CIS*6580
Enhancing Usability of Large Language Models (LLMs) in Detecting Fake News: A Heuristic Walkthrough Evaluation Study

1st Sondos Azzouz
MCTI Program
University of Guelph
Guelph , Canada
azzouzs@uoguelph.ca

2nd Papa Orleans
MCTI Program
University of Guelph
Guelph, Canada
porleans@uoguelph.ca

3rd Enyu Ma
MCTI Program
University of Guelph
Guelph, Canada
ema03@uoguelph.ca

***Abstract*—Large Language Models (LLMs) have advanced significantly in recent years, offering enhanced capabilities to the public. Detecting fake news is crucial in today's information-rich environment. Due to their robust learning abilities, LLMs have the potential to differentiate between fake and real news effectively. This paper evaluates the effectiveness of LLMs—ChatGPT, Gemini, Cohere, Copilot, and Meta AI—in detecting fake news. We assess their performance by identifying usability gaps and suggesting improvements. The study includes a user survey and interviews to collect insights and recommendations. Heuristic evaluations and user studies were used to analyze performance and usability issues. The findings reveal challenges and propose enhancements for these LLMs. This research contributes to the ongoing efforts to utilize LLMs in combating misinformation, aiming to enhance their accuracy and usability for end-users.**

1. Introduction

From weather reports and fashion to sports and entertainment, news reports have provided vast amounts of valuable information to people worldwide. The invention of the internet, with its increased availability and accessibility, has allowed news reports and content to reach more people than ever before. However, this accessibility has also led to the proliferation of fake news, which is fabricated information that mimics news media content in form but not in organizational process or intent [1]. Fake news can cause significant, irreparable damage to the intended targets, making it crucial to detect and mitigate it swiftly. Existing methods for detecting fake news include manual fact-checking and automatic fact-checking. Manual fact-checking involves human experts verifying information, which is accurate but labor-intensive and unable to keep pace with the rapid spread of information online. Automatic fact-checking, on the other hand, utilizes Natural Language Processing (NLP) and Machine Learning (ML) techniques to assess the veracity of information [2]. While these methods have advanced the field, they often struggle with accurately capturing the context, nuance, and subtlety inherent in human language, leading to limitations in detection accuracy. Large Language Models (LLMs) present a promising solution to these challenges. LLMs are advanced language models with massive parameter sizes and exceptional learning capabilities [3]. They have evolved from statistical modeling to neural language modeling [4] and now to pre-trained language models with the capacity to process and interpret complex natural language structures. By leveraging vast datasets and sophisticated algorithms, LLMs can detect fake news by identifying subtle patterns and cues that traditional methods might miss. Their ability to understand context, recognize linguistic nuances, and learn from diverse information sources makes them particularly well-suited for this task. The main contributions of this paper are as follows:

- An overview of all selected LLMs: ChatGPT, Copilot, MetaAI, Cohere, and Gemini.
- A review of the TruthSeeker dataset that will be used.
- A heuristic walkthrough and a user study evaluating the usability of the interfaces of selected LLMs.
- A presentation of results and usability issues.

2. Related Work

In our research for this project, we reviewed several studies that provided valuable insights and helped refine our work. This section discusses and analyzes these related works, beginning with a study on Large Language Models (LLMs). Chang et al. [3] define LLMs as advanced language models with massive parameter sizes and exceptional learning capabilities . Their study conducts a survey on the evaluation of LLMs and outlines the essential aspects of evaluating LLMs: what to evaluate, where to evaluate, and how to evaluate. "What to evaluate" involves identifying tasks that highlight an LLM's strengths and weaknesses. "Where to evaluate" emphasizes the importance of diverse benchmarks for assessing the datasets used in training LLMs. "How to evaluate" discusses the methods of

automatic and human evaluation to determine LLM performance. Chang et al. summarize the success and failure cases of LLMs. Successes include strong performance in generating fluent and coherent text, handling factual information accurately, and excelling in tasks like sentiment analysis and text classification. However, challenges remain in tasks involving abstract reasoning, semantic similarity, and non-Latin scripts. LLMs also exhibit biases and occasionally produce toxic content, indicating a need for improved ethical safeguards. The survey reveals that no single evaluation protocol or benchmark is universally superior; effectiveness is task-dependent. Fine-tuned models often outperform zero-shot models in task-specific evaluations. Future research challenges identified by the authors include designing AGI benchmarks for dynamic environments, enhancing LLM robustness to handle diverse inputs, and creating unified evaluation frameworks that support a wide range of tasks and provide principled, trustworthy assessments of LLM performance. Jurgell and Borgman [5] address the growing accessibility of news and the accompanying rise in misinformation. They developed an open-source fake news detection tool using an LLM. Their methodology follows a design science research (DSR) approach, integrating user-centered design principles to tackle misinformation. The first step involves identifying the problem of fake news and its complexities by reviewing existing literature. Objectives are set based on user requirements, LLM capabilities, and societal impacts. The tool's development includes selecting a suitable LLM, creating algorithms for reliability and bias detection, and integrating these into a user-friendly interface. An iterative design process allows continuous refinement based on testing and feedback. The tool is then tested in real- world settings to gather practical insights, and its performance is evaluated against predefined goals using metrics like Accuracy of Claims (AoC), Logical Consistency (LC), and Bias and Objectivity (BaO). The evaluation uses datasets such as "FA-KES" and "GonzaloA/Fake_News_TFG." Key findings from the "FA-KES" dataset show an overall accuracy of 51.32%, with high recall in identifying real articles but challenges in precision and bias detection. Real articles scored higher in AoC and LC, while fake articles showed the opposite trend. The "GonzaloA / Fake_News_TFG" dataset evaluation revealed improved accuracy at 78.86%, demonstrating better performance with tabloid-style news. The tool showed a balanced performance across all metrics, with a perfect score in BaO for fake articles, indicating strong bias identification capabilities. In summary, Jurgell and Borgman's study offers valuable insights into creating an accessible fake news detection tool using LLMs. Their method emphasizes transparency and continuous improvement, highlighting the tool's strengths and areas for enhancement, particularly in bias detection and accuracy across diverse news types [5]. Another significant study by Zhang et al. [6] explored three main questions: the construction of cybersecurity-oriented domain LLMs, potential applications of LLMs in cybersecurity, and existing challenges and future research directions . This study undertakes a systematic literature review to provide a comprehensive overview of the application of LLMs in cybersecurity. An extensive search and analysis of over 180 academic papers, published since 2023, was conducted. The review covered a diverse range of LLMs, including both open- source and closed-source models. This approach ensured the inclusion of the latest advancements and a broad spectrum of LLM applications. The systematic review identified several critical areas where LLMs have shown promise in cybersecurity. These include vulnerability detection, secure code generation, program repair, binary analysis, IT operations, threat intelligence, anomaly detection, and LLM-assisted offensive security. However, the review also highlighted significant challenges, such as the need for comprehensive methodologies, scalability, reproducibility, and addressing LLM vulnerabilities and susceptibilities to attacks. Through these studies, we gained a deeper understanding of the potential and limitations of LLMs across various fields, which guided the development and refinement of our fake news detection tool. The insights from Chang et al. helped shape our evaluation framework, while Jurgell and Borgman's approach influenced our methodology and iterative design process. Zhang et al.'s findings underscored the importance of addressing LLM vulnerabilities and ensuring the robustness and reliability of our tool in detecting misinformation.

3. Methodology

This section provides an overview of the LLM tools used in the evaluation, describing their features and functionalities. This is very crucial to understand the LLM tools to comprehend their diverse applications. Additionally, we discuss the selected dataset used to evaluate the usability of LLMs in detecting fake news content. Finally, we will explain the heuristic walkthrough approach and the user study, aiming to offer a comprehensive evaluation of the LLMs.

