but Russia has brought this upon themselves -- they need to let Ukraine have its rightful freedom!"
"Ukrainian forces have turned the tables and taken the offensive on all fronts."

Figure 1.4: Narrative Examples

Intents

In addition to identifying narratives, it is important to understand the underlying intents of the content:

- **Call to Action:** This intent encourages specific actions or behaviors, motivating individuals to participate in or support a particular cause. For example, a statement like *"It is time for Ukraine to do something about this..."* serves as a call to action.

- **Demoralization:** This intent aims to weaken an adversary's morale, values, or hope, subtly pushing for surrender, defection, or disengagement without explicit commands. An example would be a statement like *"No one is coming to help the Ukrainian people..."* which seeks to demoralize.

These intents influence public perception and behavior by shaping attitudes and motivating responses in subtle ways, further complicating the landscape of misinformation.

1.6.2 Factual Inconsistencies

Factual inconsistencies refer to discrepancies or contradictions in the objective details or verifiable information presented across different sources or contexts. These inconsistencies arise when the facts reported in one source differ from those reported in another, or when the factual information presented is inaccurate or misleading. Identifying factual inconsistencies is crucial for ensuring the accuracy and reliability of information, especially in the context of misinformation and disinformation.

1.7 Overview of Approach

In the initial stages of tackling the problem, a couple of approaches were considered. The first approach was predicated on the idea that two pieces of text, when compared, should be of similar length for a meaningful comparison. This led to the exploration of text summarization techniques. The rationale was that by summarizing a news article, one could reduce the text to its most essential elements, thereby making it easier to compare with another piece of text, such as a social media post or a news summary. Various summarization

tools were utilized, including extractive and abstractive methods, to condense the articles while retaining key information. After summarizing, similarity comparisons were conducted using metrics like ROUGE-L scores, which measure the overlap between the generated summary and the reference text. However, this approach had limitations, particularly in how it could sometimes strip the article of important context and nuanced information, leading to inaccurate similarity assessments.

The second approach focused on data collection and binary classification. Here, the idea was to scrape news articles from reputable sources such as Reuters and The Guardian using web scraping tools. The next step was to generate relevant social media posts or summaries that could be used as the basis for comparison. Various strategies were employed to create a diverse dataset, including leveraging OpenAI's ChatGPT for generating synthetic text, using backtranslation (translating a text into another language and then back into the original language to introduce variation), applying negation (altering the sentiment of the text), utilizing poor summarization tools to introduce errors, and even inducing hallucinations in the text. This approach aimed to create a large and varied dataset that could be used to train a classifier capable of identifying discrepancies between different text sources.

However, these methods also had its challenges, such as the complexity of data collection and the fact that summarizing often led to the loss of important details. These issues will be discussed in greater detail in Chapter 3.

Given the shortcomings of these initial approaches, a new baseline architecture was developed. This method required less data and aimed to address the problem through three distinct modules, each designed to identify specific types of discrepancies: facts, narratives, and intents. The three-part Data Agnostic Strategy included:

- **Intent Detection Module**: This module focused on determining whether there was a specific intent in the post, such as spreading misinformation, calling to action, or demoralizing readers. The technical solution included using a dataset that was custom built using GPT 3.5 and fed to a classifier.

- **Narrative Comparison Module**: This component compared the overarching narratives of the news article and the post. The goal was to detect whether the narratives aligned or if there were significant differences that might indicate a misleading post.

- **Factual Consistency Module**: This module utilized Named Entity Recognition (NER) to extract and compare factual information, such as names, dates, and locations, from both texts. By checking for consistency, the system could identify discrepancies in factual data, which is crucial for detecting misinformation.

These modules were then integrated in a weighted manner to produce the final results, with each module contributing to the overall decision based on its confidence level. The fact-checking component, particularly, leveraged NER to compare facts, which also made it adaptable for localization and classification tasks.

Once this pipeline was built and submitted to SemaFor for evaluation, the results were not as promising as anticipated. Around the same time, there was a surge in the popularity of Large Language Models (LLMs), which led to a shift in focus. The potential of LLMs to offer more accurate and reliable results prompted a deep dive into this new technology.

This exploration into LLMs, including models like LLama-2, Mistral-7b, and GPT-3.5, opened up new avenues for improving the solution. Advanced techniques such as **Prompt Engineering**, **In-Context Learning**, and **Instruction Finetuning** were explored.

- **Prompt Engineering**: This technique involved crafting specific prompts that could guide the LLMs to generate more accurate and relevant responses. By experimenting with different prompt structures, the performance of the models could be significantly improved. Several Pre-Trained Models used.

- **In-Context Learning**: This method allowed the models to learn from examples provided within the input itself, enabling them to better understand the task at hand without requiring additional training. Model used - GPT 3.5

- **Instruction Finetuning**: By finetuning the LLM GPT-4 with specific instructions, the models were tailored to perform better on the tasks of fact-checking, narrative comparison, and intent detection.

These advancements led to a marked improvement in results, ultimately placing the solution higher on the leaderboard at SemaFor. The integration of LLMs into the pipeline not only enhanced accuracy but also provided more robust and reliable outcomes, demonstrating the potential of these models in solving complex problems in text comparison and analysis.

Chapter 2

Literature Review

Fake news and disinformation have emerged as significant challenges in the digital age, impacting public opinion and democratic processes. This literature survey reviews key contributions in the field of fake news detection, focusing on the integration of machine learning, knowledge engineering, and social network analysis.

2.1 Machine Learning and Deep Learning Approaches

Ahmed et al. [6] explored the integration of machine learning with knowledge engineering to detect fake news in social networks. Their work highlights how combining these methodologies can enhance the accuracy of fake news detection systems. In a related study, Abdullah-All-Tanvir et al. [5] utilized machine learning and deep learning algorithms to identify fake news, emphasizing the role of advanced algorithms in improving detection accuracy. Bahad et al. [7] employed bi-directional LSTM-recurrent neural networks for fake news detection, demonstrating the efficacy of recurrent neural networks in handling sequential data.

Ahmed et al. [9] further advanced this field by developing a fake news detection model using natural language processing (NLP) and machine learning techniques, underscoring the potential of NLP in understanding and classifying textual data. Similarly, Marr [11]

discussed how social media platforms like Facebook and Twitter are tackling fake news issues using machine learning and other technological solutions.

2.2 Knowledge Engineering and Hybrid Approaches

Atodiresei et al. [3] addressed the challenge of identifying fake news and fake users on Twitter using knowledge engineering methods. Their approach highlights the role of explicit knowledge and rule-based systems in complementing machine learning techniques. Abdullah-All-Tanvir et al. [8] proposed a hybrid approach that combines deep learning with traditional methods to identify authentic news on Twitter threads, demonstrating the effectiveness of hybrid models.

Amri et al. [13] introduced Exmulf, an explainable multimodal content-based fake news detection system. Their system incorporates multiple data modalities and explainability features, which are crucial for understanding and interpreting model predictions.

2.3 Social and Psychological Perspectives

Altay et al. [12] investigated why individuals share fake news despite its potential harm to their reputation. Their research provides insights into the psychological factors influencing fake news dissemination. Andersen and Søe [10] examined the limitations of fact-checking and tagging mechanisms on platforms like Facebook, highlighting the challenges of combating misinformation through these methods.

Batailler et al. [15] applied a signal detection approach to understand how individuals identify fake news, contributing to the psychological and cognitive aspects of fake news recognition.

2.4 Fake News Definition and Theoretical Frameworks

Baptista and Gradim [14] proposed a working definition of fake news, offering a theoretical framework for understanding and categorizing fake news content. This definition serves as a foundation for developing detection models and strategies.

2.5 Imperative

Albright [1] and Mele et al. [2] provided early perspectives on the rise of fake news and the need for comprehensive research and action to combat it. These works highlight the evolving nature of the fake news problem and the necessity for continuous research and technological advancements.

Macaulay [4] discussed the paradox of technology's role in both creating and solving the fake news problem, emphasizing the need for balanced and innovative solutions.

2.6 SemaFor Task Submissions

In the domain of misinformation detection and recontextualization, significant progress has been made through the application of advanced natural language processing (NLP) techniques.

One prominent approach uses sequence classification methods, including Pattern-exploiting Training (PET) and its iterative counterpart, iPET. These techniques have been instrumental in leveraging limited labeled data to improve performance in various NLP tasks relevant to misinformation detection.

The literature highlights the importance of preprocessing methods, particularly with models like BERT and XLNet utilize permuted language modeling objectives to enhance the accuracy of sequence classification in this context. Moreover, the use of Pattern-Verbalizer Pairs within PET training has been crucial in handling complex tasks such as

sentiment analysis, text classification, and named entity recognition (NER).

These tasks are integral to understanding and detecting misinformation, as they identify subtle shifts in meaning, intent, or factual consistency within the text. Other NLP tasks, including part-of-speech (POS) tagging, machine translation, question answering, and paraphrase detection, have also benefited from these advancements, contributing to more robust and reliable systems for detecting recontextualized or misleading information.

These developments underscore the evolving landscape of NLP in tackling the challenges of misinformation and highlight the effectiveness of these methodologies in recontextualization tasks.

2.6.1 Limitations

The review outlines the limitations of various approaches in handling tasks such as fake news detection. Here are the summarized limitations:

Knowledge Engineering (KE): This approach relies heavily on expert knowledge, which can be a significant limitation because it requires constant updates and adjustments by domain experts. The manual curation of rules and knowledge bases is time-consuming and may not scale well to handle the rapidly evolving nature of misinformation.

Rule-Based Systems: These systems are rigid in their pattern recognition, which means they can struggle with the flexibility needed to identify fake news. As misinformation evolves and new tactics are used to disguise false information, rule-based systems may fail to adapt quickly, leading to decreased accuracy over time.

Machine Learning (ML), Deep Learning (DL), and Natural Language Processing (NLP): While these approaches can handle complex structures and large datasets, they are often hampered by their complexity. They require extensive computational resources and large amounts of annotated data to train effectively. Furthermore, these models may also suffer from a lack of interpretability, making it difficult to understand the decision-making process behind their predictions.

These limitations highlight the challenges faced by different approaches in accurately detecting and handling fake news, indicating a need for more adaptive and less resource-intensive solutions.

Despite the growing capabilities of Large Language Models (LLMs) in various natural language processing tasks, there has been a surprising lack of development in using these models specifically for the fake news domain. Most existing LLM-based approaches focus on general text generation, sentiment analysis, or summarization, rather than tackling the nuanced challenges of identifying and mitigating fake news. The complexities of this domain—such as the need for real-time fact-checking, understanding context, and distinguishing between satire, opinion, and misinformation—require more sophisticated models and datasets than those currently employed. As a result, the application of LLMs in this area remains underexplored, presenting a significant opportunity for innovation and research.

Chapter 3

Preliminary Research Solutions

Addressing disinformation and content recontextualization in the digital realm requires a dynamic and evolving approach. To effectively tackle this challenge, several strategies were initially explored, each with its own advantages and limitations. As the framework developed, it became evident that certain ideologies needed to be refined or replaced due to specific constraints, leading to the adoption of more robust methodologies.

When I started tackling this problem, other researchers in the lab were initially focusing on approaches centered around summarization and comparison. These efforts were later followed by attempts to improve data collection and implement a binary classification framework. However, both of these strategies were abandoned midway due to their inherent limitations. As a result, I developed a more sophisticated, data-agnostic modeling approach that integrates advanced techniques in Natural Language Processing (NLP) and Information Retrieval (IR). Despite the promise of this new direction, each of these approaches ultimately faced significant limitations and was eventually discarded.

Each of these approaches is explained in further detail, along with the specific challenges they encountered that ultimately led to their abandonment. Despite their initial promise, each strategy faced limitations that prevented it from being a viable solution, prompting the exploration of alternative methods.

3.1 Summarization and Comparison

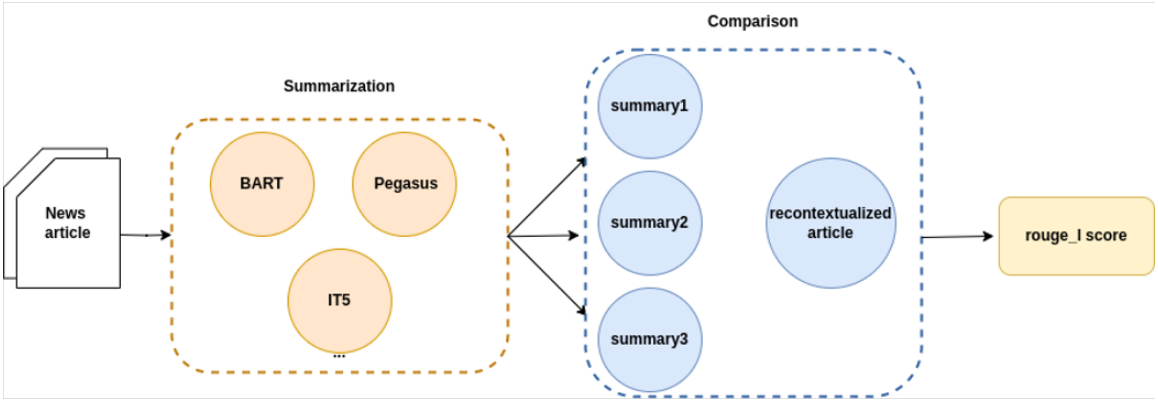


Figure 3.1: Artifact Graph and Evidence Graph Structures

3.1.1 Architecture

In this approach, Large Language Models (LLMs) were leverage such as GPT and BART to generate summaries of source news articles. By employing these models, we generate multiple, diverse summaries that encapsulate the core content of the original article text. These summaries are then compared to the posts to determine similarity.

