FinGrAct: A Framework for FINe-GRrained Evaluation of ACTionability in Explainable Automatic Fact-Checking

Islam Eldifrawi, Shengrui Wang, Amine Trabelsi
Department of Computer Science, Université de Sherbrooke
{Islam.Eldifrawi;Shengrui.Wang;Amine.Trabelsi}@usherbrooke.ca

Abstract

The field of explainable Automatic Fact-Checking (AFC) aims to enhance the transparency and trustworthiness of automated fact-verification systems by providing clear and comprehensible explanations. However, the effectiveness of these explanations depends on their actionability—the extent to which an AFC explanation pinpoints the error, supplies the correct fact, and backs it with sources. Despite actionability being critical for high-quality explanations, no prior research has proposed a method to evaluate it. This paper introduces FinGrAct, a fine-grained evaluation framework that can access the web and is designed to assess actionability in AFC explanations through well-defined criteria. We also introduce a novel dataset to evaluate actionability in AFC explanations. FinGrAct surpasses state-of-the-art (SOTA) evaluators, achieving the highest Pearson and Kendall correlation with human judgments while demonstrating the lowest egocentric bias, making it a more robust evaluation approach for actionability evaluation in AFC.

1 Introduction

Explanation of claim veracity (see Figure 1 for examples of explanations) is essential in automated fact-checking as it enhances transparency, fosters trust, and educates users by clarifying why a claim is deemed true or not.

In the domain of explainable automated fact-checking (AFC), the quality of the explanation of a claim's predicted veracity is assessed based on the presence of specific desired properties, referred to as "desiderata," as outlined by Kotonya and Toni (2020a). While some of these desiderata have been explored in the field of summarization like coherence and fluency, others, such as actionability, a critical property in fact-checking, remain unexplored. To date, no automatic evaluator for actionability has been developed. However, some general purpose SOTA evaluators emerged, but only for summarization tasks. In this work, these evaluators are adapted to AFC.

Actionability remains underexplored due to its complexity. According to Kotonya and Toni (2020a), actionability in AFC refers to **providing factual corrections for identified errors in a non-factual claim, supported by sources and references**. This suggests that actionability is highly correlated with other key properties, such as relevance to the claim and completeness, where the explanation must provide a comprehensive context for why the claim is considered true or not. Providing an automatic evaluator for this desideratum is critical, because it yields a reproducible, fine-grained metric that (i) enables large-scale benchmarking across systems and (ii) offers a reliable reward signal for training or selecting explanation models, thereby accelerating research cycles, enhancing misinformation mitigation while safeguarding user trust. Evaluating actionability, however, presents significant challenges as it requires well-defined criteria for judgment, as well as access to external sources in the internet to evaluate the supporting references. Figure 1 shows examples of explanations with different degrees of actionability. To tackle the challenge of automatically evaluating the actionability of explanations in AFC systems, we introduce FinGrAct. FinGrAct is a fine-grained auto-evaluator that systematically measures the degree of actionability in these explanations. The evaluation framework is illustrated in Figure 1. Our contributions:

We present a dataset, constructed from existing sources, containing explanations for claim labels with varying degrees of actionability. Additionally, it includes human-rated actionability scores, which can serve as benchmarks for evaluating the performance of different evaluators of actionability.

We present FinGrAct a novel fine-grained evaluator of explanations for actionability within the context of explainable AFC, based on LLM prompting, that correlates better with human ratings than other