3.1. Tools

This evaluation covers a variety of LLM tools including ChatGPT, Gemini, Cohere, Copilot, and Meta AI. These tools offer varying functionalities and features, which depend on factors such as whether accessing the free or paid version. ChatGPT is a general-purpose conversational AI that requires a lot of processing power because it was trained on a variety of datasets with different text kinds. Gemini blends reinforcement learning and language models, it requires more intricate training procedures and interacting datasets, which both complicate and enable adaptive learning. Cohere is highly effective at processing text but less flexible with non-text data since it is focused on NLP tasks and employs specialized text datasets. Large volumes of publicly available code are used to train Copilot, improving its code recommendation capacity. However, if improperly handled, Copilot might unintentionally spread unsafe coding techniques. Meta AI makes use of large and varied datasets to conduct thorough AI research.

3.1.1. ChatGPT

An adaptable conversational AI model is ChatGPT from OpenAI. There are 200 million active users of ChatGPT each month globally [5]. In order to create a comprehensive knowledge of language, ChatGPT uses a wide and diverse dataset that includes material from the internet, books, papers, and other publicly available information for training. The interaction with users is real time and it provides immediate feedback. We can also customize the instruction of ChatGPT. You can submit any information you would want ChatGPT to take into account when responding with custom instructions. Future interactions will include your personalized instructions. On Web, iOS, and Android, custom instructions are available for all plans [6]. Important features include the capacity to comprehend and produce natural language, handle contextual discourse, and adjust to different language activities including writing creatively, summarizing texts, and responding to queries. One of ChatGPT's strongest points is its versatility. ChatGPT's user interface (UI) is designed to be simple to use and intuitive, allowing for easy communication with the AI. Usually, it has a simple, minimalist layout with a text input box in the center where users may submit their commands or questions. What's more, ChatGPT supports variety of data type input which includes text file, image file and even programming file and so on. That massively facilitate the use of users by decreasing their workload in correlating data.

3.1.2. Gemini

Google DeepMind's Gemini is an artificial intelligence system that combines reinforcement learning and sophisticated language models to provide more dynamic and interactive features. Large-scale text datasets like to those used for language models are combined with interactive datasets that mimic real-world situations in Gemini's training process. Traditional supervised learning is used in the training process to interpret English, and reinforcement learning is included to help the model learn from feedback and interactive encounters. With the help of this combined method, Gemini can comprehend and produce human-like writing while also learning to make choices and act according to contextual knowledge. Gemini support CSV, DOC, DOCX, DOT, DOTX, PDF, RTF, TSV, TXT, XLS, XLSX and image types of file upload. Among the key features are natural language processing, context-aware dialogue management, and interactive, adaptive learning. Because of its capabilities, Gemini can handle challenging tasks including language understanding and decision-making, making it ideal for interactive storytelling, virtual assistants, and instructional materials. However, because to the intricacy of its training procedure, substantial computer power and advanced data management are required. Gemini is a popular AI system. In May 2024, Google Gemini had 313.9 million visitors overall. In particular, there were 143.1 million desktop visits and 170.8 million mobile visits [7]. Gemini uses complex algorithms to handle massive amounts of data, comprehend the context of user inquiries, and produce precise, related answers. The platform also offers user-friendly feedback and visuals, which help users make efficient decisions. In summary, Gemini can handle interactive and context-sensitive tasks well because of its combination of language and reinforcement learning but training it will cost a lot of money.

3.1.3. Co-pilot

Microsoft 365 Word, Excel, PowerPoint, Outlook, Teams, and other Microsoft 365 apps are linked with Copilot, an AI-powered assistant. By utilizing sophisticated language models that have been trained on extensive datasets, such as text from emails, documents, and other data from the Microsoft ecosystem, Copilot increases productivity through smart automations and recommendations. To communicate with users and datasets, it makes use of advanced machine learning models and NLP. Supervised and unsupervised learning strategies are used in the training process to comprehend user behavior, context, and particular activities linked to office efficiency. Text generation, document summarization, data visualization, task automation, contextual insights, and suggestion generation are some of the key features. Copilot integrates easily into the user's workflow and has the ability to compose emails, prepare presentations, generate reports, and analyze data. By cutting down on the amount of time spent on repetitive operations, its features are intended to increase productivity, foster innovation, and simplify processes. Using Microsoft Azure OpenAI Service, Copilot makes it easier to develop, manage, improve, and debug applications and infrastructure from the cloud to the edge [8]. Besides, Co-pilot supports various type of data upload as well with clean and intuitive interface.

3.1.4. Cohere

Using Transformer architecture and supercomputer training, Cohere's state-of-the-art big language models offer NLP solutions without the requirement for costly machine learning development [9]. To get a profound knowledge of language, the models are trained on vast text datasets, such as books, articles, webpages, and other large-scale corpora. Training entails both supervised fine-tuning for NLP tasks like sentiment analysis, text categorization, and semantic search, as well as unsupervised learning, where the models discover patterns and structures in the data. Text creation, summarization, categorization, and entity recognition are among the features offered by Cohere. By using straightforward API calls, developers can include these features in their apps and make use of robust NLP capabilities without having to start from nothing when training models. Scalability, simplicity of integration, and high-performance text processing are some of Cohere's key advantages, which make it perfect for use in data analysis, customer support, and content production applications. Though mostly restricted to text-based jobs, Cohere's main benefit is its ability to offer reliable NLP services that are simple to install and expand. In conclusion, Cohere excels in providing scalable and effective NLP capabilities via API access, enabling developers

to add advanced language production, and understanding features to their applications. What's more, The feature Command R+ supports many different languages. Cohere provides .PDF and .TXT file upload options with 20 MB maximum size limitation.

3.1.5. Meta AI

The goal of Meta AI, the artificial intelligence research branch of Meta (previously Facebook), is to advance AI technology in various domains, such as computer vision, robotics, NLP, and more. Meta said in Q3 2023 that they have surpassed 3.05 billion monthly active users [10]. When interacting with users, Meta AI is able to provide creative content, prediction, and make automated decision-making convenient. A disadvantage is that Meta AI does not support file upload. Fairness, accountability, and transparency are all incorporated into the models and applications of Meta AI, which stresses the ethical use of AI. Innovative solutions for social media, virtual reality, and other domains are offered by features including scalability, real-time processing, and interaction with Meta's social networks, which improve user experience. The main benefits of Meta AI are its broad research scope, which produces innovative and adaptable AI applications; yet its resource-intensive structure and emphasis on large-scale deployments might pose difficulties. In conclusion, Meta AI emphasizes scalability and ethical concerns for useful and significant applications while driving innovation across a range of AI areas through in-depth research.

3.2. Heuristic Walkthroughs

The heuristic walk-through is a two-phase method that combines the advantages of two usability evaluation techniques, the cognitive walkthrough (phase 1) and heuristic evaluation (phase2). In cognitive walkthrough, a task driven approach where evaluators will simulate a set of predefined tasks and feed them to the LLMs. This helps identifying usability issues based on how the LLMs perform these tasks. In heuristic evaluation, the evaluators will base their findings on heuristic measures. They will assess the LLMs against established usability principles. By using the heuristic walkthrough, which combines both phases, will ensure thoroughness and validity in identifying all usability issues. To reduce confusion, in this paper, we will refer to phase 1 as task-oriented evaluation and phase 2 as free-form evaluation [11]. The benefits of the heuristic walkthrough method rely on providing a comprehensive evaluation based on a dual approach, which proves to be more efficient than using a single method. Moreover, the method is adaptable to different evaluation environments, whether textual or graphical. Its structured framework guides the evaluation process, allowing both experts and non-experts to participate as evaluators [12]. This makes it an ideal method for evaluating the usability of LLMs in detecting fake news content by examining the user interface and functionality of the LLMs.