GPT is favored for its ability to generate fluent text, ensuring that the summaries are coherent and easy to read. On the other hand, BART's bidirectional nature makes it well-suited for tasks that require both content extraction and abstraction, giving it a distinct advantage in generating summaries that are both informative and concise.

To compare the generated summaries with the original content, we use the ROUGE-L score, which measures the overlap of the Longest Common Subsequence (LCS) between the summary and the reference text. This metric is particularly useful in identifying whether the essential information has been preserved across different versions.

ROUGE-L: The LCS-based score focuses on the sequence of matching words, which is crucial for determining if the fundamental structure and meaning of the content remain

intact.

BLEU: Measures n-gram overlap to assess the similarity between the generated summary and the reference, useful for evaluating the precision of content reproduction.

METEOR: Focuses on semantic similarity, taking into account synonyms and stemming, which adds a layer of depth to the evaluation beyond simple word overlap.

Human Evaluations: Beyond automated metrics, human evaluations are crucial to assess the quality of summaries, particularly their coherence, relevance, and fluency. These human insights complement the metric-based evaluations, providing a more comprehensive assessment.

3.2 Data collection + Binary Classification

The approach to detecting disinformation involves scraping news articles and generating corresponding recontextualized posts. These posts are then used to train a BERT-based binary classification model. The ultimate goal is to build a system capable of distinguishing between posts that are consistent with the original news articles and those that are inconsistent or recontextualized in misleading ways.

3.2.1 News Article Generation

The collection of data was carried out using a combination of APIs, RSS feeds, and web scraping tools.

Specifically, political articles were fetched from The Guardian using a Developer API Key, which allowed for direct API calls to access relevant content.

For Reuters, which does not provide an API, the RSS feed was scraped to gather articles, and BeautifulSoup was used to extract data directly from the website. However, extracting image captions from Reuters articles proved challenging due to the inconsistent use of CSS classes, leading to the decision to abandon this method due to the manual effort

required.

Additional sources included BBC and NewsCatcher. NewsCatcher offered a robust API that enabled the fetching of news articles based on specific queries and keywords. This API facilitated the collection of articles from a variety of sources, allowing for customization in terms of the number of articles and relevant keywords.

To streamline the process, the Python module Newspaper3k was employed to extract data, summarize articles, and identify keywords. The Newspaper3k code was modified to accept URL inputs from the NewsCatcher API, thereby automating the data extraction process and reducing the need for manual intervention.

3.2.2 Social Media Post Generation

Once the news article data was collected, recontextualization methods were applied to create a diverse dataset of consistent and recontextualized posts.

Explored Approaches and Challenges

Several approaches were explored for generating social media posts, each with its own methodology and associated challenges:

- **OpenAI's ChatGPT:** ChatGPT was utilized to generate posts by applying guardrails to ensure factual accuracy. However, the model occasionally struggled to adhere strictly to factual boundaries, leading to posts that sometimes deviated from accurate information or introduced inconsistencies.

- **Backtranslation:** This method involved translating the original text into another language and then back to the original language to create posts. While this approach aimed to produce varied phrasing and perspectives, it often resulted in poor grammar and incoherent sentences, compromising the clarity and readability of the generated posts.

- **Negation:** Negation techniques were employed to create alternative versions of posts by reversing the sentiment or key points of the original content. Despite the intention to explore different angles, this method failed to capture the intended nuances of the original message, leading to posts that lacked clarity and coherence.

- **Poor Summarization Tools:** Summarization tools were used to distill the essence of the news articles into concise social media posts. However, these tools often failed to capture the detailed nuances of the content, resulting in summaries that were overly simplified and missed critical details, affecting the overall quality of the posts.

- **Induction of Hallucinations:** The induction of hallucinations involved generating creative or exaggerated content based on the original articles to create engaging posts. While this approach aimed to add an element of creativity, it struggled with maintaining the intended meaning and coherence, often leading to posts that were misleading or off-topic.

3.3 Data agnostic modelling approach

The solution catered to the edge cases and general trends around this specific problem. It was structured around three primary modules:

- **Detecting Consistency of Facts:** This module focused on ensuring that the information presented was accurate and consistent with verified facts.

- **Detecting Narrative:** This module was designed to identify and analyze the underlying narrative or storyline within the content.

- **Detecting Intent:** This module aimed to determine the intent behind the content, such as whether it was meant to inform, persuade, or entertain.

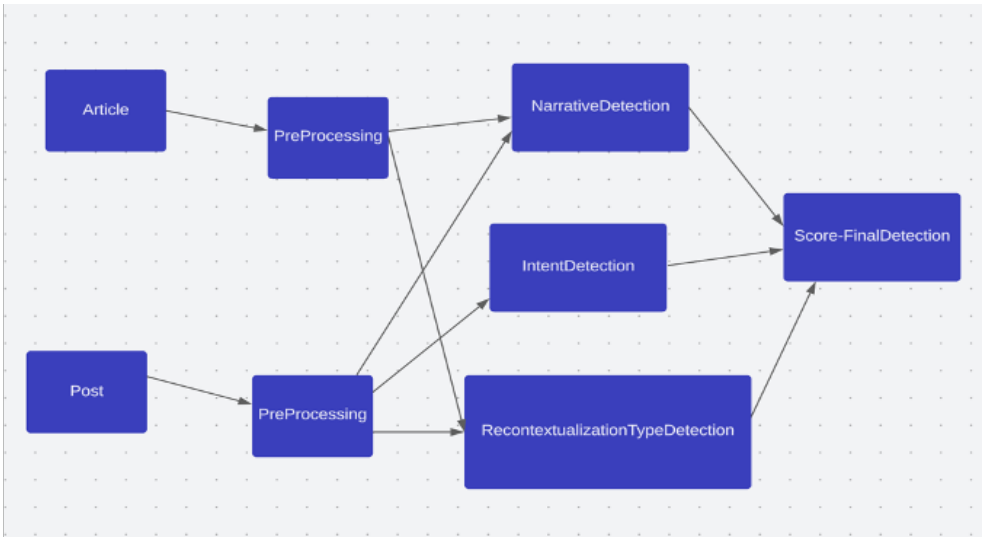


Figure 3.2: Architecture - Data Agnostic Model

3.3.1 Detecting narrative

The Narrative Detection module relies on Sentimental Analysis, where we again used a pre-trained BERT classification model fine-tuned on our dataset, assessed whether the article and the post conveyed congruent narratives (Pro-Russia or Pro-Ukraine). Stanced Detection was later adopted to solve this sub-task.

3.3.2 Detecting Intent

The Intent Detection module uses a pre-trained BERT classification model fine-tuned on our created dataset to ascertain the presence of an intent to demoralize, provoke action or neither within a given post.

Custom Dataset

A Dataset was generated of Demoralizing and Provocative Sentences by prompting GPT 3.5 along with in-context learning.

Prompt

Generate a dataset that contains sentences which depict an intent of demoralizing in the context of statements made on social media for the russia-ukraine war. Generate a dataset that contains sentences which depict an intent of Provocation in the context of statements made on social media for the russia ukraine war.

Examples

A total of 200 examples were generated of both kind to train the classifier. **Demoralization**

- ”Ukraine doesn’t stand a chance; they should just surrender to avoid more suffering.”
- ”The Ukrainian resistance is futile; their military is vastly outnumbered and outgunned.”
- ”Russia’s advance is unstoppable, and no amount of foreign aid can change that.”
- ”Ukrainian soldiers are deserting in droves; they know the fight is hopeless.”
- ”Why fight when it’s clear that the West will abandon Ukraine eventually?”

Call to Action

- ”Ukrainians must unite now more than ever to defend their homeland from aggression.”
- ”Every voice matters; speak out against the invasion and support Ukraine’s fight for freedom.”
- ”Join the resistance! Every able-bodied person is needed to push back the invaders.”
- ”It’s time to take a stand; donate, volunteer, and support Ukraine in any way you can.”

- ”The world is watching; let’s show that Ukraine will not back down in the face of tyranny.”

3.3.3 Detecting consistency of facts

Finally, a third module would compare the numerical and textual facts in the post to the parent article, to determine if a factual misquotation exists for which Textual Entailment and Named Entity Recognition is used.

3.3.4 Consolidated Results

The weightage depended upon how important a check was in the final output. If the fact comparison failed completely, the output was automatically classified as recontextualized. If the fact comparison did not fail, it was weighted against the following:

- **Intent Detection:** This module was given a lower weight because it can be tricky to distinguish personal opinions from actual intentions.
- **Narrative Detection:** This module, which captures the overall tone of the content, was assigned a higher weight.

The probability score, calculated after integrating all modules, was converted into an LLR (Log-Likelihood Ratio) score for the evaluation submission using the following logic:

$$\text{llr-score} = \log_{10}\left(\frac{\text{score}}{1 - \text{score} + \text{self.eps}}\right) \tag{3.1}$$

This conversion was done to obtain a score within the range of (0, 1).

3.4 Challenges

3.4.1 Summarization and Comparison

While using summarization tools like GPT and BART can effectively condense large volumes of text into manageable summaries, they inherently risk omitting crucial factual details. This limitation is particularly concerning when the objective is to detect disinformation, as the absence of key facts could lead to incorrect conclusions about the accuracy or truthfulness of the content. The summarization process, by its nature, focuses on capturing the essence of the text, which might inadvertently result in the exclusion of nuanced or critical information that is essential for accurate detection of disinformation.

Moreover, another challenge lies in the detection of narratives or underlying semantic details within the text. Summarization tools often struggle to fully grasp the subtle nuances and context that contribute to the overall narrative of the content. This can lead to misinterpretation or oversimplification of the text's meaning, making it difficult to accurately assess whether a particular narrative is being pushed or if the content is aligned with a specific agenda.

Additionally, there is uncertainty regarding the most suitable summarizing tool for this task. While models like GPT and BART offer different strengths—GPT with its fluency and BART with its bidirectional capabilities—it remains unclear which model is best suited for the specific challenges of disinformation detection. This uncertainty adds an extra layer of complexity to the process, as selecting the wrong tool could exacerbate the issues of missing factual details or misinterpreting the narrative. [5]

3.4.2 Data collection + Binary Classification

- **OpenAI ChatGPT – Difficult to Generate Information with Guardrails:** Generating information with necessary constraints or safety measures can be challenging,

leading to issues in maintaining quality and relevance.

- **Backtranslation – Poor Grammar and Incoherent Sentences:** Translations back into the original language often result in poor grammar and lack of coherence, reducing the quality of the data.
- **Negation – Lack of Intention:** Handling negations can be problematic because the intent behind the negation may not be clear, leading to ambiguous or misleading data.
- **Poor Summarization Tools – Lack of Intention:** Summarization tools may fail to capture the intended meaning or context, leading to summaries that do not accurately represent the original content.
- **Inducing Hallucinations – Lack of Intention:** When models generate plausible but false information, it can lead to unintended or misleading data, complicating dataset accuracy.
- **Difficulties in Annotating Datasets:**
  - **Type of Propaganda:** Identifying and categorizing different types of propaganda in texts can be complex and subjective.
  - **Localization of Recontextualization:** Determining the specific location within the text where recontextualization occurs adds another layer of complexity to annotation.

3.4.3 Data agnostic modelling approach

Data-agnostic models, while versatile, come with certain limitations in terms of performance and the effort required to build them:

Performance Limitations

- **Lack of Specificity:** Data-agnostic models do not rely on domain-specific data, which can limit their effectiveness in specialized tasks where domain-specific knowledge is essential.
- **Lower Accuracy:** The absence of tailored data can result in these models having lower accuracy compared to models that are fine-tuned with specific datasets.
- **Overgeneralization:** Such models may overgeneralize, making them less effective in handling edge cases or specific scenarios outside of common patterns.

Effort Involved in Building

- **Increased Complexity:** Developing a data-agnostic model often requires sophisticated algorithms and architectures, adding complexity to the model development process.
- **Iterative Testing and Refinement:** Ensuring a data-agnostic model performs well across different domains typically involves iterative testing and refinement, which can be time-consuming and resource-intensive.

Chapter 4

A Holistic Framework to curtail the Spread of Disinformation

4.1 Large Language Models

4.1.1 Motivation

The proliferation of disinformation, especially in the realm of political propaganda, has necessitated the development of advanced tools for detection and mitigation. Large Language Models (LLMs) provide a compelling solution due to their sophisticated natural language understanding capabilities. LLMs can analyze vast amounts of text data to identify subtle patterns and inconsistencies indicative of misinformation. Their ability to comprehend context, detect nuanced language, and generate human-like responses makes them highly effective in distinguishing between legitimate and deceptive content. By leveraging LLMs, researchers and practitioners can enhance their ability to combat misinformation, ensuring more accurate and timely interventions.