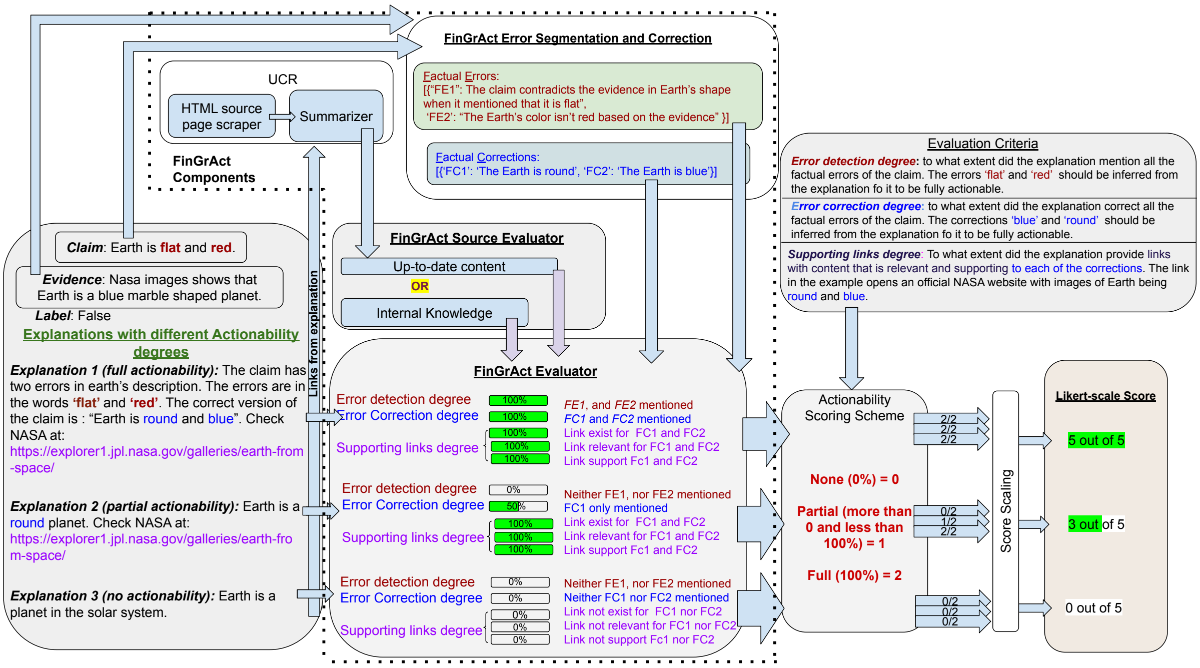
FinGrAct Error Segmentation and Correction
Factual Errors:
[{"FE1": The claim contradicts the evidence in Earth's shape when it mentioned that it is flat",
'FE2': "The Earth's color isn't red based on the evidence" }]
Factual Corrections:
[{'FC1': 'The Earth is round', 'FC2': 'The Earth is blue'}]
UCR
HTML source page scraper
Summarizer
FinGrAct Components
Claim: Earth is flat and red.
Evidence: Nasa images shows that Earth is a blue marble shaped planet.
Label: False
Explanations with different Actionability degrees
Explanation 1 (full actionability): The claim has two errors in earth's description. The errors are in the words 'flat' and 'red'. The correct version of the claim is : "Earth is round and blue". Check NASA at: https://explorer1.jpl.nasa.gov/galleries/earth-from-space/
Explanation 2 (partial actionability): Earth is a round planet. Check NASA at: https://explorer1.jpl.nasa.gov/galleries/earth-from-space/
Explanation 3 (no actionability): Earth is a planet in the solar system.
Links from explanation
FinGrAct Source Evaluator
Up-to-date content
OR
Internal Knowledge
FinGrAct Evaluator
Error detection degree
Error Correction degree
Supporting links degree
100%
100%
100%
100%
100%
FE1, and FE2 mentioned
FC1 and FC2 mentioned
Link exist for FC1 and FC2
Link relevant for FC1 and FC2
Link support Fc1 and FC2
Error detection degree
Error Correction degree
Supporting links degree
0%
50%
100%
100%
100%
Neither FE1, nor FE2 mentioned
FC1 only mentioned
Link exist for FC1 and FC2
Link relevant for FC1 and FC2
Link support Fc1 and FC2
Error detection degree
Error Correction degree
Supporting links degree
0%
0%
0%
0%
0%
Neither FE1, nor FE2 mentioned
Neither FC1 nor FC2 mentioned
Link not exist for FC1 nor FC2
Link not relevant for FC1 nor FC2
Link not support Fc1 nor FC2
Evaluation Criteria
Error detection degree: to what extent did the explanation mention all the factual errors of the claim. The errors 'flat' and 'red' should be inferred from the explanation fo it to be fully actionable.
Error correction degree: to what extent did the explanation correct all the factual errors of the claim. The corrections 'blue' and 'round' should be inferred from the explanation fo it to be fully actionable.
Supporting links degree: To what extent did the explanation provide links with content that is relevant and supporting to each of the corrections. The link in the example opens an official NASA website with images of Earth being round and blue.
Actionability Scoring Scheme
None (0%) = 0
Partial (more than 0 and less than 100%) = 1
Full (100%) = 2
2/2
2/2
2/2
0/2
1/2
2/2
0/2
0/2
0/2
Score Scaling
Likert-scale Score
5 out of 5
3 out of 5
0 out of 5

Figure 1: **FinGrAct Flow**. *Input*: Claim, Evidence, Label, Explanation to be evaluated (with varying degrees of actionability or none). *Processing*: FinGrAct Error Segmentation and Correction, FinGrAct Evaluator, FinGrAct Source Evaluator, URL Content Retriever (UCR). *Output*: 3 evaluation scores (Error Detection, Error Correction, Supporting Links Degrees) classified as None (0), Partial (1), Full (2), aggregated into a Likert scale score (1 to 5).

adapted SOTA evaluators. We examined SOTA evaluators like TIGER-Score, G-Eval, Promethuse and FLAME with detailed analysis in Section 2, then we choose the ones suitable for AFC in Section 5.1. These evaluators are SOTA across multiple datasets and benchmarks like SummEval (Fabbri et al., 2021), and WMT-22 (Freitag et al., 2022) surpassing InstructionScore (Xu et al., 2023), BartScore (Yuan et al., 2021), BLEURT (Sellam et al., 2020), and ROUGE (Lin, 2004). In addition, we showed that adding a simple component, named the URL Content Retriever (UCR), to fetch and evaluate web-link **textual** content, enhanced the performance of actionability evaluation in all SOTA evaluators. UCR enables LLMs to assess the relevance of the supporting sources.