3.2.1. Phase 1: Task-Oriented Evaluation

In this phase, the goal is to identify any usability issue that a regular user might encounter when performing a specific task during typical usage scenarios. Evaluators will complete designated tasks and document usability problems. The main task that actual users would perform with the LLM for fake news detection is (use the LLM tool to classify the tweets as real or fake). The following guidelines are essential to navigate the task:

- classify tweets as real or fake news and observe LLM outputs for predictions.
- explain the reasoning behind LLMs classification for a specific tweet.
- Compare and contrast the LLM's analysis of two different tweets.
- Compare LLM's predictions with the corresponding "Majority_target" label to assess its accuracy.

Evaluators will also record observations during this phase and note down the following:

- Ease of use for analyzing the tweets with LLM
- Clarity of LLM outputs and how well that represent fake news classification.
- Usability issues encountered while interacting with the LLM when performing the task.
- System limitations that might prevent or harden the fake news analysis task including interface design issues.
- Challenges in formulating queries for LLM.

To understand the user experience and the effectiveness of the LLM, the Sear's guideline questions [13] were used. The questions focus on key usability areas such as learnability, efficiency, errors, and satisfaction. The list of selected questions from the Sear's guideline will be included in appendix A.

3.2.2. Phase 2: Free-Form Evaluation

In this phase, the evaluator's role is exploring the LLM more freely using pre-defined heuristics to oversee the evaluation. Using ChatGPT, we used a set of heuristics tailored to evaluating the usability of LLMs in detecting fake news. Moreover, these heuristics are adapted from both generic and domain-specific principals to make them relevant to the context of LLMs and fake news detection. In Appendix D, table **??** provides the list of heuristics and their descriptions. The evaluators will use the principles and record any observations based on the chosen heuristics. They will also record any usability issues identified during this evaluation. By the end of the evaluation, we should be able to categorize the identified usability issues based on different metrics such as severity, and frequency to help prioritize areas for improvements.

3.3. User Study

In our assessment of the five chosen LLM tools for detecting fake news, we carried out a a user study to corroborate the findings from the preliminary heuristic evaluation. We enlisted six participants for the study, referred to as P1–P6, with varying levels of familiarity with the LLMs and experience in detecting fake news. Participants were interviewed about their knowledge of the five LLMs under evaluation (ChatGPT, Cohere, Copilot, Meta, and Google Gemini) as well as their familiarity with the concept of fake news. Table 3 summarizes their responses. In brief, participants self-reported familiarity with LLMs (median 3/5) and fake news detection (median 4/5). ChatGPT and Gemini were the most recognized models among the group. We presented each participant with tasks and asked them to input fake news content items,ensuring a balanced ratio between real and fake items for each model. Participants engaged with each LLM for approximately 20 minutes, including time for inputting news items and receiving outputs and addressing any errors that arose. We encouraged participants to share their thinking process with us while performing the tasks. Following the task, we conducted semi-structured post-task interviews to gather detailed feedback on their experience with each model. At the end of the session, we collected demographic data. In the post-task interviews, participants described their experience with the LLMs, focusing on the models' ability to detect and differentiate between fake and genuine news. Guiding Questions in Appendix D lists the questions we used to guide this discussion. To reduce fatigue, we ensured that the task duration was manageable. The mean session duration was 25 minutes, including 5 minutes for the post-task interview. To avoid learning effects between the models, we varied the order in which participants interacted with the LLMs using a balanced design.The distribution of the five LLMs tools among the six participants was designed to ensure a balanced evaluation. Each participants was presented with two tools in a round-robin fashion, resulting in every tool being evaluated by exactly two participants. This method guarantees an even and fair assessment for all tools. Additionally, providing comprehensive insights into their usability and proficiency in detecting fake news

3.3.1. Data Extraction

We recorded participants' interactions through screen video recording of the sessions, and answering post-task questionnaire. Subsequently, the recorded sessions were reviewed to identify and extract usability issues encountered by the participants. we identified 24 usability issues. Following this, we classified these issues into distinct usability themes categories that have substhemes as well.

4. Experimental Setup

Before conducting the heuristic walkthrough and user study, it was crucial to prepare the LLM environments for detecting fake news content. This section will discuss the analyzed dataset and data preparation process, including data split. These steps ensure consistent and reliable results across all tested models.

4.1. The Analyzed Dataset (The Truth Seeker Dataset)

Social media platforms generate a vast amount of data daily. Using tools such as bots and content farms can significantly contribute to the widespread of misleading and false information. This can lead to profound consequences such as losing public trust in institutions and causing actual harm such influencing elections and causing a public health crisis. Human fact-checking social media content becomes impossible and impractical due to the amount of generated data. Moreover, on social media, people tend to get influenced by information based on posts and not necessarily reading the original source. Therefore, automation detection using LLMs is needed to quickly identify fake content and fight against misinformation [14]. To test the usability of LLM models in detecting fake news content, The Truth Seeker dataset was used for training and evaluating the LLMs. The dataset provides a comprehensive collection of "ground-truth" tweet datasets based on fake and real news. The dataset was designed to facilitate the detection on fake news content on the X platform (formally, Twitter). Moreover, it provides a major enhancement for models handing short-length text in addressing real-world classification challenge, like automatically detecting fake content on social media platforms [14].

The dataset contains more than 180,000 labeled tweets related to fake and real news from the PolitiFact dataset that have ground-truth values. Then, the tweets were crawled using keywords related to the news topics. The dataset includes 50 features that are categorized as the following:

- **Text and lexical features:** Analyzing the tweet itself and how it is written, including features such as word count, punctuation, part-of-speech tags, named entities, and capitalization.
- **Metadata features:** Analyzing the user who posted the tweet and the context surrounding it, such as follower count, friends count, bot score, credibility score, and influence score.

Additionally, the dataset contains features that analyze the linguistic and syntactic structure of the tweets. These features include the number of unique words, present and past tense verbs, adjectives, pronouns, determiners, conjunctions, and various punctuation marks. SpaCy tag percentages, such as ORG, NORP, GPE, PERSON, MONEY, DATA, CARDINAL, PERCENT, ORDINAL, FAC, LAW, PRODUCT, EVENT, TIME, LOC, WORKOF_ART, QUANTITY, and LANGUAGE, provide detailed insights into the types of entities and information mentioned in the tweets. The credibility and influence scores help assess the reliability of the users posting the tweets.

The LLMs will be trained on the "Majority_target" label which represents the truth value of the tweet to identify fake news. The label provided the most reliable indicator of the tweet's stance on the news statement's truthfulness. Training the LLMs on this label will prepare them to analyze how news is discussed and perceived on social media platforms regardless of whether the original source (statement) is fake or real.

4.1.1. Data Preparation

For this experiment, we used the (Features_For_Traditional_ML_Techniques) dataset file. The file contains more than 50+ features utilized to be used with classical machine learning techniques which take features as inputs rather than generating features from data like when using deep learning models. The data preparation process includes several key steps to ensure the dataset is ready for analysis:

- **Data Cleaning**: Ensuring the dataset is free from duplicates and missing values to maintain data integrity.
- **Data Balancing**: Ensuring an equal representation of real and fake news to prevent model bias.
- **Feature Engineering**: Extracting relevant features from the tweets and user metadata to enhance the model's ability to detect fake news.

All LLMs models will be trained on the same set of data for training, validation and testing sets.

4.1.2. Data Split

The dataset is splitted into **three** main categories:

- **Training set:** to train the model with labeled tweets.
- **Validation set:** to help fine-tune the model and and optimize its performance.
- **Testing set 1:** to evaluate the final model performance on unseen data.
- **Testing set 2:** for the user study to assess the usability and effectiveness of the LLMs in a real-world scenario.