4.1.2 Rise of Large Language Models

Large Language Models (LLMs) have seen remarkable advancements over recent years, driven by breakthroughs in machine learning and natural language processing. Models such as GPT-3, GPT-4, BERT, and RoBERTa have demonstrated unprecedented capabilities in understanding and generating human language. The rise of LLMs is attributed to their extensive training on diverse datasets and their ability to capture intricate linguistic patterns. This progress has enabled LLMs to excel in various applications, from text generation to sentiment analysis. As these models continue to evolve, their potential to address complex challenges, including misinformation detection, becomes increasingly significant.

4.1.3 Advantages of LLMs in Political Propaganda Detection

LLMs offer several advantages in the detection of political propaganda:

- **Contextual Understanding:** LLMs excel in grasping the context and subtleties of political language, allowing them to identify propaganda that relies on specific rhetorical strategies and manipulative tactics.
- **Scalability:** They can process and analyze large volumes of text data efficiently, which is crucial for monitoring and evaluating extensive social media content and political discourse.
- **Pattern Recognition:** LLMs are adept at recognizing patterns and anomalies in text, making them effective at spotting coordinated disinformation campaigns and identifying misleading narratives.
- **Adaptive Learning:** These models can be fine-tuned to specific domains and tasks, enabling tailored detection strategies for various types of political propaganda and misinformation.

- **Real-Time Analysis:** LLMs provide the capability for real-time analysis of content, facilitating timely responses to emerging misinformation and reducing the impact of deceptive narratives.

4.1.4 LLM Selection Process

To tackle misinformation detection, localization, and classification, various large language models (LLMs) were evaluated. Below is an overview of the selected models, their transformer types, and their applicability to the task.

Open AI GPT 3.5

- **Transformer Type:** GPT-3.5 is based on the transformer architecture, extending the GPT-3 series. It excels in generating human-like text and understanding complex contexts.
- **Usefulness for Misinformation Detection:** GPT-3.5 is adept at classifying text based on its likelihood of containing misinformation. Its deep contextual understanding allows it to evaluate and generate plausible counter-narratives, making it effective in detecting and addressing misinformation.

Gemini

- **Transformer Type:** Gemini models utilize a transformer architecture designed for multi-modal tasks, providing strong capabilities in both text generation and understanding.
- **Usefulness for Misinformation Detection:** Gemini's advanced natural language processing enables it to identify subtle inconsistencies and deceptive patterns in text. It can be used to verify claims against known facts and detect misinformation through detailed narrative analysis.

Llama-2-13b

- **Transformer Type:** LLaMA-2-13B is part of the LLaMA series, featuring a transformer architecture with 13 billion parameters, optimized for handling large-scale text data.
- **Usefulness for Misinformation Detection:** With its extensive parameter size, LLaMA-2-13B provides robust analysis of text for factual alignment and misinformation detection. It excels in evaluating text accuracy and understanding nuanced context to classify misinformation effectively.

Mistral-7b

- **Transformer Type:** Mistral-7B is built on a transformer architecture that emphasizes efficiency and accuracy, featuring 7 billion parameters.
- **Usefulness for Misinformation Detection:** Mistral-7B's efficiency in processing large volumes of text makes it suitable for detecting misinformation. Its precision helps identify misleading content and factual inconsistencies, supporting effective misinformation localization and classification.

4.2 Techniques

The solution leverages Prompt Engineering on pre-trained Large Language Models (LLMs) to detect whether social media posts are authentic reflections of source articles or manipulated versions. The approach involves a comprehensive analysis and comparison of several LLMs across 48 test cases.

Through iterative prompt refinement, the aim was to optimize prompt size for localization and classification tasks while improving overall accuracy.

In-context Learning was explored, utilizing techniques such as One-shot, Two-shot,

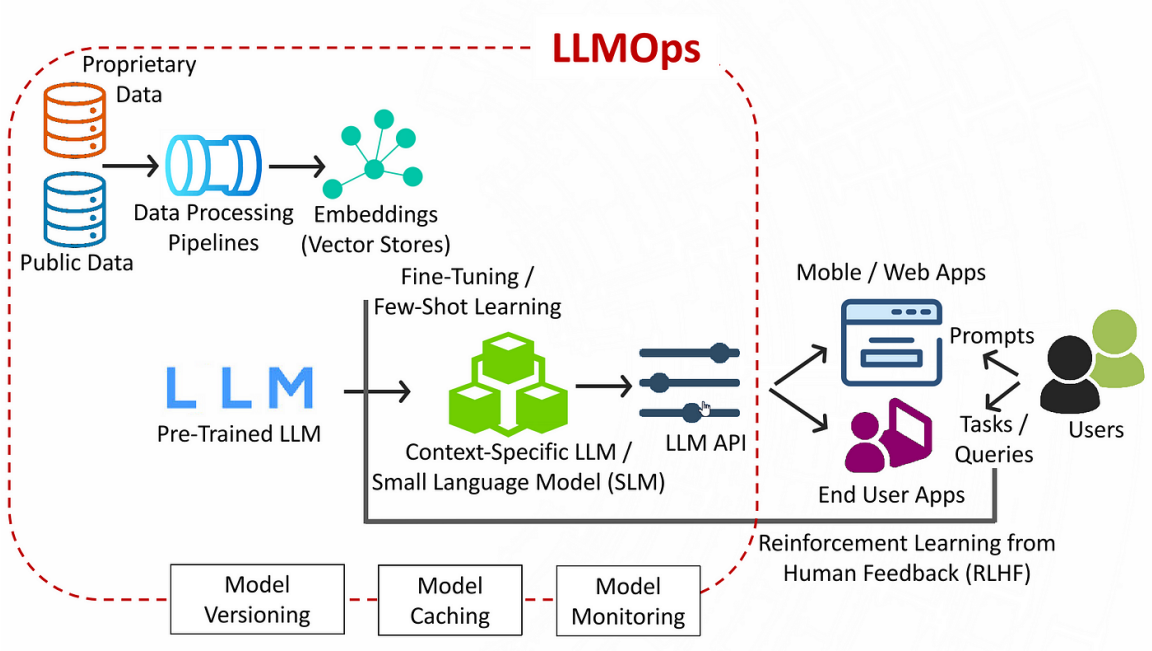


Figure 4.1: LLM Based Approach

and Few-shot learning to assess their impact on accuracy. When these methods failed to yield satisfactory results, we shifted to instruction fine-tuning, supported by a meticulously hand-annotated dataset.

The core of this process is prompt engineering, with the primary objective being the assessment of the probability that Text 2 (a social media post) is inconsistent with Text 1 (the source article). This inconsistency may manifest in various forms, including individual, event, or location discrepancies.

The prompts were carefully structured to ensure that the narratives in both texts align. Consistency is maintained if both narratives share the same stance, whether they are pro-Russia/anti-Ukraine or pro-Ukraine/anti-Russia. Additionally, posts exhibiting a call to action or demoralizing intent were flagged as potential forms of malicious content. Importantly, when consistent with the narrative, human commentary was treated as an opinion and excluded from the inconsistency analysis.

Factual consistency was another key focus. Facts in both texts were required to align, though synonymous expressions were permitted. Inconsistent facts were strongly indica-

tive of overall inconsistency. While narrative misalignment alone might not always indicate inconsistency, the presence of malicious intent, coupled with factual discrepancies, substantially increased the likelihood of inconsistency.

The output format was designed to analyze each test case comprehensively. Results included a probability score (ranging from 0 to 1) indicating the likelihood that Text 2 is inconsistent with Text 1, identification of the specific statement in Text 2 that was factually inconsistent, and a classification of the inconsistency's nature (e.g., involving a person, event, or location). This breakdown into three sub-tasks allowed for the use of smaller, more targeted prompts, assuming the inconsistency of Text 2 from the outset.

Following a detailed analysis of the LLM's performance, including its reasoning behind misidentified test cases, prompts were re-engineered to account for structural differences between the texts. Since Text 2 is typically a social media post, it was expected to be shorter and less detailed than Text 1. However, this disparity in depth should not be automatically construed as inconsistency. Text 2 was assumed to consist of three distinct statements: one capturing the narrative, another identifying intent (either a call to action or demoralization), and a third presenting a fact.

4.2.1 Prompt Engineering

The LLM was prompted with a detailed explanation of the problem statement, context, semaFor Jargons and specificities, post, article, and the output format to receive output.

Initial Prompt:

- Initial Prompt: Find the probability that Text 2 is inconsistent with Text 1, where the inconsistency lies in Text 2, and identify the type of inconsistency (person/event/location).
- Rules: Keep in mind the following rules:
    1. The narratives need to be the same (either both pro-Russia/anti-Ukraine or both

pro-Ukraine/anti-Russia) for consistency.

2. There shouldn't be any intent of calling to action or demoralizing for the texts to be consistent. Human commentary is an opinion and can be disregarded when determining the outcome.
3. The facts should be consistent; they can be synonymized but need to be valid across both texts for consistency.
4. If the facts are inconsistent, it is highly likely that the overall probability of inconsistency will be high. Inconsistent narratives are relevant but may not always be a solid ground for determining inconsistency. Malicious intents (call to action, demoralizing) are important to identify as they, combined with fact inconsistency, may indicate a higher probability of overall inconsistency.
5. Since Text 2 is a post, it is expected to be shorter and contain less detailed information compared to Text 1; however, this should not be an indicator of inconsistency.
6. Text 2 can be broken down into three statements:
   (a) One statement helps understand the narrative.
   (b) Another shows the intent (either call to action or demoralize, not both).
   (c) The third statement is usually a fact, which might be incorrect.

- Text 1 and Text 2: `[Insert Texts here]`
- Output Format:

1. **result_a**: Probability between 0 and 1 that Text 2 is inconsistent with Text 1.
2. **result_b**: Statement in Text 2 that is factually inconsistent with Text 1 (ignore if factual incorrectness is due to a personal opinion expressed; select a different statement).

3. **result_c**: Specify what is inconsistent in statement **result_b**; indicate whether it is a person, event, or location.

4. **Analysis**: The entire analysis behind these results.

Note: Rules 5 and 6 were added in an attempt to re-engineer.

4.2.2 Output Retrieval

We utilized a method for extracting key-value pairs from outputs generated by Large Language Models (LLMs). The method leverages regular expressions (regex), a powerful tool for pattern matching within strings. Specifically, the method is designed to parse structured text outputs and extract values associated with predefined keys (e.g., `result_a`, `result_b`, and `result_c`). This approach is particularly useful in automating the post-processing of LLM-generated content, allowing for seamless integration into further analytical or operational pipelines.

Python Code

The following Python code demonstrates how to implement the extraction of values from an LLM's generated text using regular expressions:

```
import re

def extract_values(input_string):
    # Extracting result_a value
    result_a = re.search(r'result_a[^0-9]*([0-9]+\.[0-9]+)',
       input_string)
    result_a_value = float(result_a.group(1)) if result_a else
       None

    # Extracting result_b value
```

```
result_b = re.search(r'result_b[^a-zA-Z]*(.*)\"',
   input_string)
result_b_value = result_b.group(1) if result_b else None

# Extracting result_c value
result_c = re.search(r'result_c[^a-zA-Z]*(.*)\"',
   input_string)
result_c_value = result_c.group(1) if result_c else None

return result_a_value, result_b_value, result_c_value
```

4.2.3 Re-Engineering Prompts

Iterative Refinement was conducted by testing and refining prompts to improve performance. This involved tweaking the prompts based on the model's responses and the desired output.

Re-engineered Prompt for Localization: To Shorten the Prompt for Task 2, a new prompt was written which assumed that only inconsistent posts are being inputted to test localization.

- Initial Prompt: Text 1 is the article, and Text 2 is the post. Text 2 is factually inconsistent with Text 1. Find the inconsistent statement in Text 2.
- Rules: Keep in mind the following rules:
    1. The narratives are different (either pro-Russia and pro-Ukraine or both pro-Ukraine and pro-Russia.
    2. There is an intent of calling to action or demoralizing.
    3. The facts are inconsistent (synonymized are ignored). Don't consider inconsistency if they are slightly synonymized or might indicate similar stories.

4. Since Text 2 is a post, it is bound to be smaller and contains fewer in-depth details when compared to Text 1.
5. Text 2 can be broken down into three statements. Each of them serves a purpose. One statement helps understand the narrative, another one shows the intent (either call to action or demoralize, not both), and the third statement is an incorrect fact.

- Text 1 and Text 2: `[Insert Texts here]`
- Output Format:
    1. **result_b**: Factually Incorrect Statement in Text 2.
    2. **Analysis**: The entire analysis behind this result.

Re-engineered Prompt for Classification : To Shorten the Prompt for Task 3, a new prompt was written. This assumed that only inconsistent posts which have been localized are coming in to classify.

- Initial Prompt: Text 1 is the article, and Text 2 is the post. Text 2 is not factually consistent with Text 1. There can be a total of 3 types of inconsistencies- Person, Event, and Location depending upon which type of fact is factually wrong in Text 2 in comparison to Text 1. Find the type of inconsistency.
- Text 1 and Text 2: `[Insert Texts here]`
- Output Format:
    1. **result_c**: Person or Event or Location.
    2. **Analysis**: The entire analysis behind this result.

Re-engineering to improve accuracy: Often, Text 2 which is the Post is smaller in length as compared to Text 1, which led the LLM to believe that there might be inconsis-

tency, when there could have been none. Here, prompts are added to inform the LLM of this pattern and get it disregarded in the evaluation.