We conduct an investigation into the ego-centric bias present in LLMs. Ego-centric bias happens when LLMs acting as judges, tend to assign higher scores to their own output. FinGrAct has the least ego-centric bias among all other adapted SOTA evaluators.

2 Related Work

With the emergence of LLMs such as ChatGPT, recent studies have employed these models as evaluators (Fu et al., 2024) to assess LLM performance across various benchmarks, including G-EVAL (Liu et al., 2023), and FineSurE (Song et al., 2024a). Nonetheless, it has been observed that LLMs functioning as auto-raters tend to show a preference for their own generated responses (Panickssery et al., 2024). This is called ego-centric bias. Training open-source general-purpose LLM autoraters has been investigated recently. TIGERScore, a Llama-2 model trained on GPT-4 generated error analysis data across multiple tasks, such as summarization, translation, and instruction-following, is presented by Jiang et al. 2023. However, it doesn't follow a likert-scale scoring or can it be scaled to it. For instance, it can produce a score of -12 making it hard to be compared to other evaluators.

Prometheus (Kim et al., 2023), InstructScore (Xu et al., 2023), and Prometheus-2 (Kim et al., 2024) are comparable methods. Vu et al. 2024 developed reward models used for aligning LLMs to human preferences via reinforcement learning and called their model FLAME. However, they haven't published their model or their dataset collection. SOTA evaluators like G-Eval and Prometheus are widely used in summarization to assess properties such as coherence and faithfulness. Unlike property-specific tools like FineSurE, they are adaptable for evaluating new attributes. FineSurE gives a specific fine-grained definition for each property it measures in a summary based on the key-facts it found in the transcript. Making its adaptation to new tasks and new properties very hard.

All the mentioned work is directed towards evaluating certain desired properties in diverse tasks like summarization and question answering. However, little effort has been directed towards developing an autorater for the desiderata of the explanations of AFC. For instance, Feher et al. (2025) tried to evaluate properties like (self)-contradiction, hallucination, convincingness and overall quality. Kotonya and Toni (2024) evaluated free-form properties like coherence, deductive Properties like non-redundancy, and argumentative Properties like self-support. Xing et al. (2025) introduced a new evaluation protocol – citation masking and recovery – to assess attribution quality in generated explanations. However, they didn't address some critical properties, mentioned by Kotonya and Toni (2020a) and Eldifrawi et al. (2024), and were deemed critical for fact-checking explanation like actionability for instance.

3 FinGrAct Framework

3.1 Implementation Details

Inspired by recent advancements in fine-grained evaluation criteria for summarization—particularly in assessing key properties such as faithfulness and completeness, as demonstrated in works like Ye et al. (2023); Zhang et al. (2024); Song et al. (2024a)—we propose a specialized fine-grained evaluation methodology designed to assess the actionability of AFC explanations. As depicted in Figure 1, we propose a divide-and-conquer approach to evaluating the actionability of an explanation—given a false claim and supporting evidence—by breaking it down into three distinct tasks: *Error Segmentation and Correction*, *Explanation Evaluation*, and *Source Evaluation*. Detailed prompts can be found in Appendix H.

Error Segmentation and Correction: This task is necessary for FinGrAct to preprocess a claim, extract what it identifies as *factual errors*, and determine how *corrections* within this claim should be made based on reliable given evidence (see Figure 1 for an example). Once this information is extracted, FinGrAct can evaluate the actionability of any given explanation of why a claim is false in the next task by aligning it with this information. Here, the underlying LLM is instructed to decompose the claim into atomic (sub)claims, assessing each for factual errors based on the evidence. It then explains the error (error reason) (e.g., FE1 in Figure 1) and provides a subclaim correction (e.g., FC1 in Figure 1). The output consists of triples (JSON output): false subclaim, error reason, and correction for each false subclaim. The corresponding prompt to this description is in Appendix H.

Explanation Evaluation: FinGrAct Evaluator Given lists of error explanations and generated corrections from the previous task, this step verifies whether these elements are explicitly inferred from the provided explanation. For a given explanation, the evaluation outputs a boolean value "Yes" or "No" for each error and each correction across all false subclaims. The prompt for this evaluation phase also includes instructions to assess the web sources mentioned in the explanation. Their verification is conducted in the Sources Evaluation.

Sources Evaluation: The goal here is to determine whether a link in an explanation exists, is relevant, and its content supports the needed corrections. Two methods were tested. The first relies on the LLM's prior knowledge, while the second involves retrieving the link's content using an external component (Figure 1), the *URL Content Retriever (UCR)*. For the first method (without UCR), the LLM is instructed to check, based on its knowledge, if there are relevant links in the explanation that support the needed corrections and to respond with a Yes or No.