The training set is used to feed the model with tweets and their corresponding truth label to identify patterns associated with real and fake news content. The validation set will help optimize the performance of fake news detection task and the testing set will evaluate the model's performance on an unseen data, which will provide unbiased measures on analyzing new data and detect fake new in real-life scenarios. To ensure data balancing, all sets will include equal number of real and fake news. This will ensure that the LLMs learn effectively from both categories. Since this paper also covers the user study. A forth category will be allocated as the testing set for participants to evaluate the usability of the LLMs. The dataset will be split the dataset into training (60%), validation (20%), and two testing sets (10% each).

5. Results

After carefully executing all the phases in our heuristic walkthrough for this evaluation, we identified several pressing usability issues with these LLMs. Similar usability issues were also discovered during the implementation of our user study. We identified 54 issues in total: our heuristic walkthrough identified 31 issues, and our user study revealed 24 issues. Please note that issues discovered in both categories were considered the same issue, and only novel issues were added as part of the user study. This means that the total count of 54 issues reflects unique usability concerns without duplicating those identified in both phases. Moreover, these issues present improvement opportunities for the LLM tools and have all been collated and presented in the supplementary materials provided [15]. We also provided a detailed overview where we categorized the issues under each theme and subtheme, highlighting the unique challenges associated with each LLM tool [16].

This section discusses usability issues the evaluators encountered during the evaluation of the LLMs. These usability issues have then been grouped into specific general themes for general descriptions and then been classified further into subthemes, which give more insight into said usability issues. Next to each theme title, we report the number of usability issues identified related to the theme. To enhance the results outcomes, next to each sub-theme title are the LLM tools in {braces} that the issues of the subtheme apply to. It is in earnest hope that the explanation of these themes and subthemes within this section will help paint a clearer picture of the problems the authors faced when training and testing the LLMs to detect fake news. The content of Table **??** provides an overview of the themes and subthemes. We also found it worth noting that the term "evaluators" in this paper encompassed both authors and participants of our user study.

To provide a quantitative measure of the performance of each LLM tool, we rated them based on four key criteria: Accuracy, Stability, Tolerance, and Efficiency. The table 2 presents these ratings. A more detailed breakdown of each criterion can be found in Appendix G. The ratings are based on the following criteria:

- **Accuracy:** Measures the correctness of the responses.
- **Stability:** Assesses how the performance varies with increased data input.
- **Tolerance:** Evaluates the flexibility in accepting different types of data input.
- **Efficiency:** Determines the speed and reliability in handling tasks without significant slowdowns or errors.

Moreover, To further analyze the identified issues, we used our categorized issues list [15] to show the the heuristic category in relation of each LLM tool. This categorization helps in understanding the distribution of issues across different usability aspect for each LLM. The chart in Figure 1 illustrates the count of heuristic categories for each LLM tool.

Theme	Subtheme
Practical Processing Limits (7)	**Input Segmentation (4)** **Timeout and Error Handling (3)**
Prompt LLM Disconnect (5)	**Misinterpretation of Inputs (5)**
Hidden Information (5)	**Insufficient Documentation (5)**
Performance under Load (6)	**Degraded Performance (6)**
Response Inaccuracy (5)	**Incorrect Outputs (5)**
User LLM Input Options (2)	**Limited Input Flexibility (2)**
Core Interaction Challenges (17)	**Save and Share Capabilities (3)** **Analysis and Input Limits (5)** **Error Handling (4)** **Contextual Understanding (5)**

TABLE 1: Overview of Themes and Subthemes

Tool	Accuracy	Stability	Tolerance	Efficiency
ChatGPT	●●○○○	●●●○○	●●●●●	●●●●○
Cohere	●●●○○	●●●●○	●●●●●	●●●●○
Copilot	●○○○○	●●●○○	●●●●○	●●●○○
Gemini	●●○○○	●●●○○	●●●○○	●○○○○
Meta AI	●●○○○	●●●●●	●○○○○	●●●●●

TABLE 2: Evaluation of LLM Tools

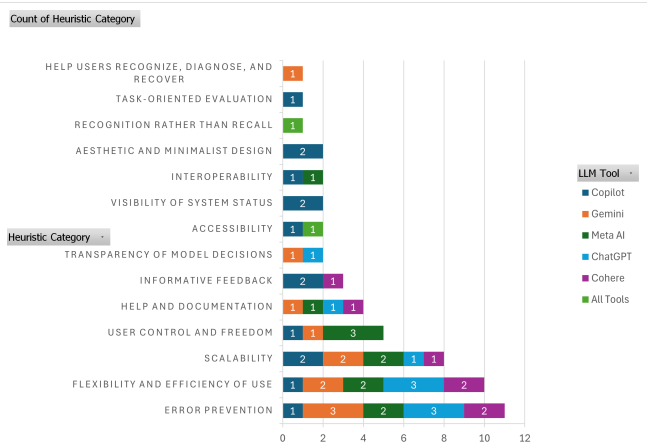

Figure 1: Count of Heuristic Categories Per LLM Tool

5.1. Practical Processing Limits (7)

balance **Input Segmentation (4)** {Meta AI, Gemini, Copilot, Cohere}: One significant issue encountered was the LLMs' inability to simultaneously process substantial amounts of user input. For instance, Cohere and Meta AI faced similar challenges with significant batch inputs. Both needed to be prompted again to complete their analysis. Copilot frequently timed out when asked to process large documents in a single instance. Additionally, it struggled to handle multiple parts of the same document effectively. When processing two parts of the same document, it provided incomplete summaries indicating difficulties in segmenting and processing the input data. Gemini also had a challenge segmented data due to missing or improperly formatted segments.

Timeout and Error Handling (3) {Meta AI, Gemini, Copilot}: Meta AI also struggled with large inputs in a single session, leading to timeouts. This required users to divide their queries into smaller portions to prevent errors or system failures. This fragmentation undermined the seamless interaction with the LLMs and made the process cumbersome. In Gemini, several instances of errors and warnings such as ConvergenceWarning and ValueError indicate poor handling of model convergence and data consistency issues. This required user intervention to correct the data or modify the model's features. Copilot frequently timed out with large documents.

User Study: Participants experienced significant limitations with all tools except ChatGPT, which could process large datasets without requiring segmentation.Participants **(P01)** and **(P05)** reported that the requirement to manage input processes manually in Cohere and Meta AI disrupted their workflow and introduced inefficiencies, with **(P05)** stating Cohere "repeatedly asking for additional prompts to process the batch". **(P02)** and **(P03)** noted that Copilot frequently timed out when processing large documents, requiring additional segmentation. **(P04)** and **(P06)** highlighted the challenges with Gemini, where improperly formatted segments required extra effort to ensure correct data formatting. ChatGPT stood out by processing large datasets without requiring segmentation, significantly improving workflow efficiency for **(P01)** and **(P06)**.

Discussion: To address these processing limitations, enhancing LLM architectures to support larger input sizes and more efficient memory management is crucial. Automatic input segmentation could be integrated into the LLMs, allowing users to input large datasets without manual adjustments and preserving context throughout the analysis. This would streamline workflows and reduce the burden on users, aligning with feedback from participants like **(P01)** and **(P05)**, who emphasized the need for seamless and efficient processing capabilities. Additionally, improving error handling

and providing clearer feedback to users about potential input issues could help mitigate the frustrations experienced with Meta AI, Gemini, and Copilot, leading to more reliable and user-friendly models.

5.2. Prompt, LLM Disconnect (5)

Misinterpretation of Inputs (5) {All}: A recurring problem was the LLMs incorrectly interpreting user input. This disconnect between the intended query and the LLM's interpretation resulted in irrelevant or incorrect responses. For instance, when using ChatGPT to analyze a nuanced query about the credibility of a tweet, the model often misinterpreted the input, providing responses that did not align with the query's intent as shown in Figure 2. Users had to rephrase their prompts and make multiple iterations to achieve the desired outcome, which was time-consuming and frustrating. Cohere and Copilot exhibited similar issues where complex sentence structures or ambiguous phrasing led to misinterpretations, necessitating repeated refinements and clarifications. Copilot misinterpreted the user's instructions multiple times. Despite the user repeatedly asking to ignore unknown data columns and proceed with predictions, the LLM failed to follow these instructions properly. Meta AI also struggled with accurately interpreting nuanced queries, often requiring users to rephrase their inputs multiple times to obtain relevant responses because some responses lacked clarity, leading to a disconnect between the user's expectations and the provided answer. Google Gemini faced similar challenges where minor input ambiguities led to significant output deviations, making it difficult to rely on the model for precise analysis. For example, Gemini often produced errors like "Analysis unsuccessful" due to sensitive information or data conflicts.