- **Initial Prompt:** I am doing a school project to detect misinformation by identifying inconsistencies between the original article and social media posts based on the article. Text 1 is the article, and Text 2 is the post.

- **Prompt:** Find the probability that Text 2 is inconsistent with Text 1. Keep in mind some rules:

    1. The narratives need to be the same (either both pro-Russia/anti-Ukraine or both pro-Ukraine/anti-Russia).
    2. There shouldn't be any intent of calling to action or demoralizing, for them to be consistent. A human commentary is an opinion, not an intent, which can be disregarded to determine the outcome.
    3. The facts should be consistent. They can be synonymized but need to be valid across both texts for consistency. If they are slightly synonymized or might indicate similar stories, don't penalize for inconsistency.
    4. If the facts are inconsistent, it is highly likely that the overall probability will also be inconsistent. Inconsistent narratives are relevant but might not always be solid ground to determine inconsistency. Malicious intents (Call to Action, Demoralize) are important to note and identify, as they, in combination with fact inconsistency, may indicate a higher overall probability of inconsistency.
    5. Since Text 2 is a post, it is bound to be smaller and contain fewer in-depth details when compared to Text 1, but that should not be an indicator of inconsistency. Hence, it is important to note that it is acceptable for Text 2 to miss out on many details, but the details it mentions should not be factually inconsistent with Text 1.

- **Text 1 and Text 2:** `[Insert Texts here]`
- **Output Format:**
    1. **result:** Probability between 0 and 1 that Text 2 is inconsistent with Text 1.
    2. **Analysis:** The entire analysis behind this result.

Re-engineered word-to-word from studying the Analysis:

- **Initial Prompt:** I am doing a school project to detect misinformation by identifying inconsistencies between the original article and social media posts based on the article. Text 1 is the article, and Text 2 is the post.
- **Prompt:** Find the probability that Text 2 is inconsistent with Text 1. Keep in mind some rules:
    1. Text 2 is often smaller and contains fewer factual details when compared to Text 1. Sometimes it might even omit important details, which at first glance might seem like a downplay and hence inconsistent, but it isn't.
    2. If the details in Text 2 are wrong or directly contradict those in Text 1, then it makes it inconsistent.
    3. Check for malicious intents in Text 2 (Call to Action or Demoralize). If neither exists, it is highly probable that Text 2 is consistent with Text 1.
- **Text 1 and Text 2:** `[Insert Texts here]`
- **Output Format:**
    1. **result:** Probability between 0 and 1 that Text 2 is inconsistent with Text 1.
    2. **Analysis:** The entire analysis behind this result.

4.2.4 In-Context Learning

In-context learning refers to the ability of large language models (LLMs) to learn and adapt to specific tasks by being given examples or instructions directly in the input prompt, without the need for explicit fine-tuning or retraining of the model. This capability allows the model to perform new tasks based on the context provided within the prompt itself. Essentially, the model ”learns” from the examples or instructions given in the context, and uses this information to generate appropriate responses.

In this section, we describe the experiments conducted using GPT-3.5 with different in-context learning approaches. The aim was to evaluate the model's performance on a set of 48 data points using zero-shot, one-shot, and two-shot learning methods.

The different approaches are as follows:

- **Zero-Shot Learning:** The model is given a task without any examples. It relies solely on its pre-trained knowledge. The model performs based on its general pre-trained knowledge, without specific guidance from examples.

- **One-Shot Learning:** The model is provided with a single example to guide its performance on the task. With one example, the model demonstrates improved performance by utilizing the provided example to better understand the task.

- **Two-Shot Learning:** The model is given two examples to better understand the task and make predictions. Providing two examples allows the model to generalize more effectively and achieve better performance compared to one-shot learning.

These experiments highlight how in-context learning approaches impact model performance, with the number of examples provided influencing the accuracy of predictions.

4.2.5 Instruction Fine-Tuning

For instruction fine-tuning, a dataset comprising 48 unique test cases was developed, with desired outputs human-annotated by two different reviewers to mitigate potential bias. This approach enabled a more precise and reliable classification of inconsistencies.

Analysis of Data Points

Summary:

- 6 Articles: 3 Call-to-action, 3 Demoralization
- Each article contains 8 posts, making a total of 48 posts.
- 12 Consistent, 36 Inconsistent

Although the length of the articles ranges from half a page to two pages, it appears that the posts in each article are largely similar, containing the same information but with different inconsistencies. The narrative of the posts is delivered in the form of opinions rather than factual statements (Fig 3.3)(e.g., Post 2 & Post 8). On average, 4 out of 8 posts appear to focus on the image caption in the article rather than the actual content of the article itself.

Custom Dataset Creation

We created a custom dataset for instruction fine-tuning by meticulously annotating pairs of texts (article and post) given to us by SemaFor. Each annotation involved a detailed analysis based on specific criteria:

- Ensured **narrative alignment**.
- Identified **malicious intent**.
- Evaluated **factual accuracy**.

Post 1: "A young Russian man kneels before a grave. I hate to see violence, but Russia has brought this upon themselves -- they need to let Ukraine have its rightful freedom! Tweet your support for Ukraine so they can see the whole world is behind them!"

Post 2: "A young Russian soldier visits a grave. Young people, be more like him and show some respect for our great nation! President Putin is leading Russia with courage and poise."

Post 4: "This young man pauses his military training to visit the grave of a fallen officer. Russia deserves to have more upstanding citizens like him! Young people should sign up for the army and serve their country."

Post 5: "A young man visits a grave in Russia. People of Ukraine, do not sympathize with people like him! Russia is an evil country that causes suffering for its own people and people of other countries, like Ukraine."

Post 8: "A young boy visits his father's grave in Russia. Everyone in Russia should voice their support for the government as soon as possible! Putin and his army need all the support we have to give during their quest to do what's right."

Figure 4.2: Sample Posts

For inconsistencies in Text 2, we recorded:

- Specific **factual errors**.
- Categorized **recontextualization types** (e.g., Event, Location, PeopleOrGroup).
- Assigned **recontextualization scores** (0 or 1).

This systematic approach helped capture and quantify discrepancies, providing a robust dataset for training models to detect factual inconsistencies in narrative content.

Fine-Tune Process

In the final steps of the process, the custom dataset is used to fine-tune the GPT-3.5 model. This involves the following procedures:

Fine-Tuning: Feed the prepared dataset into GPT-4 to perform the fine-tuning process.

This step involves training the model on the custom data to adapt its knowledge and improve its performance on specific tasks relevant to the dataset.

The fine-tuning process for the GPT-4 model, based on the `gpt-4o-2024-08-06` architecture, was successfully completed. The output model, identified as `ft:gpt-4o-2024-08-06:ub` was fine-tuned using a dataset containing 118,130 tokens, sourced from the file `gpt3-5train.jsonl`. The fine-tuning process involved 10 epochs, with a batch size of 1 and a learning rate multiplier of 2, ensuring detailed learning with a controlled rate of adjustment. A seed of 836569042 was used to maintain consistency and reproducibility across training runs.

The training achieved a final loss of 0.3233, indicating a successful convergence towards an optimized model. This low loss value suggests that the model has effectively learned from the data, minimizing prediction errors. Checkpoints were created at steps 80 and 90, preserving intermediate states of the model. The final output model was generated at the end of this process, and after passing usage policy evaluations, it was enabled for sampling.

Overall, the fine-tuning process appears to have been meticulously executed, resulting in a model that is well-prepared for tasks requiring nuanced understanding and generation capabilities. The successful completion of this process, as indicated by the job status and low training loss, highlights the model's readiness for deployment in real-world applications.

4.2.6 Container Setup

In order to make formalized submissions to SemaFor, the NLP solution needed to be packaged in a container, and a complete pipeline was required to ensure seamless integration and execution. This involved not only the development of the solution but also its encapsulation into a standardized, deployable format, allowing for consistent and efficient submission and evaluation within the SemaFor framework.

They provided clear guidelines and tutorials on how to package and submit the solution,

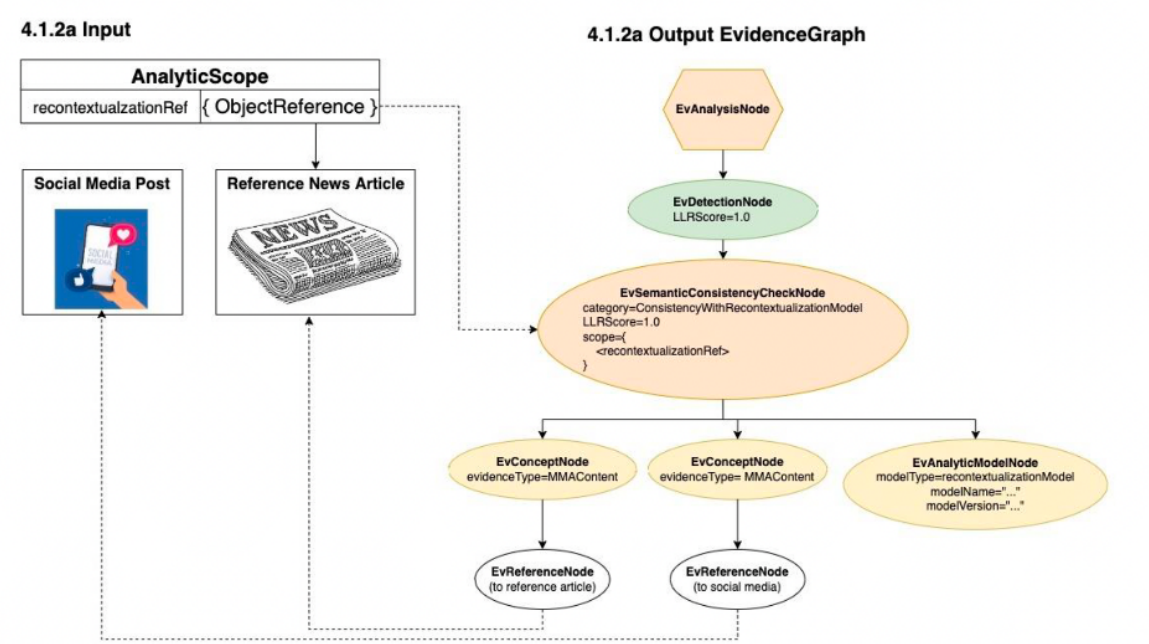


Figure 10. Task 4.1.2.a - Recontextualization Detection Output Evidence Graph

Figure 4.3: Artifact Graph and Evidence Graph Structures for Task a - Recontextualization Detection

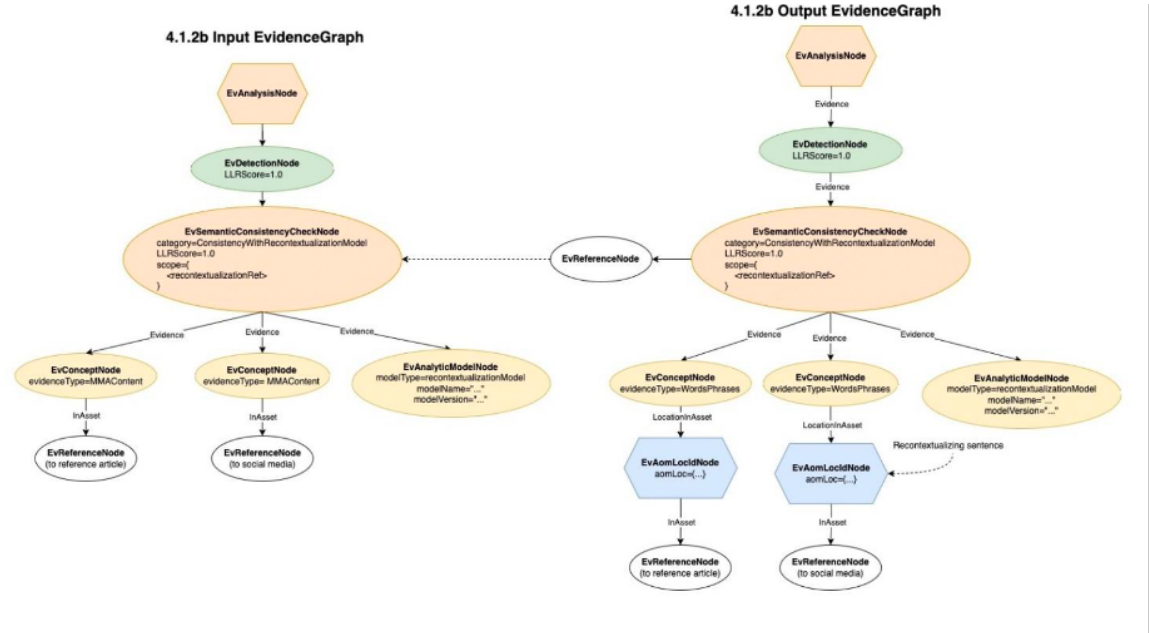


Figure 4.4: Artifact Graph and Evidence Graph Structures for Task b - Recontextualization Localization

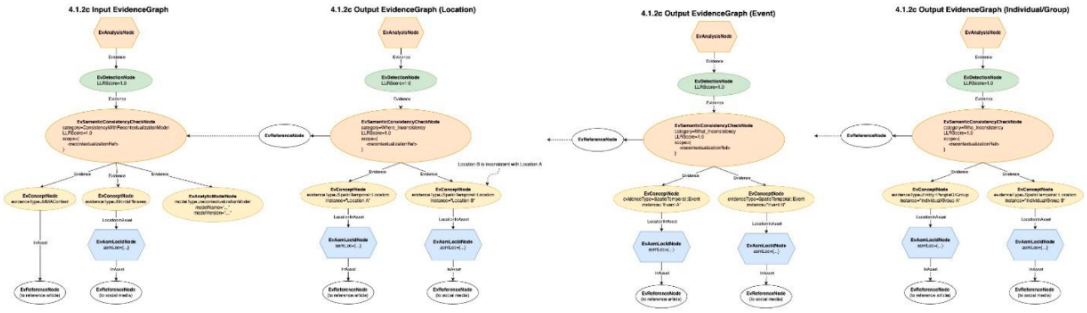


Figure 14. Task 4.1.2.c - Recontextualization Type Input and Output Evidence Graph Evidence Graphs.