For the second method (with UCR), the UCR external component is integrated *before prompting* the evaluator to verify and validate the sources referenced in the explanations. The 'requests' library in python is used to scrap the text of the web-page of the link in several steps: Firstly, the HTML textual content of each link is scraped. Secondly, all HTML tags are removed to extract clean, readable text. Thirdly, the extracted textual content is then summarized using the MiniLM-L6-v2 model (Susanto et al., 2024). Summarization is used to control the amount of tokens that will be inputted into the LLM and to discard any irrelevant text. Lastly, the summarized content is subsequently incorporated into the prompt provided to the auto-evaluator, which assesses the relevance of the link's content and determines whether it supports the needed corrections. More specifically, the LLM is instructed to check from the output of the UCR if the links are working (Yes or No for Link exist, Figure 1), to check if the content is relevant (Yes or No, Link relevant) and to check if it supports the corrections (Yes or No, Link support).

For a given explanation on why a claim is false, the output of this phase is a set of false subclaims S, each containing information on whether the explanation has mentioned the error, corrected the error, and includes a functional supporting link or not (for the version without UCR, only error mention, error correction and supporting link exist).

3.2 Actionability Scoring Scheme

Algorithm 1 Actionability Scoring Algorithm (Case With UCR)

```
**Require:** False subclaims $S = \{s_1, s_2, \ldots, s_n\}$
**Ensure:** Scores $(E_d, E_c, L_e, L_r, L_s)$
$E_d \leftarrow 0, E_c \leftarrow 0, L_e \leftarrow 0, L_r \leftarrow 0, L_s \leftarrow 0$
**for** $s_i \in S$ **do**
    $E_d \leftarrow E_d + \mathbf{1}(s_i \text{ has detected errors})$
    $E_c \leftarrow E_c + \mathbf{1}(s_i \text{ has corrected errors})$
    $L_e \leftarrow L_e + \mathbf{1}(s_i \text{ has a functional link})$
    $L_r \leftarrow L_r + \mathbf{1}(s_i \text{ has a relevant link})$
    $L_s \leftarrow L_s + \mathbf{1}(s_i \text{ has a supporting link})$
**end for**
▷ Categorize (i): 2 if i = 1, 1 if 0 < i < 1, else 0
$E_d \leftarrow \text{Categorize}(E_d/n)$
$E_c \leftarrow \text{Categorize}(E_c/n)$
$L_e \leftarrow \frac{\text{Categorize}(L_e/n)}{2}$
$L_r \leftarrow \frac{\text{Categorize}(L_r/n)}{4}$
$L_s \leftarrow \frac{\text{Categorize}(L_s/n)}{4}$
**return** $(E_d, E_c, L_e + L_r + L_s)$
```

Given the output described in the previous paragraph, this phase deals with returning a final evaluation score of the actionability of an explanation. As shown in Algorithm 1 and Figure 1, error detection and correction are categorized into three levels: full, partial, or none. ***Error Detection:*** A score of 2 is awarded if all factual errors in the claim are fully identified, 1 if only some errors are detected, and 0 if no errors are recognized (Categorize function in Algorithm 1). ***Error Correction:*** If all identified errors are fully corrected, the explanation receives a score of 2; if only some are addressed, it scores 1; and if no corrections are made, it scores 0. ***Supporting Links:*** Evaluation is based on the sum of the scores of three key factors: Link functionality – whether the link is accessible (score: 1). Relevance – whether the linked content pertains to the explanation's topic (score: 0.5). Support – whether the linked content directly substantiates the corrections needed to make the claim true. (score: 0.5). The maximum possible score is 6. To normalize this score for comparison with other SOTA evaluators, we apply a scaling factor of 5/6, approximating the final score to a Likert scale ranging from 0 to 5.

4 The Evaluation Dataset

4.1 Data Collection

The aim is to develop a dataset that encompasses varying levels of actionability. The dataset must include explanations with different degrees of error detection and correction, with some incorporating supporting references while others do not. Furthermore, a distinct category of actionable explanations includes counterfactual (CF) explanations. In conclusion, the dataset should include both counterfactual and non-counterfactual explanations, each demonstrating different levels of actionability.

The sources of this dataset are two benchmark datasets. The first was created by Dai et al. 2022, and it contains false claims and counterfactual explanations that explains the reason why the claims are false in three different formats. From this dataset, we were able to generate six different categories of actionable explanations as shown in Figure 2. The categories are: **Error Detection Only:** The explanation only highlights the error in the claim. **Error Correction Only:** The explanation only provides a corrected version of the non-factual claim. **Error Correction and Detection Only:** The explanation does both error detection and correction, however, it doesn't provide any sources/links that support its content. **Error Detection with Sources:** The explanation does error detection and it has sources/links that should support its content. **Error Correction with Sources:** The explanation does error correction and it has sources/links that should support its content. **Error Detection and Correction with Sources:** The explanation does error detection and correction, and it has sources that should support its content.