User Study: All six participants experienced issues where the LLM could not understand their prompts. Participants found themselves reiterating and modifying their prompts to make them understandable by the LLMs. **(P02)** and **(P03)** particularly noted that Copilot frequently failed to follow explicit instructions, leading to repeated clarifications. **(P04)** and reported that "Meta AI often required several iterations of rephrasing to yield relevant responses". **(P04)** and **(P06)** faced similar challenges with Gemini, where even minor ambiguities in input led to substantial output deviations.

Discussion: To address the disconnect between user prompts and LLM interpretations, improvements in natural language understanding and context awareness are crucial. Enhancing the models to better recognize and interpret complex sentence structures and ambiguous phrases can reduce the need for users to repeatedly rephrase their inputs. Additionally, implementing more robust feedback mechanisms could help users understand why their inputs are misinterpreted and provide guidance on how to phrase queries more effectively. These improvements would align with participant feedback, particularly from **(P01)** and **(P04)**, who emphasized the need for more intuitive and responsive interactions.

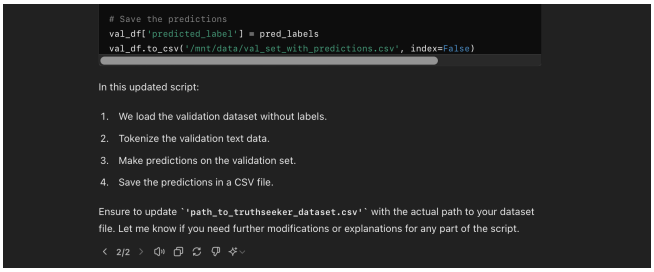

Figure 2: ChatGPT misinterpreting a request

5.3. Hidden Information (5)

Insufficient Documentation (5) {Meta AI, Gemini, ChatGPT, Copilot, Cohere}: Comprehensive and easily accessible documentation is crucial for effectively utilizing LLMs. However, it was observed that the documentation for these models often lacked clarity and detail regarding their capabilities and limitations. For instance, Meta AI's documentation was found insufficient in guiding users in optimizing their queries, leading to inefficient model use. Users frequently encountered trial-and-error scenarios, which could have been avoided with more detailed and user-friendly documentation. Similarly, Google Gemini's documentation lacked specificity, making it challenging for users to understand the model's full potential. ChatGPT's more comprehensive documentation still had gaps in explaining the nuances of the model's behavior under different conditions. Copilot also faced similar issues where users found it challenging to locate specific information or guidelines on leveraging the models effectively.

User Study: : Most participants reported that Cohere documentation was the easiest to navigate, with clear sections and descriptions of their content. **(P01)** and **(P03)** found Cohere's documentation more accessible compared to others. Participants noted that ChatGPT provided the best explanations for how specific steps were handled, as highlighted by **(P0)**, **(P02)** and **(P06)**. However, **(P04)** mentioned that in Gemini, they were "unaware of the steps taken by the LLM to reach the result", indicating a lack of transparency. **(P05)** noted that Meta AI also lacked documentation on response handling. In Copilot, users were not given clear clarifications about errors encountered, which left participants without a clear path for resolution. Some participants, including **(P01)** and **(P05)**, reported that Cohere lacked clarity on required data formats and interaction steps.

Discussion: Improving documentation for LLMs is essential to help users effectively utilize these tools. Clear, detailed, and user-friendly documentation can significantly enhance user experience by reducing trial-and-error scenarios and providing guidance on optimizing queries and interactions. Integrating examples, FAQs, and troubleshooting sections can help users better understand model behavior and capabilities. Additionally, real-time assistance features, such as in-tool tips or guided tutorials, could further support users in navigating complex model features, addressing concerns

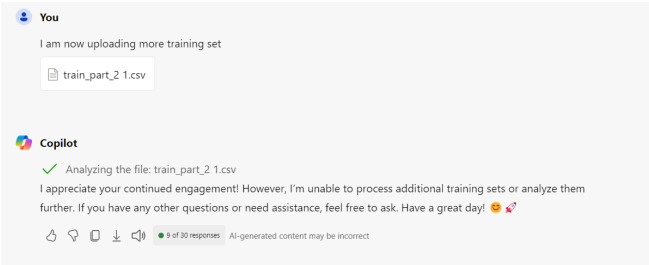

Figure 3: Analysis Error in Co-pilot

raised by participants like **(P04)** and **(P06)** about the lack of clarity and guidance in existing documentation.

5.4. Performance under Load (6)

Degraded Performance (6) {ChatGPT, Copilot, Cohere, Meta AI, Gemini}: The performance of LLM interfaces tended to degrade under varying user loads, especially during intensive phases like training, validation, and testing. Users reported slower response times and occasional timeouts when multiple processes were run concurrently. For instance, during the training phase with ChatGPT, the interface became sluggish and less responsive as the user load increased, affecting the overall efficiency of the process. Copilot also showed inconsistencies in performance, where the models would occasionally time out or slow down significantly during peak usage times. This unreliability undermined confidence in the model's ability to handle high-demand scenarios, making it less suitable for real-time applications 3. Cohere exhibited similar performance issues under load, causing interruptions and delays in the workflow. Meta AI also faced performance issues where increased user load led to slower response times and reduced efficiency. Google Gemini experienced similar problems where varying user loads during training and validation phases affected the model's responsiveness and reliability.

*User Study:*Participants discovered that the LLMs that accepted file uploads, such as ChatGPT and Cohere, took longer than usual to process and analyze the uploaded datasets. **(P01)** and **(P03)** noted that this delay impacted their "workflow efficiency". Meta AI, Copilot, and Gemini also faced similar issues, although they were not as slow as the LLMs that did accept .csv file types. **(P02)** and **(P05)** found that the increased processing time during high-demand scenarios caused frustration and hindered their ability to complete tasks promptly.

Discussion: To address performance issues under load, it is essential to optimize LLM infrastructure to better handle increased user demand. Enhancements in server capacity and load balancing can mitigate slowdowns and timeouts. Implementing more efficient data processing algorithms and resource allocation strategies can improve response times during intensive phases. Additionally, providing users with clear indicators of system load and estimated processing times can help manage expectations and improve user experience, addressing concerns raised by participants like **(P01)** and **(P04)** about performance reliability and efficiency.

5.5. Response Inaccuracy (5)

Incorrect Outputs (5) {Gemini, Meta AI, ChatGPT, Cohere, Copilot}: Inaccurate responses were a significant concern when using LLMs. Even minor misinterpretations of user input could lead to irrelevant or incorrect outputs. This issue was particularly problematic in tasks requiring high precision, such as fake news detection. Among all LLMs, we found that the Copilot kept producing incorrect outputs and failed to successfully execute the prediction task. When using Google Gemini to assess the truthfulness of a tweet, slight ambiguities in the input often resulted in inaccurate assessments, necessitating multiple iterations to obtain the correct results. Meta AI also faced similar issues where users had to refine their queries multiple times to avoid undesired and unexpected outputs. ChatGPT, while more accurate, still struggled with complex queries that required nuanced understanding. Users had to carefully craft their inputs and provide extensive context to minimize the risk of misinterpretation. Cohere and Copilot also exhibited response inaccuracies where the models provided outputs that did not align with the user's expectations, leading to repeated refinements and adjustments.