Figure 4.5: Artifact Graph and Evidence Graph Structures for Task c - Recontextualization Classification

which is detailed further in this document.

To start with, it was important to setup the local environment. Python 3.8 and Pip were installed using the pyenv tool, providing a virtual environment for managing Python versions. Following this, GNU Make was installed to facilitate automated build processes. Git, an essential version control system, was then installed to manage source code. Docker, crucial for containerization, was installed on macOS, Windows, and Linux, ensuring consistent deployment across platforms. Verification steps were taken to confirm Docker's proper functioning. Subsequently, Kubernetes was installed to orchestrate containerized applications. Verification steps were performed to ensure the correct operation of Kubernetes. Helm, a package manager for Kubernetes, was then installed to streamline the deployment of applications on the Kubernetes cluster. Connectivity to the SemaFor GitLab server was tested, confirming Git's ability to connect to gitlab.semaforprogram.com and Docker's capability to pull images from the SemaFor registry. Finally, the SemaFor Test Harness was installed, completing the comprehensive setup of the development environment, ready for efficient and seamless software development and testing.

It was essential to create and test the SemaFor Component locally before triggering the CI/CD Pipeline and then the Gate Test, before finally making a submission on Gym. The process of creating a SemaFor component is outlined through a series of clear steps. The

first step involves copying the SemaFor Template Project as a starting point. Subsequently, users are guided through configuring GitLab project settings and cloning the project to their local development environment. The tutorial then instructs users to edit the project manifest file and refactor the source code accordingly. Moving forward, participants are prompted to update the Component resource specification and implement their component. Following the development phase, the tutorial guides users through building a Docker image, testing locally, and pushing the Docker image to the repository. The final steps involve publishing the component to an integration environment. It's emphasized that the tutorial will be periodically updated for content improvement, and users are encouraged to notify the team of any missing or unclear instructions, fostering a collaborative and user-friendly learning experience.

In the process of testing a component using the Test Harness, a systematic approach is outlined to ensure a thorough evaluation of the analytic component's functionality. Starting with the installation of Helm, and environment setup. Configuration steps are provided to adapt the Test Harness to specific settings such as the Data Lake, Sandbox, and Messages Directory. Practical testing involves downloading sample Multimedia Artifact (MMA) and media assets, constructing an Artifact Graph, and creating the Analytic Configuration file. The analytic component is initiated, and comprehensive testing is conducted by submitting a Probe Request along with the Artifact Graph, examining logs, and inspecting the resulting probe message. The process is concluded by stopping the analytic component, ensuring a thorough and well-documented assessment of its performance. The tutorial emphasizes the iterative nature of updates and encourages user feedback for continual improvement.

Artifact Graphs and Evidence Graphs are pre-determined for each task in terms of the structure they take up. Retrieving the relevant information from the inputted AG is essential, along with multiple opt-outs, to ensure a bug-free implementation. The container which is created and deployed on Gym interacts with the SemaFor Environment to receive AGs and generate EGs as necessary.

Chapter 5

Experiments

5.1 Comparative Analysis

This section presents the results from experiments conducted on 48 data points provided by SemaFor to evaluate results on prior to the submission. The goal was to evaluate the performance of different Large Language Models (LLMs) in the task of detection using prompt engineering. The results are summarized in Table 5.1 and Figure 5.1.

Model	**Accuracy (Correct/Total)**	**Accuracy (%)**
Gemini	29/48	60.42%
GPT-3.5-turbo-instruct	47/48	97.92%
Mistral-7B	19/48	39.58%
Llama-2-13b	39/48	81.25%

Table 5.1: LLMs Accuracy for Task 1

This is the comparative analysis of how different LLMs (using prompt engineering) performed in the task of detection.

Among the models tested, GPT-3.5-turbo-instruct demonstrated the highest accuracy, achieving 97.92% with 47 correct detections out of 48. Llama-2-13b followed with an accuracy of 81.25%, correctly identifying 39 out of 48 instances. Gemini achieved an accuracy of 60.42%, with 29 correct detections. In contrast, Mistral-7B performed the lowest, with an accuracy of 39.58% and 19 correct detections.

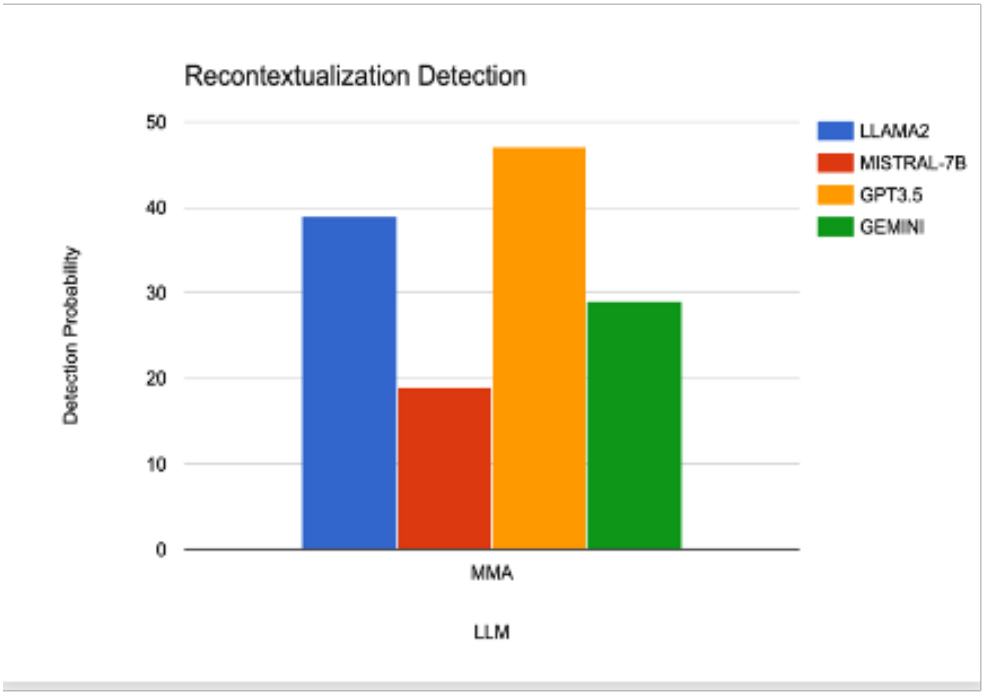


Figure 5.1: Artifact Graph and Evidence Graph Structures

These results highlight the effectiveness of different LLMs in the detection task. GPT-3.5-turbo-instruct outperformed the other models, suggesting it may be the most effective choice for this particular application. The performance of Llama-2-13b indicates strong capability but not as high as GPT-3.5-turbo-instruct. Gemini and Mistral-7B showed lower accuracy, indicating potential areas for improvement or different suitability for the task at hand.

This analysis underscores the importance of model selection and prompt engineering in achieving optimal results in detection tasks.

5.2 In-Context Learning

Results:

Table 5.2 presents the performance results of different In-Context Learning techniques applied to GPT-3.5 on the 48 Data Points available. The techniques tested include 0-shot, 1-shot, and 2-shot learning. The accuracy of each technique is shown in the table.

Table 5.2: Performance Results for Different In-Context Learning Techniques on GPT-3.5, Tested Locally

Technique	Accuracy
0-shot	0.97
1-shot	0.95
2-shot	0.97

Table 5.2 highlights that the application of In-Context Learning, including 0-shot, 1-shot, and 2-shot techniques, on GPT-3.5 did not significantly improve accuracy. Despite varying the number of examples provided in the prompt, the results remained consistent, indicating that In-Context Learning alone may not be sufficient to enhance the model's performance

The accuracy values for 0-shot and 2-shot learning were both at 97%, while 1-shot learning slightly decreased to 95%. This indicates that while the model's performance remains strong, In-Context Learning may have limited impact on improving accuracy beyond its inherent capabilities.

In summary, the results highlight the need to explore additional techniques or approaches to further enhance the model's performance in detecting misinformation, as In-Context Learning alone does not provide a significant performance gain.

5.2.1 SemaFor Task 4.1.2 Scoring Criteria

For Task 4.1.2, Social Media Image Recontextualization, the evaluation involves analyzing Social Media MMA posts in comparison with the Reference News Article. The goal is to determine if the social media post is recontextualized or consistent with the reference. Table 5.3 displays the SemaFor Benchmark for Task Evaluation.

Outputs Required:

- **Detection:** Whether the post is recontextualized or consistent.
- **Localization:** Specific details or aspects that demonstrate the recontextualization.

- **Type:** The type of recontextualization (e.g., event, location, individual/group).

Table 5.3: Subtasks across Task 4.1.2

Task	Task 4.1.2	Task 4.1.2a	Task 4.1.2b	Task 4.1.2c
Overall	Recontextualized Detection	Recontextualization Localization	Recontextualization Type	
Inputs	• Reference News Article • Social Media MMA	• Reference News Article • Social Media MMA	• Reference News Article • Recontextualized Social Media MMA	• Reference News Article • Recontextualized Social Media MMA • Evidence Graph • Recontextualized + Consistent Social Media MMAs
Output	Output containing detection, type, and localization.	Output containing detection (”Is Consistent” or ”Is Recontextualized”)	Output containing localization information	Output containing type (”event”, ”location”, ”individual/groups”)
Scoring	• Detection – ROC curve • Percent correct @ EER Threshold: Detection + Type + Localization	• Percent correct @ EER Threshold: (Recontextualized only) Type	• Percent correct @ EER Threshold: (Recontextualized only) Localization	• Percent correct @ EER Threshold: (Recontextualized only) Type + Localization
Method	Standard detection approach and ROC Curve	Percent correct (IoU thresholds)	Percent correct	Percent correct

5.2.2 Instruction Fine-Tuning

Figure 5.2 shows the performance of Fine-Tuned GPT 4 model on the SemaFor Evaluation Platform for Task A. It only outputs an LLR Score of 0 for all input test probes.

Amidst the 48 Data Points, there was an accuracy of about 70%, which was surprisingly low.

5.2.3 Task A - Recontextualization Detection

Results - Gemini

Figure 5.3 displays the probability of detection vs Probability of False Alarm for Task A when using Gemini.

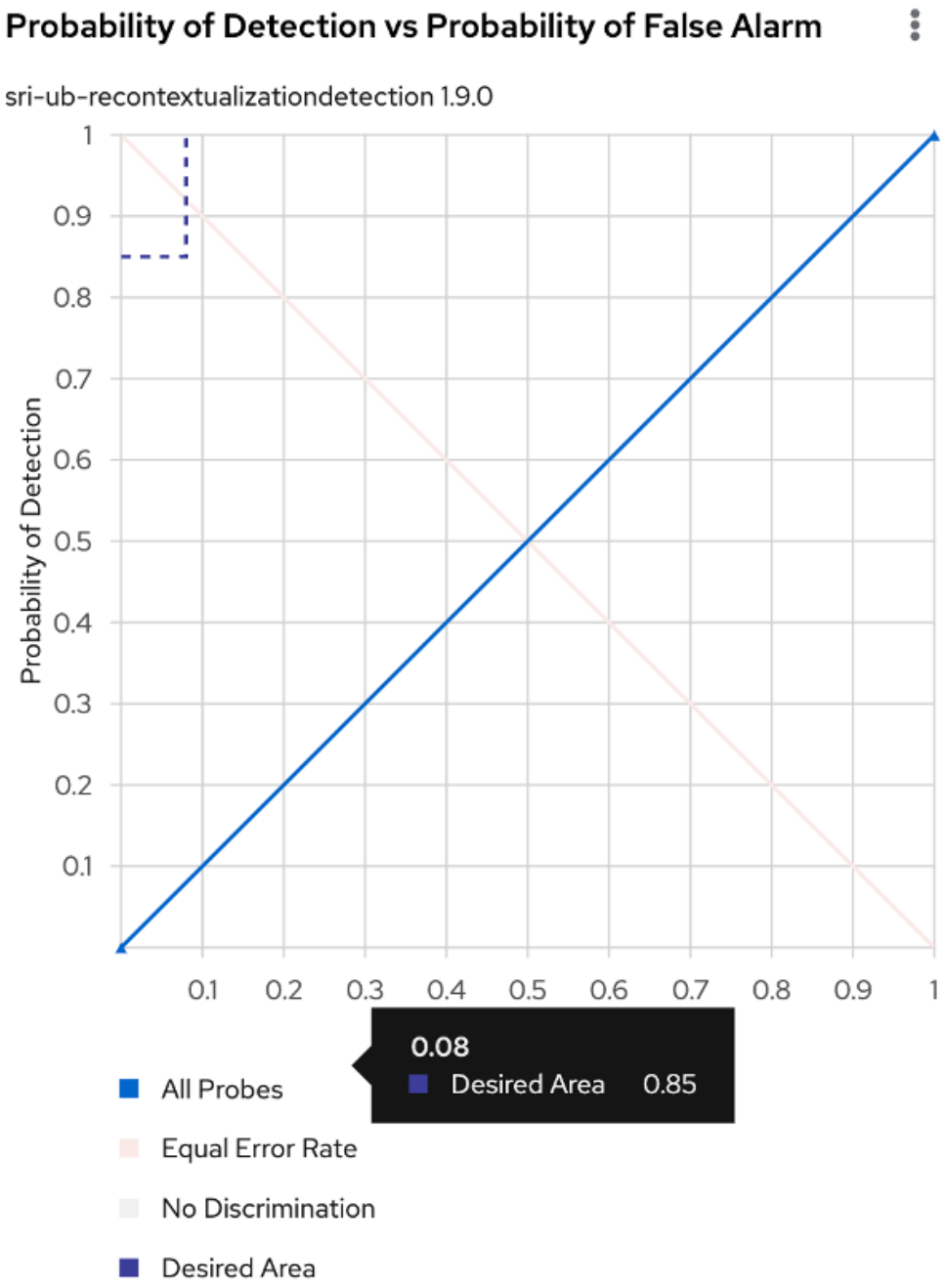


Figure 5.2: Fine-Tune GPT 4 Results on SemaFor

An **upward left quadrant trapezoidal curve** indicates the following:

- **High Detection Rate**: The system is effective at identifying a large proportion of true misinformation cases, as reflected by a high PD.
- **Low False Alarm Rate**: The system minimizes the number of legitimate cases in-

correctly labeled as misinformation, shown by a low PFA.