The second source is the data from Kotonya and Toni (2020b). The explanations are summarizations of evidence containing different degrees of actionability. We extracted false and partially true claims with their evidence and explanations. Half of each category has sources and half does not, resulting in four different categories in Figure 2.

4.2 Dataset Creation

The dataset was constructed in three stages. The first, illustrated in Figure 2, involves categorizing the counterfactual (CF) data from (Dai et al., 2022) into six distinct categories. These categories, detailed in Section 4, are based on the extent of error detection, the level of correction provided, and the presence of supporting sources. Then the second as shown in Figure 2 is where the dataset from (Kotonya and Toni, 2020b) is used to generate explanations that are divided into four different categories as mentioned in Section 3.2. The data generated from both stages are merged and then sampled to obtain 203 random instances, ensuring representation across all categories mentioned earlier in Section 3.2. As a result, we have constructed a diverse dataset that includes varying degrees of actionability, comprising both counterfactual (CF) and non-counterfactual explanations with average input length of 8.93 words and average explanation length of 33.1 words.

Subsequently, the dataset is augmented with other generated explanations from three large language models (LLMs): LLAMA-7B, Mistral-7B and GPT-4. These generative models serve as the foundation for Prometheus, G-Eval, and FinGrAct. The generated explanations are only later utilized in the ego-centric bias study to assess whether the evaluators exhibit preferential bias toward outputs generated by their respective underlying LLMs. Finally, each claim is accompanied by four explanations—one sourced from the combined dataset and three generated by the specified LLMs, following the prompts detailed in the Appendix E. Three human annotators independently assess the actionability of each explanation based on the provided evaluation guidelines, outlined in Appendix D.

In addition, the annotators underwent training through a video demonstration and followed an iterative annotation process. Initially, they were provided with a small subset of the data and encouraged to ask questions and share feedback. Based on their input, the instructions were refined to minimize confusion and clarify the task. This process was repeated in stages, with annotators receiving additional data incrementally, ensuring continuous improvement in understanding and consistency in annotation. The scores assigned by the annotators are then averaged and normalized on a scale from 0 to 5. The dataset shows diverse degrees of actionability based on human annotation (Figure 3).

5 Experimentation and Results

We applied our fine-grained evaluation methodology for actionability using OpenAI's GPT-4-1106-preview model. This choice was made because GPT-4-1106-preview serves as the primary model for G-Eval and other baseline evaluators, ensuring a fair and consistent comparison between FinGrAct and existing evaluators such as G-Eval. The focus of our study is not on the model itself but rather the evaluation methodology, ensuring that the assessment framework remains the central point of analysis. We design four distinct experiments:

Comparison with SOTA Evaluation Methods: The first experiment aims to compare our evaluation methodology with existing SOTA models. The primary metric used for comparison is the correlation with human annotations.

Testing the External URL Content Retriever Component: The second experiment assesses the effectiveness of our external component, which aids in retrieving and processing link content. We analyze its impact on the correlation with human annotations across different models and evaluation methodologies.

Ego-Centric Bias Analysis: The third experiment investigates ego-centric bias and how our methodology influences this bias, which was initially identified in G-Eval by Liu et al. (2023).

FinGrAct on open-source models: The fourth experiment assesses the effectiveness of FinGrAct approach on open-source models.

In all experiments, the reported scores represent the rounded average of three independent runs for each model. This approach ensures a degree of consistency in the results.

5.1 Baseline Models

In the summarization domain, zero-shot SOTA LLM-based evaluators such as G-Eval and Prometheus are widely used to assess key properties such as coherence, faithfulness, and completeness, among others. G-Eval and Prometheus are general-purpose and can be adapted to evaluate new properties. This adaptability makes them valuable as SOTA baseline evaluators. Since these baseline models were never used to evaluate actionability, the adaptation prompts are in Appendix F for G-Eval, and in Appendix G for Prometheus.