User Study: Participants initially noticed that the LLMs often did not fully comprehend their prompts, leading to unexpected results. **(P02)** and **(P03)** mentioned that ChatGPT, for example, would respond to their prompts with Python code, thinking the participant wanted to build their own machine-learning model. This misalignment in understanding between user intent and model output required participants to rephrase their prompts repeatedly, as noted by **(P04)** and **(P06)**, resulting in a "time-consuming process" to achieve the desired outcomes.

Discussion: Improving response accuracy is crucial for enhancing user trust and efficiency when using LLMs. Solutions could include incorporating more sophisticated natural language understanding techniques to better interpret complex queries and reduce ambiguities. Providing users with real-time feedback or suggestions during input formulation could help clarify intentions and guide models toward more accurate responses. Implementing these enhancements would address participant concerns, like those raised by **(P02)** and **(P04)**, about the need for consistent and reliable outputs across various tasks.

5.6. User LLM Input Options (2)

Limited Input Flexibility (2) {Copilot, Meta AI}: The ability to upload files directly to the LLMs would significantly enhance their usability. Currently, Copilot and Meta AI users have to manually input data, which can be restrictive and inefficient, especially for large datasets. For example, when working with the Truth-seeker dataset, the lack of a straightforward file upload feature meant that users

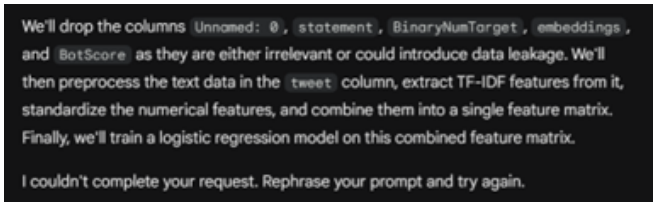

Figure 4: Numeric Input Requirement in Gemini

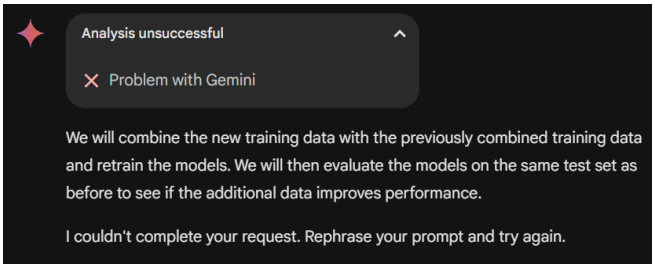

Figure 5: Unexpected Error in Gemini

had to pre-process the data and integrate it into the model manually. This added unnecessary complexity to the workflow and increased the risk of errors during data integration.

User Study: Participants noted that ChatGPT (Pro), Cohere, and Gemini (Plus) were the only LLMs that offered the option to upload datasets as .csv files directly. **(P01)** and **(P05)** found this feature particularly useful as it streamlined the process and reduced manual data entry efforts. Conversely, Meta AI did not allow any file uploads, and Copilot only accepted image uploads, which greatly affected usability for participants like **(P02)** and **(P03)**, who mentioned that the lack of comprehensive file upload capabilities limited their efficiency when handling large datasets. **Discussion:**Enhancing input flexibility by allowing direct file uploads could streamline the workflow for users and reduce the risk of errors during data integration. By supporting various file types, particularly for large datasets, LLMs can better accommodate diverse user needs and improve efficiency. Implementing these features would address participant concerns, like those raised by **(P02)** and **(P05)**, about the challenges of manual data entry and integration. Furthermore, providing detailed documentation on supported file types and formats, as well as offering guidance on best practices for data preparation, could further enhance user experience and model usability.

5.7. Core Interaction Challenges (17)

Save and Share Capabilities (3) {Meta AI, Copilot, Cohere}: Meta AI and Copilot lacked sharing the full LLM conversation publicly. They only allowed for copying responses individually, which reduces collaborative potential. Copilot does not allow users to save previous conversations or chats, limiting the ability to reference past interactions. In Cohere, the user had to keep track of predictions and results manually without easy saving or sharing mechanisms integrated into the interaction.

Analysis and Input Limits (5) {Cohere, Copilot, Gemini, Meta AI, ChatGPT}: Cohere was only able to process predictions for the first 500 tweets and required multiple steps and iterations, suggesting interaction limits when dealing with larger datasets. Meta AI imposes a word limit of 400 to 500 words and a character limit of 2048 per query, which restricts the scope of inputs. According to the process of heuristic walkthrough in Gemini, we found out that Gemini is not good at analyzing non-numerical data. For example, When we input a large dataset with too much non-numerical data, the analysis was unsuccessful.The analysis ceased because of the transformation from non-numerical data to numerical data by TF-IDF vectorization with a large dataset upload 4. We assume the reason is that the TF-IDF mechanism inside the Gemini is not complete enough to handle this large dataset. After the conversion of the files such as changing TRUE and FALSE values to 1 and 0, Gemini to process data. The error "Analysis unsuccessful" in Gemini can easily occur if the data input it is not satisfied or there are some conflicts or mismatching with the data it analyzed before. Politics and elections are sensitive information for Gemini. Analysis would interrupt if it was detected. Sometimes Gemini failed at analysis for no reason as well 5.

Error Handling (4) {Gemini, Meta AI, Copilot, Cohere}: The high occurrence rate of "Analysis unsuccessful" messages in Gemini and the inconsistent handling of sensitive information led to interrupted analysis. Meta AI also did not communicate errors effectively, leading to confusion and frustration for the user. Similarly, Copilot's error-handling capabilities were insufficient and lacked meaningful corrections for encountered issues. In Cohere, error responses were repetitive. For example, there were repeated errors around file upload errors without clear guidance on how to solve the problem.

Contextual Understanding (5) {All}: All tools lack contextual understanding from previous inputs, forcing users to repeatedly provide the same context, which is inefficient and frustrating.

User Study: Participants **(P02)** to **(P05)** to found that the lack of save and share capabilities in Meta AI, and Copilot significantly hindered collaborative potential and workflow efficiency. **(P03)** and **(P05)** noted that the analysis limits in Cohere and Meta AI led to cumbersome workflows, as they had to break down input data manually. **(P04)** and **(P06)** experienced frequent analysis interruptions in Gemini due to data conflicts and sensitive information, which disrupted their tasks. All participants highlighted the need for better contextual understanding across all tools, as they had to repeat context in their inputs, leading to frustration and inefficiency.

Discussion: Enhancing LLM capabilities in saving and sharing interactions would facilitate greater collaboration and ease of use. Addressing analysis and input limits by allowing larger datasets and non-numerical data processing could improve efficiency. Improving error handling with clearer, more actionable feedback and enhancing contextual

understanding across sessions would significantly enhance user experience and satisfaction. Implementing these improvements, as suggested by participant feedback, would help LLMs better meet user needs and expectations.

6. Recommendations and Improvements

Based on the comprehensive heuristic evaluations and the user study, we identified several key areas for potential enhancements to improve the usability of LLMs in detecting and analyzing fake news . The following recommendations are made to address issues and optimize the overall user experience.