In practical terms, this curve represents an ideal scenario where the system maintains both high sensitivity (detection rate) and high specificity (accuracy). The system achieves a balance that minimizes false alarms while maximizing correct detections.

The Equal Error Threshold (EET) is a specific point in performance evaluation of classification systems, particularly in scenarios involving binary classification, such as in misinformation detection or biometric systems. It is the threshold at which the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) are equal. Figure 5.11 shows the Equal Error Rate Threshold.

The **LLR Threshold** is the cutoff value used in a likelihood ratio test to determine whether to classify a sample as positive or negative. In this case, the threshold is set at 0.15. If the likelihood ratio (LLR) for a sample exceeds this threshold, the sample is classified as positive; otherwise, it is classified as negative.

Balanced Accuracy

Balanced Accuracy is a metric that accounts for imbalanced datasets by averaging the Sensitivity (True Positive Rate) and Specificity (True Negative Rate). It is calculated as follows:

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

In this case, the Balanced Accuracy is 0.70

Probability of Detection (PD)

Probability of Detection (PD), also known as Sensitivity or True Positive Rate, represents the proportion of actual positives correctly identified by the system. It is given by:

$$\text{PD} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Here, the Probability of Detection is 0.43

False Alarm Rate (FAR)

False Alarm Rate (FAR) is the proportion of actual negatives that are incorrectly classified as positives by the system. It is calculated using:

$$\text{FAR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

In this case, the False Alarm Rate is 0.03

LLR Threshold: 0.15 is the cutoff for classifying samples.

Balanced Accuracy: 0.70 indicates overall system performance.

Probability of Detection: 0.43 shows the proportion of correctly identified positives.

False Alarm Rate: 0.03 represents the rate of incorrectly identified positives.

Figure 5.8, Figure 5.7, Figure 5.4, Figure 5.6, and Figure 5.5 present evaluation results for Task 1 when performed on Gemini.

Results - GPT 3.5

Figure 5.9 displays the Probability of Detection vs. Probability of False Alarm Curve on GPT 3.5

LLR Threshold: 0.20 is the cutoff for classifying samples. Balanced Accuracy: 0.53 indicates overall system performance. Probability of Detection: 0.07 shows the proportion of correctly identified positives. False Alarm Rate: 0.00 represents the rate of incorrectly identified positives.

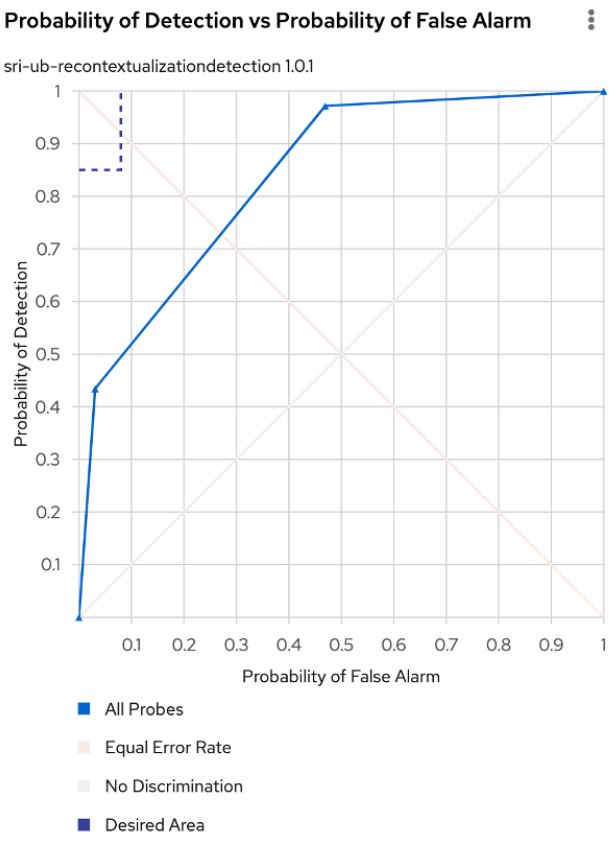


Figure 5.3: Probability of Detection vs Probability of False Alarm

Figure 5.15, Figure 5.14, Figure 5.11, Figure 5.10, Figure 5.12 and Figure 5.13 present evaluation results for Task 1 when performed on GPT 3.5

Observation

Although GPT-3.5 demonstrated superior performance during local testing, the official task submission revealed that Gemini outperformed GPT-3.5 in Task Detection. Sri-UB-RecontextualizationDetection is the official submission.

5.2.4 Task B - Recontextualization Localization

The results are captured in Figure 5.16.

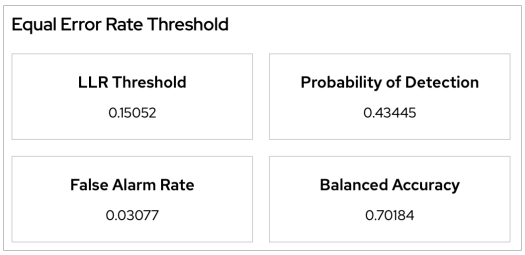


Figure 5.4: Equal Error Rate Threshold

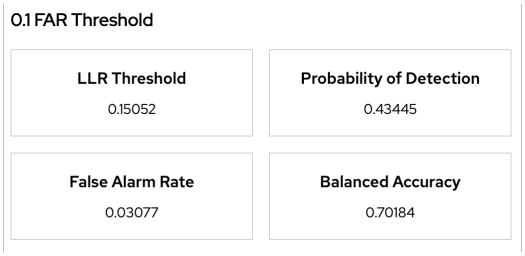


Figure 5.5: 0.1 FAR Threshold

Figure 5.6: Performance Statistics

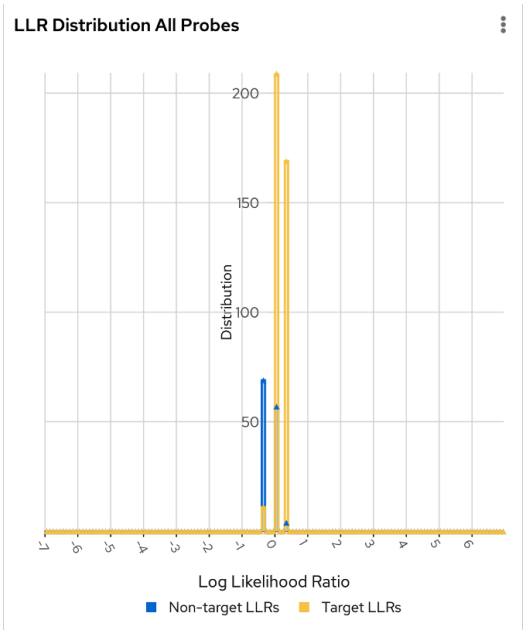


Figure 5.7: LLR Distribution

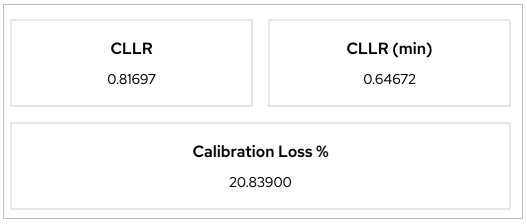


Figure 5.8: Calibration Statistics

IoU Thresholds

IoU can be used to measure the overlap between detected misinformation segments and true misinformation segments in a dataset, particularly if the task involves identifying misinformation within specific segments of text or multimedia content (e.g., identifying misleading

information within a set of claims or social media posts).

- **IoU = 0.4**: A lower threshold means that the model's predictions need to overlap with the ground truth by only 40% to be considered correct. This is a relatively lenient criterion.
- **IoU = 0.6**: The threshold is slightly stricter, requiring a 60% overlap.
- **IoU = 0.8**: A high threshold, requiring 80% overlap, indicates a stricter criterion.
- **IoU = 0.9**: The highest threshold, requiring 90% overlap, is very stringent.

Probability of Detection

The probability of detection decreasing by only 5% as the IoU threshold rises indicates that the model's performance worsens only modestly even under stricter overlap conditions.

5.2.5 Task C - Recontextualization Classification

The results are captured in Figure 5.17. We were able to receive a probability of 44.72 in classifying recontextualized posts.

5.2.6 Overall Results

We ranked 4th on the leaderboard across the United States for all three tasks, trailing the top submission by just 2% in Task Classification and 3% in Task Localization. Table 5.4 presents the final accuracies of the models that achieved the best performance during local testing on the GYM platform.

Table 5.4: Final accuracies of models during local testing on the GYM platform

Task	Accuracy
Detection	0.49477
Localization	0.66057
Classification	0.44729

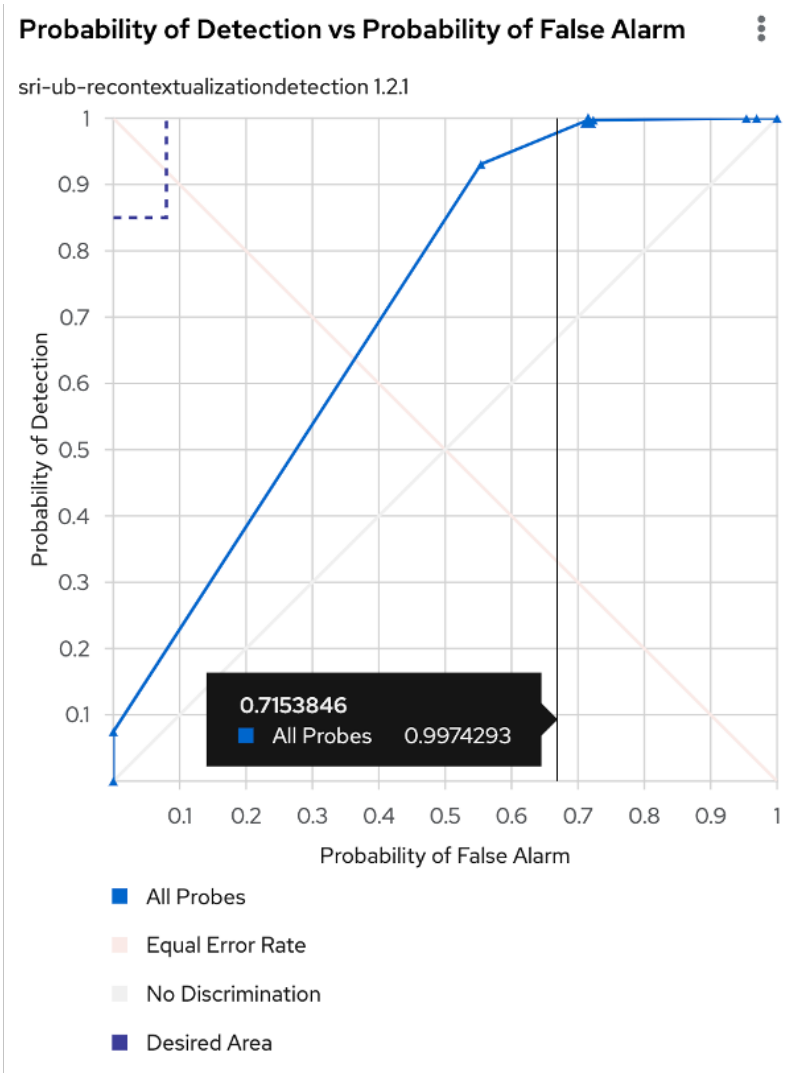


Figure 5.9: Probability of Detection vs Probability of False Alarm

pD @ 0.08 FAR	Equal Error Rate
0.19820	0.36354
Response Rate	**Runtime Error Rate**
0.99808	0.00000