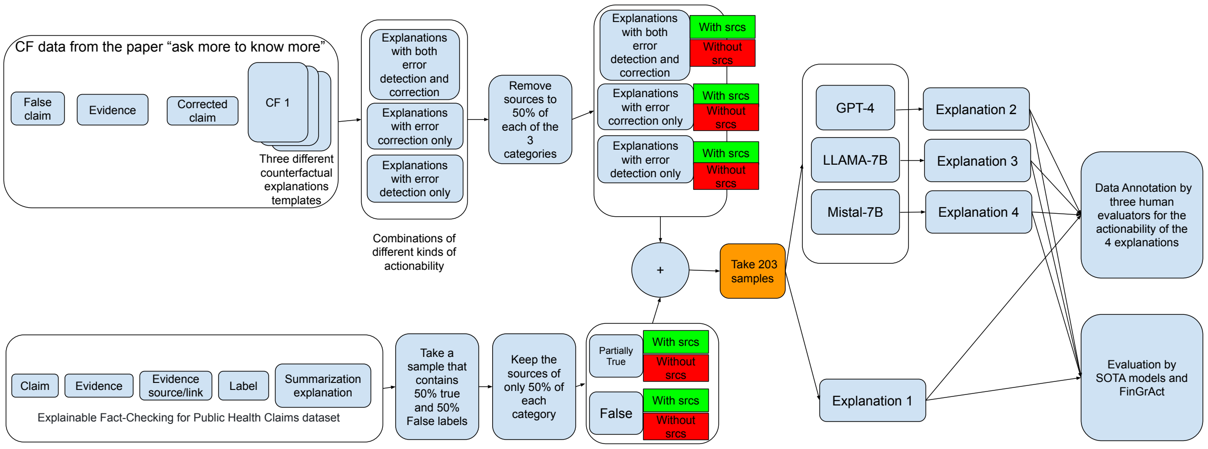
CF data from the paper "ask more to know more"
False claim
Evidence
Corrected claim
CF 1
Three different counterfactual explanations templates
Explanations with both error detection and correction
Explanations with error correction only
Explanations with error detection only
Combinations of different kinds of actionability
Remove sources to 50% of each of the 3 categories
Explanations with both error detection and correction
With srcs
Without srcs
Explanations with error correction only
With srcs
Without srcs
Explanations with error detection only
With srcs
Without srcs
+
Take 203 samples
GPT-4
LLAMA-7B
Mistal-7B
Explanation 2
Explanation 3
Explanation 4
Explanation 1
Data Annotation by three human evaluators for the actionability of the 4 explanations
Evaluation by SOTA models and FinGrAct
Claim
Evidence
Evidence source/link
Label
Summarization explanation
Explainable Fact-Checking for Public Health Claims dataset
Take a sample that contains 50% true and 50% False labels
Keep the sources of only 50% of each category
Partially True
With srcs
Without srcs
False
With srcs
Without srcs

Figure 2: **The dataset creation process**.The dataset combines counterfactual data from Dai et al. 2022, categorized by error detection, correction, and supporting sources, with non-counterfactual actionable explanations from (Kotonya and Toni, 2020b). The combined dataset was then sampled and used to generate three additional explanations from three LLMs to analyze ego-centric bias.

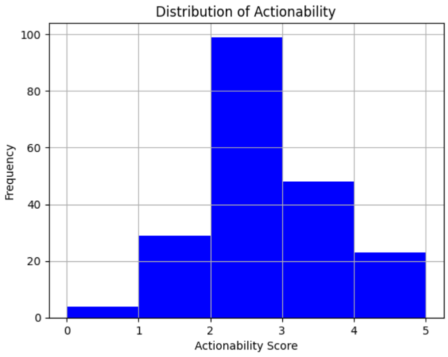

Figure 3: Distribution of actionability in the dataset based on average human annotation rating. This distribution shows that the dataset has diverse degrees of actionability.

5.2 Experiment 1: Comparison with SOTA Evaluation Methods

Setup: The original parameter settings for both G-Eval and Prometheus are preserved to ensure a fair comparison: G-Eval: temperature = 1, n = 20, top_p = 1 Prometheus: temperature = 1, top_p = 0.9. For FinGrAct, we set the temperature to zero and clear the conversation history before processing each new sample in GPT-4, following best practices from prior research (Shen et al., 2023; Song et al., 2024b) to enhance reproducibility. To evaluate FinGrAct against Prometheus and G-Eval, human annotators' scores serve as the ground truth benchmark. The performance of each evaluation methodology is measured using two correlation metrics: Pearson correlation and Kendall's tau correlation with human scores. In this setup, the links and sources within the explanations are evaluated without access to the web. Instead, LLMs assess the relevance of these sources solely based on their pre-existing, and potentially outdated, knowledge. This means that the evaluation of sources relies on the model's internal representations rather than real-time verification.

Results: In this experiment, as shown in Table 1, the Pearson correlation with human evaluations indicates that G-Eval achieves 0.147, Prometheus scores 0.328, and FinGrAct attains 0.46. Notably, FinGrAct correlates better than Prometheus (13.2% greater Pearson Correlation) demonstrating its superiority over the adapted SOTA evaluators in aligning with human judgments. Similarly, for Kendall's tau correlation, G-Eval achieves 0.117, Prometheus scores 0.294, and FinGrAct reaches 0.409. Again, FinGrAct outperforms Prometheus by 11.5%, reinforcing its effectiveness in producing evaluations that better correlate with human assessments.

Analysis: LLMs tend to achieve higher correlation with human annotations when provided with more detailed instructions. This could explain why Prometheus exhibits a higher correlation with human evaluations compared to G-Eval, as Prometheus requires a scoring rubric and a clear definition of the property being evaluated, whereas G-Eval relies solely on the property definition.