- **Enhance Input Processing Capabilities:** LLMs should improve their ability in processing large inputs without requiring segmentation. this will reduce time for user to manually divide data, thus minimizing errors and increasing efficiency . Also, there should be better LLM error handling and timeout mechanisms to manage large documents. LLMs should provide clear feedback on processing status and expected completion times. LLMs should be able to process various data types to streamline the user workflow.
- **Improve Prompt Interpretation:** LLMs need to enhance their capitals in their ability to accurately response to complex queries. this can be done by improving LLMs understanding capabilities. LLMs should provide examples of crafting effecting prompts, when it can not produce required output to guide users about the process. this can reduce repetitive prompts of the same task.
- **Clarify Documentation:** All LLMs should have detailed and user-friendly documentations that explain their functionalities and limitations. this will help user understand how to utilize the tools and what are the expected outcomes. Additionally, there should be transparency in model s to ensure user's trust in the provided outputs.
- **Optimize Performance Under Load:** the scalability of LLMs needs to be enhanced to handle user load without leading to performance degradation. Additionally, resource management should be considered to maintain responsiveness and efficiency during peak usage times.
- **Increase Response Accuracy:** There should be continuous updates for algorithms to improve accuracy of outputs. Users should be able to report inaccuracies easily through a clear feedback method.
- **Improve User Interaction Design:** LLMs should have a simple design to improve user's experience.
- **System feedback:** LLMs should provide real-time updates about ongoing processing prompts to help user identify current state and output completion of the LLM. Also, errors should have clarity and suggestions to recover from the encountered errors.

7. Limitations

There are several constraints that need to be addressed when conducting the LLMs usability issues evaluations. First, we encountered many issues during the experimental setup, each LLM has a unique method in handing the same dataset for training, validating, and testing phases. This variation poses challenges when working on standardizing the setup which will lead to inconsistencies in the assessment process. for example, due to some LLMs limitations, not all LLMs accepted file uploads, and some had to be entered manually which cluttered the data and impacted the representation of dataset features to the LLM. Second, there are limited resources on algorithms and learning models used in these LLMs. Moreover, the dynamic nature of these LLM tools where they are frequently updated with new features and improvements, makes some identified usability issues obsolete or introduces new ones. Third, there were some constraints regarding limited user representation. All participants come from similar backgrounds thus making the evaluation not presenting a diverse user base. Finally, there are other factors that contribute to the performance of the LLM such as hardware and software environments which can affect the user experience.

8. Conclusion

This paper focuses on the usability evaluation of five AI tools (ChatGPT, Cohere, Gemini, Copilot, and Meta AI) used for fake news detection. These tools were assessed through both user studies and heuristic walkthroughs. Our evaluation identified numerous usability issues, providing a comprehensive analysis and discussion of the challenges associated with each tool. The heuristic walkthrough methods employed in this study are also applicable to other AI tools, offering a framework for detecting and analyzing usability issues in the context of fake news detection. Our analysis offers valuable insights into the usability strengths and weaknesses of these AI tools, presenting a broad perspective on LLM tool evaluation. We hope that our work will be beneficial to researchers and users alike, aiding in the development and utilization of AI tools. By addressing the identified usability issues, these tools can become more effective and efficient, enhancing their usability and reliability in practical applications.

References

[1] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, and J. L. Zittrain, "The science of fake news," *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[2] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 109:1–109:40, 2020.

[3] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 15, no. 3, pp. 39:1–39:45, 2024.

[4] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2024. [Online]. Available: http://arxiv.org/abs/2307.06435

[5] Backlinko, "Chatgpt statistics 2024: How many people use chatgpt?" June 4 2024. [Online]. Available: https://backlinko.com/chatgpt-stats

[6] OpenAI, "Custom instructions for chatgpt — openai help center," retrieved 3 July 2024. [Online]. Available: https://help.openai.com/en/articles/8096356-custom-instructions-for-chatgpt

[7] n. nguyen, "Google gemini statistics: Key insights and trends [2024]," May 28 2024. [Online]. Available: https://doit.software/blog/google-gemini-statistics

[8] A. Beatman, "Azure openai service powers the microsoft copilot ecosystem," December 14 2023, microsoft Azure Blog. [Online]. Available: https://azure.microsoft.com/en-us/blog/azure-openai-service-powers-the-microsoft-copilot-ecosystem/

[9] Cohere, "About," retrieved 3 July 2024. [Online]. Available: https://cohere.com/about

[10] The Social Shepherd, "21 essential meta statistics you need to know in 2024," retrieved 3 July 2024. [Online]. Available: https://thesocialshepherd.com/blog/meta-statistics

[11] J. Smith, L. N. Q. Do, and E. Murphy-Hill, "Why can't johnny fix vulnerabilities: a usability evaluation of static analysis tools for security," in *Proceedings of the Sixteenth USENIX Conference on Usable Privacy and Security*, ser. SOUPS'20. USA: USENIX Association, 2020.

[12] O. Leßenich and S. Sobernig, "Usefulness and usability of heuristic walkthroughs for evaluating domain-specific developer tools in industry: Evidence from four field simulations," *Inf. Softw. Technol.*, vol. 160, no. C, aug 2023. [Online]. Available: https://doi.org/10.1016/j.infsof.2023.107220

[13] A. Sears, "Heuristic walkthroughs: Finding the problems without the noise," *International Journal of Human–Computer Interaction*, vol. 9, no. 3, pp. 213–234, 1997. [Online]. Available: https://doi.org/10.1207/s15327590ijhc0903_2

[14] S. Dadkhah, X. Zhang, A. G. Weismann, A. Firouzi, and A. A. Ghorbani, "The largest social media ground-truth dataset for real/fake content: Truthseeker," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 3, pp. 3376–3390, 2024.

[15] [All], "Heuristic walkthrough result list of issues," URL: https://uoguelphca-my.sharepoint.com/:x:/g/personal/azzouzs_uoguelph_ca/EV6xf7EgqItJki4Nd6Buo44BuIbIX1J5-e-vc2Wh2WrhJw?e=eIlK0v.

[16] ——, "Categorized usability issues across themes and subthemes," URL: https://uoguelphca-my.sharepoint.com/:w:/g/personal/azzouzs_uoguelph_ca/EXaBC4TZ6B9IjBm0W5u9SuoBogx_h7fwlHA-nFmSi3EbcA?e=toUkyM.

Appendix A.
User Study Briefing:

We really appreciate your involvement in this study. We are eager to improve the usability of our models in different LLMs to better in detecting fake news. Your test experience and suggestions will become valuable for our study. You will use two models in two different AI tools to test the accuracy of detecting fake news, and we will ask you some questions about your experience and recommendations. During this session, We will be recording both the computer screen and audio from our conversation. Please let me know at any point if you'd like to stop or pause the recording.You agree to these terms, could you please sign this consent form for me? Sign form: ¡obtain signature from participant¿

Appendix B.
User Study Task Briefing:

You have been asked to evaluate two models in two different tools to test their usability. These models have been trained and validated by fake news dataset.the datset has fake and real statements with related tweets along with thier features such as word counts, named entities, capitalization and more. you will ask the LLM to analyze the truthfulness of the tweets. I will let you test our models separately. You will be given testing instructions to guide your test procedure. There are some questions prepared for you to answer after your test to share your perspectives and offer some valuable suggestions. The whole process will take around 20 – 30 minutes.

list of Tasks:

- Upload the testing set (without labels) to the LLM. Observe how the LLM processes the data and outputs its analysis on the truthfulness of the tweets.
- Upload the testing set (with labels) to the LLM. Compare the LLM's analysis with the actual labels to evaluate its accuracy. Note the number of correct and incorrect predictions.
- For predictions, analyze what led the LLM to that conclusion. For each and investigate why the LLM made that error
- Compare LLM's predictions with the corresponding "Majority target" label to assess its accuracy.

Appendix C.
Post-Study Questions:

1) Which issues did you encounter when using the LLMs?
2) How user-friendly or easy-to-use were the LLMs?
3) How fast were the LLMs in detecting fake news?
4) How fast were the LLMs in detecting genuine news?
5) Were there any moments when the LLMs confused fake news with genuine news?
6) Were there any moments when the LLMs confused genuine news with fake news?

Appendix D.
Heuristic Walkthroughs Guide:

balance **Pass 1:**
You work at the security department in a company and asked to filter fake news for clients and protecting the reputation of company. Your security team plans to choose a AI tools from ChatGPT, Gemini, Cohere, Co-pilot and Meta AI to build a model for detecting fake news. You need to find the best tools among them by evaluating their performance and usability.