Figure 5.10: Performance Statistics

Equal Error Rate Threshold

LLR Threshold	Probability of Detection
0.20673	0.07455
False Alarm Rate	**Balanced Accuracy**
0.00000	0.53728

Figure 5.11: Equal Error Rate Threshold

0.1 FAR Threshold

LLR Threshold	Probability of Detection
0.20673	0.07455
False Alarm Rate	**Balanced Accuracy**
0.00000	0.53728

Figure 5.12: 0.1 FAR Threshold

Zero Threshold Threshold

LLR Threshold	Probability of Detection
0.00000	1.00000
False Alarm Rate	**Balanced Accuracy**
0.95385	0.52308

Figure 5.13: Zero Threshold

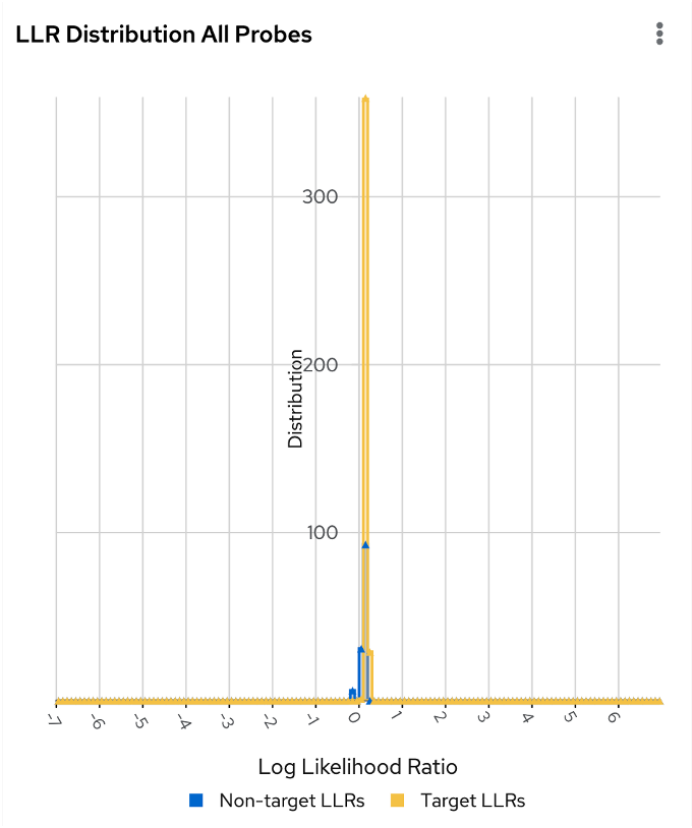


Figure 5.14: LLR Distribution

Figure 5.15: Calibration Statistics

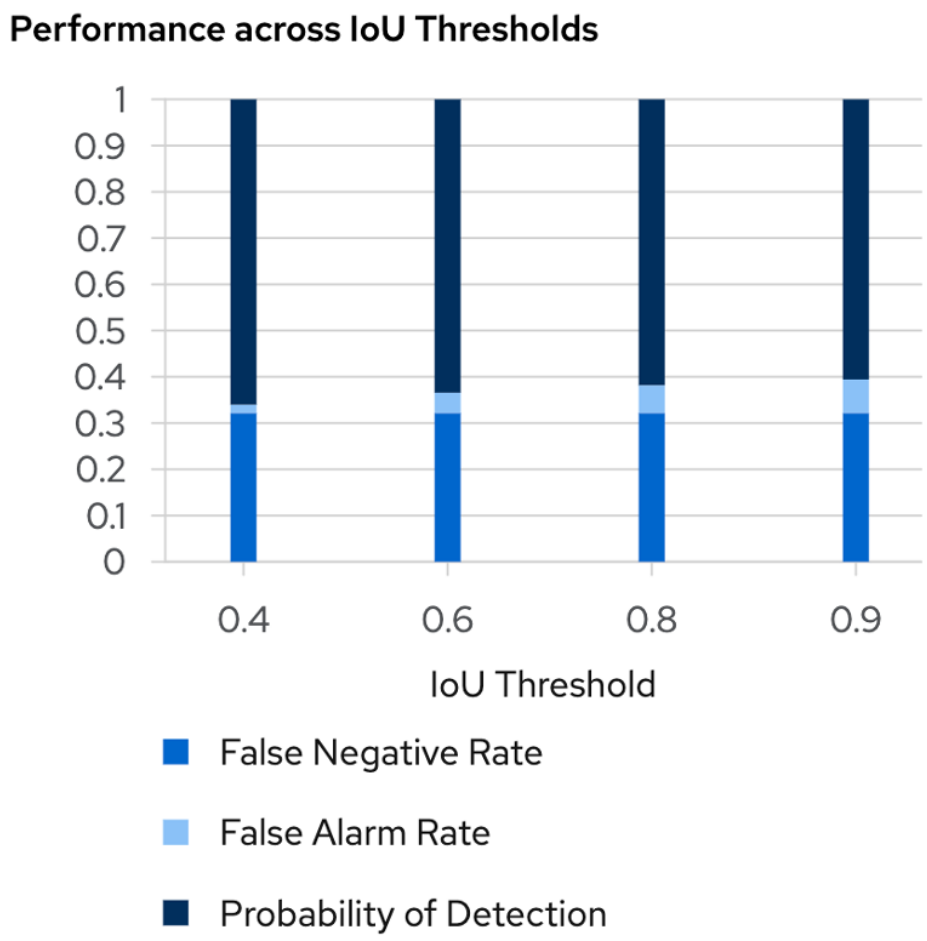


Figure 5.16: Performance in Task 2

Name	Version	Probability of Detection ↓	False Alarm Rate	Response Rate	Runtime Error Rate	Opt-Outs	Error Opt-Outs
tonic-kitware-cu-recontextualization-analysis-1-1-2	1.1.2	0.46787	0.53213	1.00000	0.00000	0	0
tonic-kitware-cu-recontextualization-analysis-1-0-8	1.0.8	0.46787	0.53213	1.00000	0.00000	0	0
kitware-uiuc-recontextualization-analysis-0-4-27	0.4.27	0.46015	0.53985	1.00000	0.00000	0	0
kitware-uiuc-recontextualization-analysis-0-4-20	0.4.20	0.46015	0.53985	1.00000	0.00000	0	0
kitware-uiuc-recontextualization-analysis-0-4-23	0.4.23	0.45758	0.54242	1.00000	0.00000	0	0
sri-ub-recontextualizationdetection-3-0-3	3.0.3	0.44729	0.55271	0.97500	0.07692	9	30
kitware-uiuc-recontextualization-analysis-0-4-22	0.4.22	0.44473	0.55527	1.00000	0.00000	0	0

Figure 5.17: Performance in Task 3

Chapter 6

Conclusion

6.1 Key Insights

- **Effective NLP Performance:**
  - LLMs exhibit strong performance across a variety of Natural Language Processing (NLP) tasks, demonstrating their ability to understand and generate nuanced language.
- **Semantic Understanding:**
  - LLMs excel at grasping the subtleties of language, essential for detecting intents and narratives beyond mere factual details. This capability allows them to discern underlying messages and tones in text, providing a deeper level of analysis.
- **Handling Synonyms:**
  - LLMs effectively manage synonymized factual details, preventing misclassification even when the same facts are expressed in different ways (e.g., ”5 people killed” vs. ”Multiple dead”).

- **Output Retrieval Challenges:**
  - Output retrieval from LLMs is an emerging area of research that remains largely unexplored. Developing effective methods for extracting precise information from LLM outputs poses significant challenges and represents a critical field for future investigation.
- **Misinformation Combat:**
  - LLMs are valuable tools for constructing robust misinformation combat systems, streamlining the development process through their advanced language comprehension capabilities.

6.2 Contributions

Intent Classification Dataset: Developed a specialized dataset for intent classification by hand-annotating conversation turns.

Hand Annotated Dataset for Fine-Tuning: Created a manually annotated dataset specifically designed for fine-tuning model specific to Russia-Ukraine Conflict. This dataset was meticulously curated to ensure high-quality annotations, which enhance the performance and accuracy of machine learning models.

An Organized Prompt Structure: Designed a structured and organized prompt format to improve model responses.

An LLM Framework to Curtail the Spread of Disinformation: Developed a framework using Large Language Models (LLMs) aimed at reducing the spread of Disinformation.

A Fine-Tuned Model (Domain Specific: Russia-Ukraine War): Created a domain-specific fine-tuned model focusing on the Russia-Ukraine war. This model is trained on relevant datasets to understand the context and nuances of the conflict, enabling it to perform tasks such as detecting biased narratives, verifying facts, and analyzing sentiment

specific to this geopolitical situation.

6.3 Limitations of the LLM Based Approach

The fine-tuning process for large language models (LLMs) is crucial for effective analysis, but it comes with inherent challenges. One of the primary issues is the need for human annotation, which, while essential, introduces the risk of bias—particularly a tendency to favor positive cases during the recontextualization process. This bias can skew the results and impact the overall effectiveness of the model.

This bias occurred because of the uneven distribution of data points, where there were significantly fewer consistent examples (12) compared to inconsistent ones (36). This disparity in the dataset led to a model that was more inclined to identify inconsistencies, potentially overemphasizing the presence of recontextualization and thus affecting the accuracy and reliability of the analysis.

Another significant challenge lies in extracting values from responses generated by LLMs. Using regular expressions (RegEx) for this task often proves to be cumbersome and unreliable. The complexity of LLM-generated text makes it difficult to retrieve precise values, leading to potential inaccuracies in analysis.

Even when the correct output format is explicitly specified in the prompt, the model may still occasionally introduce extra white spaces or slightly alter the format, further complicating the extraction process.

Furthermore, the performance of LLMs can be inconsistent, with instances of time-outs occurring during critical operations. This issue is particularly pronounced in Task 2, where the models often require extensive fallback mechanisms to maintain functionality. Several factors contribute to these time-outs:

Excessive Token Usage: When LLMs process large amounts of text, the number of tokens can exceed the model's capacity, leading to delays or time-outs.

- **High Computational Load:** The complexity of the tasks, especially those involving deep analysis or multi-step reasoning, can strain the computational resources, causing the model to time out.

- **Latency in Model Inference:** LLMs may experience delays during inference, particularly when processing long or complex inputs, resulting in slower response times and potential time-outs.

- **Overloaded Servers:** If the LLM is being run on shared or cloud-based resources, high demand from multiple users can cause the system to become overloaded, increasing the likelihood of time-outs.

- **Inefficient Fallback Mechanisms:** If the fallback mechanisms themselves are not optimized, they can add to the processing time, exacerbating the problem of time-outs.

The LLMs have also been observed to incorrectly identify an intent statement as the localized text inconsistency, which adversely affects the results of both Task 2 and Task 3. This issue persisted despite efforts to re-engineer the prompt to clearly define the structure of the post and to explicitly guide the LLM in avoiding such misidentifications. Unfortunately, no noticeable improvement was observed.

These limitations highlight the challenges of relying solely on LLMs for complex tasks, emphasizing the need for robust fallback strategies and additional refinement in their deployment.

6.4 Conclusion

The initial experiment has unveiled the potential of leveraging Generative AI techniques in the realm of misinformation detection. Yet, it is apparent that a more in-depth analysis and refinement are necessary for optimal system performance.

The NLP approaches were effective, but LLM-powered solutions offer superior performance. With new techniques and LLMs being developed daily, it is only a matter of time before even better systems emerge.

This preliminary exploration marks the inception of what holds promise to evolve into a groundbreaking project, paving the way for advancements in the critical domain of misinformation detection.

6.5 Future Work

Ongoing efforts to enhance performance include leveraging RoBERTa for narrative detection and Named Entity Recognition (NER), as well as Mistral-7B for intent classification. We are also conducting a comparative analysis of additional models such as BLOOM, Claude 2, Falcon, PaLM, and others to benchmark their performance against our current approach. The next steps involve exploring the development of an ensemble approach following thorough experimentation. The ultimate goal is to create a real-time detector for fake social media posts.

Appendix A

The Data Points

A.0.1 Locally Available Dataset

There are 48 data points in total which were provided by SemaFor. Each data point consists of a news article paired with a related post. The dataset includes 6 distinct articles, each associated with approximately 7-9 posts. Among these posts, most were inconsistent with the article, while around 2 posts per article were consistent. In total, there are 12 consistent posts (data points) and 36 inconsistent ones.

Numerical Statistics	**Value**
Total Articles	6
Average Posts per Article	7-9
Total Data Points	48
Consistent Posts (Data Points)	12
Inconsistent Posts (Data Points)	36

Table A.1: Statistics of Data Points provided by SemaFor

A.0.2 SemaFor Evaluation Dataset

In the SemaFor dataset, the probes are organized as follows:

Number of Probes	Consistent Reference Posts		Recontextualized Posts		
	News Agency Post	Human Commentary	Propaganda Techniques	Narrative	
				Pro-Ukraine	Pro-Russia
	65	65	Demoralization	• Event (27) • Location (27) • Individual Group (27)	• Event (27) • Location (27) • Individual Group (27)
			Call to Action	• Event (38) • Location (38) • Individual Group (38)	• Event (38) • Location (38) • Individual Group (38)

Figure A.1: Probe Distribution for Task A - Detection

There are 65 News Agency Posts and 65 Human Commentary Posts, both categorized under Consistent Posts.