Prometheus uses a single, complex multi-class classification approach with six score levels (0 to 5) for evaluating explanations, while FinGrAct simplifies the process by breaking it into three independent subtasks (error detection, correction, and sup-

	Without UCR			With UCR		
	G-Eval	Prome-theus	FinGr-Act	G-Eval	Prome-theus	FinGr-Act
Pearson	0.147	0.328	0.460	0.213	0.405	**0.520**
Kendall	0.117	0.294	0.409	0.207	0.341	**0.419**

Table 1: The following table presents the Pearson and Kendall Tau correlations between human annotator and SOTA evaluators for explanations in the combined dataset. The first half of the table displays the correlation values without incorporating the URL textual content retriever (UCR), while the second half shows the correlations after its inclusion. The underlined scores are the highest scores without incorporating the UCR, while the bold scores are those after incorporating it. All the p-values are less than 0.05.

	G-Eval	Promethuse	FinGrAct
# of scores > human scores + 2	99	52	17
# of scores < human scores - 2	12	10	4

Table 2: The analysis examines ego-centric bias in evaluator scoring across 203 samples. It identifies cases where an evaluator overestimates actionability by scoring at least 2 units higher than human annotations and instances where it underestimates actionability, with human scores exceeding the evaluator's by 2 or more units. This assessment highlights discrepancies between model evaluations and human judgment.

porting links evaluation), each with three levels (0 for none, 1 for partial, 2 for complete). **This modular fine-grained design appears to improve LLM performance and correlation with human annotation.** The results in Table 1 are further confirmed in Experiment 5 in Appendix C where FinGrAct is proved to be superior w.r.t. correlation with human evaluation even with less detailed instructions provided for human annotators.

Manual analysis revealed that the main error pattern is LLMs, lacking web access, often assume all provided URLs are functional, inflating evaluation scores. Detailed examples are in Appendix A.

5.3 Experiment 2: Testing the External URL Content Retrieval Component

Setup: The same setup outlined in Section 5.2 is used, with one key modification: before constructing the prompt, the links are processed. The source pages of the links are scraped, their text is extracted, and then summarized using the UCR Section 3.1. Next, the summarized content from all linked sources is concatenated and incorporated into the prompt. The LLM is then tasked with evaluating whether the aggregated content from these URLs is both relevant to and supportive of the explanation and the needed corrections to the claim. Finally, the actionability score is computed as mentioned in Section 3.2.

Results: In Table 1, incorporating the UCR —which retrieves and integrates the content of the linked sources into the prompt instead of relying solely on the LLM's internal knowledge—led to an increase in correlation with human annotations across all SOTA evaluators in both Pearson and Kendall's correlations. G-Eval's Pearson correlation increased from 0.147 to 0.213, reflecting a 6.6% improvement, while its Kendall's correlation showed a 9% increase. Prometheus showed a 7.7% improvement in Pearson correlation and a 4.7% increase in Kendall'. FinGrAct's Pearson correlation improved by 6%, and its Kendall's by 1%.

Analysis: The results of Experiment 2 indicate that adding real-time source content through the UCR improved evaluation accuracy for actionability, but the improvement was less than expected. After investigation, this is mainly because the UCR cannot process non-text elements like images or JavaScript-rendered content, leading to gaps in extracted information and lower scores. In contrast, human evaluators can interpret such content, resulting in higher scores and a discrepancy. Also, when UCR is implemented, FinGrAct tends to assign higher scores in error correction and detection, as it incorporates the retrieved web content into its explanations. More details are in Appendix B.

5.4 Experiment 3: Ego-Centric Bias Analysis

In their study, Liu et al. (2023) identified a bias in evaluators, where they tend to favor their own model's generations over those from other models, even when the latter are objectively better. Ye et al. (2024) addressed this issue, referring to it as 'ego-centric Bias,' and we adopt this terminology in our work. The purpose of this study is to compare the effect of ego-centric bias on our fine-grained evaluation "FinGrAct" and on other SOTA evaluators. In this paper, we propose a simple yet effective method for identifying this bias within the context of actionability evaluation in explainable AFC when the probability distribution of LLM generations is not available.

We identify biased samples by observing that evaluators tend to assign significantly higher scores to explanations generated by their own underlying

LLMs compared to human ratings. For instance, G-Eval exhibits a preference for GPT-4-generated explanations, even when alternative explanations may be superior. We implement a Likert-scale scoring system ranging from 0 to 5, with a tolerance of a 1-point difference between human and LLM scores. If multiple annotators rate an explanation as 2 and LLM assigns a 3, the sample is excluded from bias analysis. If LLM scores the same explanation as 4 or 5, it is classified as ego-centric.