Steps of tasks:

- Upload the testing set (without labels) to the LLM. Observe how the LLM processes the data and outputs its analysis on the truthfulness of the tweets.
- Upload the testing set (with labels) to the LLM. Compare the LLM's analysis with the actual labels to evaluate its accuracy. Note the number of correct and incorrect predictions.
- For predictions, analyze what led the LLM to that conclusion. For each and investigate why the LLM made that error
- Compare LLM's predictions with the corresponding "Majority target" label to assess its accuracy.

Repeat these tasks until you feel satisfied with your assessments. Use the questions below to guide your evaluation. Record any usability problems you encounter during this phase.

Guiding Questions:

1) How easy was it to understand the dataset structure and content? Were the instructions for uploading the testing set clear?
2) How intuitive was the process of uploading the testing set and receiving the analysis? Did you encounter any difficulties while interacting with the LLM? Some tools do not have the file upload options and have small limits in prompt input.
3) How accurate was the LLM in detecting the truthfulness of the tweets? For correct predictions, what patterns or features did the LLM seem to rely on? For incorrect predictions, what were the common errors, and how could they be mitigated?
4) What improvements would you suggest for the LLM's performance? How could the process be made more user-friendly?
5) How satisfied are you with the LLM's usability in detecting fake news content? What aspects did you find most beneficial, and which aspects need improvement?

Pass 2: Heuristics
You can evaluate the AI tools by the heuristics below after you went through Pass 1. The definitions of every heuristic are shown below:

- **Visibility of System Status**
  - The system should keep users informed about ongoing processes, errors, and results in a clear and timely manner.
- **Match Between System and the Real World**
  - Use terminology and concepts familiar to users rather than technical jargon, ensuring ease of understanding and communication.
- **User Control and Freedom**
  - Provide users with clear options to undo actions, exit unwanted states, and navigate through the system without difficulty.
- **Consistency and Standards**
  - Ensure uniformity in interface elements, interactions, and terminology across different sections of the LLM tool.
- **Error Prevention**
  - Design the system to anticipate and prevent errors through validation mechanisms and clear guidelines for input and processes.
- **Recognition Rather Than Recall**
  - Minimize the user's memory load by making options, actions, and settings visible and accessible throughout the interaction.
- **Flexibility and Efficiency of Use**
  - Cater to both novice and expert users by offering shortcuts, customizable settings, and efficient interaction methods.
- **Aesthetic and Minimalist Design**
  - Keep the user interface uncluttered and focused on essential information, avoiding unnecessary distractions or complexity.
- **Help Users Recognize, Diagnose, and Recover**
  - Provide clear, plain-language error messages that help users understand issues, suggest solutions, and recover from errors effectively.
- **Help and Documentation**
  - Offer comprehensive and easily searchable help resources that provide guidance on system usage, troubleshooting, and task completion.
- **Task-Oriented Evaluation**
  - Ensure the system supports users in performing tasks such as data loading, preprocessing, model training, and result evaluation effectively.
- **Informative Feedback**
  - Provide timely and detailed feedback on user actions and system state, particularly after critical processes like model training.
- **Transparency of Model Decisions**
  - Explain the rationale behind system decisions and predictions to build user trust and understanding, particularly in fake news detection.
- **Data Privacy and Security**
  - Ensure users' data is handled securely, provide clear

information on privacy practices, and offer options for managing data privacy settings.

- **Scalability**
  - Design the system to efficiently handle large datasets and varying user demands without significant performance degradation.
- **Interoperability**
  - Facilitate seamless integration with other tools and systems commonly used by users for data analysis and visualization.
- **Accessibility**
  - Ensure the system is accessible to users with different abilities by supporting features like screen readers and keyboard navigation.

Appendix E.
Summary of Participants Experience

Participant	LLM	Prior Usage Experience	Core Functionality	Knowledge of Advanced Features
P1	ChatGPT	●●●●●	●●●●●	●●●●●
P2	ChatGPT	●●●●●	●●●●●	●●●●○
P3	Copilot	●●●○○	●●●●○	●●●○○
P4	Meta AI	●●○○○	●●●○○	●●○○○
P5	Cohere	●●○○○	●●●●○	●●●○○
P6	Gemini	●●●●●	●●●●●	●●●●○
Participant	**LLM**	**Prior Usage Experience**	**Core Functionality**	**Knowledge of Advanced Features**
P1	Cohere	●●●●○	●●●●○	●●●○○
P2	Copilot	●●○○○	●●●○○	●●●○○
P3	Cohere	●●○○○	●●●○○	●●○○○
P4	Gemini	●●●●●	●●●●○	●●○○○
P5	Meta AI	●●●○○	●●○○○	●●○○○
P6	ChatGPT	●●●●●	●●●●●	●●●●●

TABLE 3: Responses of Participants

Appendix F.
Heuristics Summarized

Heuristic	Description
Visibility of System Status	The system should keep users informed about ongoing processes, errors, and results in a clear and timely manner.
Match Between System and the Real World	Use terminology and concepts familiar to users rather than technical jargon, ensuring ease of understanding and communication.
User Control and Freedom	Provide users with clear options to undo actions, exit unwanted states, and navigate through the system without difficulty.
Consistency and Standards	Ensure uniformity in interface elements, interactions, and terminology across different sections of the LLM tool.
Error Prevention	Design the system to anticipate and prevent errors through validation mechanisms and clear guidelines for input and processes.
Recognition Rather Than Recall	Minimize the user's memory load by making options, actions, and settings visible and accessible throughout the interaction.
Flexibility and Efficiency of Use	Cater to both novice and expert users by offering shortcuts, customizable settings, and efficient interaction methods.
Aesthetic and Minimalist Design	Keep the user interface uncluttered and focused on essential information, avoiding unnecessary distractions or complexity.
Help Users Recognize, Diagnose, and Recover	Provide clear, plain-language error messages that help users understand issues, suggest solutions, and recover from errors effectively.
Help and Documentation	Offer comprehensive and easily searchable help resources that provide guidance on system usage, troubleshooting, and task completion.
Task-Oriented Evaluation	Ensure the system supports users in performing tasks such as data loading, preprocessing, model training, and result evaluation effectively.
Informative Feedback	Provide timely and detailed feedback on user actions and system state, particularly after critical processes like model training.
Transparency of Model Decisions	Explain the rationale behind system decisions and predictions to build user trust and understanding, particularly in fake news detection.
Data Privacy and Security	Ensure users' data is handled securely, provide clear information on privacy practices, and offer options for managing data privacy settings.
Scalability	Design the system to efficiently handle large datasets and varying user demands without significant performance degradation.
Interoperability	Facilitate seamless integration with other tools and systems commonly used by users for data analysis and visualization.
Accessibility	Ensure the system is accessible to users with different abilities by supporting features like screen readers and keyboard navigation.

TABLE 4: Heuristics for Evaluating LLM Usability

Appendix G.
Model Evaluation Criteria

All ratings are out of 5.

Accuracy

- **Poor (1)**: Average accuracy below 50%.
- **Satisfied (2)**: Average accuracy between 50% and 70%.
- **Good (3)**: Average accuracy between 70% and 80%.
- **Great (4)**: Average accuracy between 80% and 90%.
- **Outstanding (5)**: Average accuracy between 90% and 100%.

Stability

- **1 - 2**: Accuracy decreased by more than 10% as more data is input.
- **3 - 4**: Accuracy decreased as more data is input.
- **5**: Accuracy steadily increased as more data is input.

Tolerance

- **1 - 2**: Data input is very limited.
- **3 - 4**: Accepts the majority of data input.
- **5**: Accepts many types of data input.

Efficiency

- **1 - 2**: Slow, gets stuck when encountering errors.
- **3 - 4**: Good speed, can skip errors.
- **5**: Steadily fast.