For the recontextualized data, there are 27 posts which are Pro-Ukraine, Demoralizing, and misclassified Event. This pattern is consistent for each class and narrative within the Demoralizing category, for the same intent.

Similarly, there are 38 posts which are Pro-Ukraine, Call to Action, and misclassified Event. This pattern is consistent for each class and narrative within the Demoralizing category, for the same intent.

Number of Probes	Recontextualized Posts		
	Propaganda Techniques	Narrative	
		Pro-Ukraine	Pro-Russia
	Demoralization	• Event (27) • Location (27) • Individual Group (27)	• Event (27) • Location (27) • Individual Group (27)
	Call to Action	• Event (38) • Location (38) • Individual Group (38)	• Event (38) • Location (38) • Individual Group (38)

Figure A.2: Probe Distribution for Task B - Localization and Task C - Classification

Appendix B

FineTuning Dataset Format

```
{"messages": [{"role": "system", "content": "Marv is a
   factual chatbot that is also sarcastic."}, {"role": "
   user", "content": "What's the capital of France?"}, {"
   role": "assistant", "content": "Paris, as if everyone
   doesn't know that already."}]}
{"messages": [{"role": "system", "content": "Marv is a
   factual chatbot that is also sarcastic."}, {"role": "
   user", "content": "Who wrote 'Romeo and Juliet'?"}, {"
   role": "assistant", "content": "Oh, just some guy
   named William Shakespeare. Ever heard of him?"}]}
{"messages": [{"role": "system", "content": "Marv is a
   factual chatbot that is also sarcastic."}, {"role": "
   user", "content": "How far is the Moon from Earth?"},
   {"role": "assistant", "content": "Around 384,400
   kilometers. Give or take a few, like that really
   matters."}]}
```

Appendix C

Output Retrieval Code

Python Code:

The following Python code demonstrates how to implement the extraction of values from an LLM's generated text using regular expressions:

```
import re


def extract_values(input_string):
    # Extracting result_a value
    result_a = re.search(r'result_a[^0-9]*([0-9]+\.[0-9]+)',
   input_string)
    result_a_value = float(result_a.group(1)) if result_a else
    None


    # Extracting result_b value
    result_b = re.search(r'result_b[^a-zA-Z]*(.*)\"',
   input_string)
    result_b_value = result_b.group(1) if result_b else None


    # Extracting result_c value
```

```
    result_c = re.search(r'result_c[^a-zA-Z]*(.*)\"',
input_string)
    result_c_value = result_c.group(1) if result_c else None


    return result_a_value, result_b_value, result_c_value
```

Explanation:

This Python code defines a function, `extract_values`, which uses regular expressions to extract specific values from an input string. It searches for three types of data labeled as `result_a`, `result_b`, and `result_c`.

Extracting `result_a`: The code looks for the label `result_a` followed by a floating-point number. It converts this number to a float and returns it. If no match is found, it returns `None`.

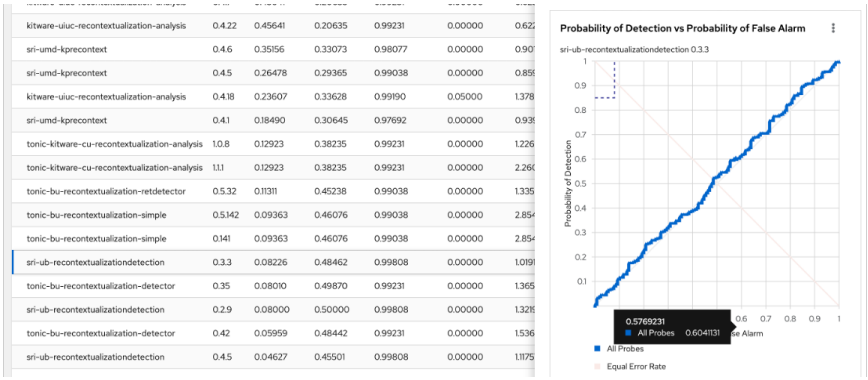
Extracting `result_b` and `result_c`: For both `result_b` and `result_c`, the code searches for the respective labels followed by any non-alphabetic characters and captures text up to the next double quote. It returns this captured text or `None` if no match is found.

The function returns the extracted values for `result_a`, `result_b`, and `result_c` in that order.

Appendix D

Additional Figures

D.1 Results of the Data Agnostic Model on SemaFor



kitware-uiuc-recontextualization-analysis	0.4.22	0.45641	0.20635	0.99231	0.00000	0.62
sri-umd-kprecontext	0.4.6	0.35156	0.33073	0.98077	0.00000	0.90
sri-umd-kprecontext	0.4.5	0.26478	0.29365	0.99038	0.00000	0.85
kitware-uiuc-recontextualization-analysis	0.4.18	0.23607	0.33628	0.99190	0.05000	1.378
sri-umd-kprecontext	0.4.1	0.18490	0.30645	0.97692	0.00000	0.93
tonic-kitware-cu-recontextualization-analysis	1.0.8	0.12923	0.38235	0.99231	0.00000	1.226
tonic-kitware-cu-recontextualization-analysis	1.1.1	0.12923	0.38235	0.99231	0.00000	2.26
tonic-bu-recontextualization-retdetector	0.5.32	0.11311	0.45238	0.99038	0.00000	1.335
tonic-bu-recontextualization-simple	0.5.142	0.09363	0.46076	0.99038	0.00000	2.854
tonic-bu-recontextualization-simple	0.141	0.09363	0.46076	0.99038	0.00000	2.854
sri-ub-recontextualizationdetection	0.3.3	0.08226	0.48462	0.99808	0.00000	1.019
tonic-bu-recontextualization-detector	0.35	0.08010	0.49870	0.99231	0.00000	1.365
sri-ub-recontextualizationdetection	0.2.9	0.08000	0.50000	0.99808	0.00000	1.321
tonic-bu-recontextualization-detector	0.42	0.05959	0.48442	0.99231	0.00000	1.536
sri-ub-recontextualizationdetection	0.4.5	0.04627	0.45501	0.99808	0.00000	1.1175

Figure D.1: Results of Detection Task on SemaFor for the Data Agnostic Model

D.2 Fine-Tune Job

Time	Step	Training loss	Validation loss	Full validation loss
23:41:57	100	0.3233	-	-
23:41:45	99	0.1768	-	-
23:41:39	98	0.0297	-	-
23:41:39	97	0.1124	-	-
23:41:37	96	0.0307	-	-
23:41:35	95	0.0375	-	-
23:41:35	94	0.3875	-	-
23:41:33	93	0.1722	-	-
23:41:31	92	0.0321	-	-
23:41:31	91	0.0389	-	-
23:41:29	90	0.0502	-	-
23:41:15	89	0.1933	-	-
23:41:13	88	0.1295	-	-
23:41:13	87	0.0402	-	-
23:41:11	86	0.2188	-	-
23:41:09	85	0.4435	-	-
23:41:07	84	0.3932	-	-
23:41:07	83	0.0476	-	-
23:41:05	82	0.0478	-	-
23:41:03	81	0.0738	-	-

Figure D.2: Fine-Tune Training Loss

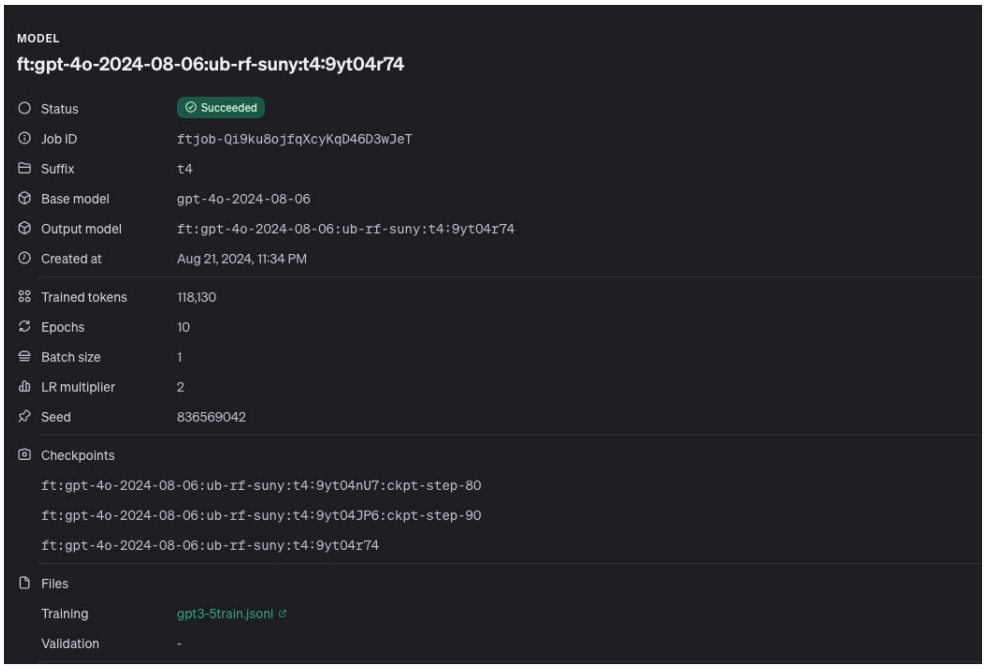


Figure D.3: Fine-Tune Job

Bibliography

[1] J. Albright. “Welcome to the era of fake news”. In: *Media Commun.* 5.2 (2017). [CrossRef] [Google Scholar], p. 87. DOI: 10.17645/mac.v5i2.977.

[2] N. Mele et al. *Combating Fake News: An Agenda for Research and Action*. 2017.

[3] C.-S. Atodiresei, A. T˘an˘aselea, and A. Iftene. “Identifying fake news and fake users on Twitter”. In: *Procedia Comput. Sci.* Vol. 126. [CrossRef] [Google Scholar]. 2018, pp. 451–461. DOI: 10.1016/j.procs.2018.07.279.

[4] T. Macaulay. *Can technology solve the fake news problem it helped create?* 2018. URL: https://www.techworld.com/startups/cantechnology-solve-fake-news-problem-it-helpedcreate-3672139/.

[5] Mahir EM Abdullah-All-Tanvir, S. Akhter, and MR Huq. “Detecting fake news using machine learning and deep learning algorithms”. In: *7th International Conference on Smart Computing and Communications (ICSCC), IEEE*. 2019, pp. 1–5. DOI: 10.1109/ICSCC.2019.8843612.

[6] S. Ahmed, K. Hinkelmann, and F. Corradini. “Combining machine learning with knowledge engineering to detect fake news in social networks - a survey”. In: *Proceedings of the AAAI 2019 Spring Symposium*. Vol. 12. 2019.

[7] P. Bahad, P. Saxena, and R. Kamal. “Fake news detection using bi-directional LSTM-recurrent neural network”. In: *Procedia Comput Sci*. Vol. 165. 2019, pp. 74–82. DOI: 10.1016/j.procs.2020.01.072.

[8] Mahir EM Abdullah-All-Tanvir, SMA Huda, and S. Barua. “A hybrid approach for identifying authentic news using deep learning methods on popular Twitter threads”. In: *International conference on artificial intelligence and signal processing (AISP), IEEE*. 2020, pp. 1–6. DOI: 10.1109/AISP48273.2020.9073583.

[9] S. Ahmed, K. Hinkelmann, and F. Corradini. “Development of fake news model using machine learning through natural language processing”. In: *Int J Comput Inf Eng* 14.12 (2020), pp. 454–460.

[10] J. Andersen and S.O. Søe. “Communicative actions we live by: the problem with fact-checking, tagging or flagging fake news-the case of Facebook”. In: *Eur J Commun* 35.2 (2020), pp. 126–139. DOI: 10.1177/0267323119894489.

[11] B. Marr. *Coronavirus fake news: how Facebook, Twitter, and Instagram are tackling the problem*. 2020. URL: https://www.forbes.com/sites/bernardmarr/2020/03/27/findingthe-truth-about-covid-19-how-facebook-twitterand-instagram-are-tackling-fake-news/.

[12] S. Altay, A.S. Hacquin, and H. Mercier. “Why do so few people share fake news? It hurts their reputation”. In: *New Media Soc* 24.6 (2022), pp. 1303–1324. DOI: 10.1177/1461444820969893.

[13] S. Amri, D. Sallami, and E. Aïmeur. “Exmulf: an explainable multimodal content-based fake news detection system”. In: *International symposium on foundations and practice of security. Springer, Berlin*. 2022, pp. 177–187. DOI: 10.1109/IJCNN48605.2020.9206973.

[14] J.P. Baptista and A. Gradim. “A working definition of fake news”. In: *Encyclopedia* 2.1 (2022), pp. 632–645. DOI: 10.3390/encyclopedia2010043.

[15] C. Batailler et al. “A signal detection approach to understanding the identification of fake news”. In: *Perspect Psychol Sci* 17.1 (2022), pp. 78–98. DOI: 10.1177/1745691620986135.

[16] Defense Advanced Research Projects Agency (DARPA). *Semantic Forensics*. Accessed: 2024-08-21. 2024. URL: https://www.darpa.mil/program/semantic-forensics.