Setup: To measure ego-centric bias, each evaluator is tasked with assessing explanations generated by its underlying LLM. For instance, Prometheus evaluates Mistral-generated explanations, while G-Eval and FinGrAct evaluate GPT-4-generated explanations. Their evaluations are then compared against human annotations, and instances of bias are identified and counted. The scores from the three human annotators were averaged and compared against the mean scores from three evaluation runs for each automatic evaluator. G-Eval exhibited the highest variance, with 113 biased samples in the first run, 101 in the second, and 84 in the third, averaging 99. Prometheus was more stable with 55, 50 and 53 biased samples for the three runs. FinGrAct showed the least variance, with 17, 19 and 17 biased samples in the three runs. Instances where the evaluators underestimate actionability relative to human judgments are recorded to determine whether ego-centric bias or underestimation contributes more to the misalignment between automated evaluators and human assessments.

Results: Ego-centric bias contributes more to the misalignment between human annotations and LLM-based evaluations than underestimation. G-Eval exhibits ego-centric bias in 99 out of 203 samples (48.7%), whereas underestimation occurs is 5.9%. Prometheus demonstrates bias in 26% of cases, while underestimation accounts for 5%. FinGrAct shows 8.4% bias and 2% underestimation.

Analysis: In G-Eval, the evaluation relies primarily on the definition of the property being measured as shown in Figure 7 in Appendix F. Prometheus improves upon this by incorporating a detailed scoring rubric, (as shown in the Prometheus prompt Figure 9 in Appendix G). FinGrAct employs the most structured approach, implementing a detailed framework where claims are segmented, errors are explicitly identified, and corrections are validated along with supporting links, each scored in granular detail as mentioned in Appendix H. This evaluation process explains why FinGrAct exhibits the lowest ego-centric bias and the fewest mis-alignments due to underestimation.

	G-Eval Prompt		Prometheus Prompt		FinGrAct Prompt	
	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall
LLAMA-3.1	0.014	0.002	0.366	0.327	**0.431**	**0.382**
Mistral-7B	0.244	0.248	0.246	0.255	**0.293**	**0.270**

Table 3: The following table presents the Pearson and Kendall Tau correlations between human annotators and two open-source models (LLAMA-3.1 and Mistral-7B) when given the prompts of G-Eval, Prometheuse and FinGrAct. The highest correlation scores are in bold, showing that FinGrAct is superior due to its structured and fine-grained evaluation. All the p-values < 0.05.

5.5 Experiment 4: FinGrAct with open-LLMs

Setup: To demonstrate that the fine-grained framework is the key factor behind FinGrAct's higher correlation with human annotations, we applied FinGrAct prompts to LLaMA-3.1–8B and Mistral-7B. In earlier experiments, FinGrAct was implemented using GPT-4, a larger model. This experiment aims to isolate the impact of the framework, independent of model size or capacity.

Results: We conducted the experiments by prompting both LLaMA-3.1 and Mistral-7B using FinGrAct's fine-grained evaluation prompts. As shown in Table 3, this experiment confirms that the structured and fine-grained evaluation used in FinGrAct correlates better with human annotations.

Analysis: The results indicates that the superiority of FinGrAct in experiments 1 and 2 isn't caused by depending on GPT-4 , a powerful and large LLM. LLaMA 3.1-8B demonstrates strong performance, approaching that of GPT-4 despite its smaller size. It even outperforms Mistral-7B when using the Prometheus prompts, despite Prometheus being based on Mistral. This highlights the critical role of fine-tuning in enhancing Mistral's capabilities. Notably, without fine-tuning, LLaMA 3.1 achieves better correlation than Mistral when provided with detailed instructions. We also observe that LLM performance improves significantly on simpler tasks, such as 3-class classification [0–2] (FinGrAct), compared to more complex ones like 6-class classification [0–5] (Prometheus).

6 Conclusion

The paper introduces FinGrAct, a fine-grained evaluation method for assessing actionability in AFC, a crucial but underexplored property. The study shows that FinGrAct outperforms SOTA evaluators, achieving the highest Pearson and Kendall correlation with human judgments and exhibiting the lowest ego-centric bias, making it more reliable. Additionally, incorporating retrieved and summarized content from referenced sources further improved actionability evaluation across all models. These findings establish FinGrAct as a superior framework for assessing actionability in AFC.

Limitations

The limitations can be summarized in the following points:

1. The URL content retriever (UCR) component is currently limited to extracting textual content from the provided URLs. This restriction led to performance issues in instances where the referenced URLs contained primarily images or JavaScript-based content, as no retrievable text was available. As a result, these cases were misinterpreted, affecting the overall evaluation accuracy. In future work, we plan to develop a multimodal URL content retriever capable of processing both textual and non-textual content, including images and JavaScript-rendered elements. This enhancement will ensure more comprehensive content retrieval, leading to a more accurate and reliable evaluation.

2. All experiments in this study were conducted using a zero-shot learning approach. Fine-tuning was not explored due to its high computational and financial cost, particularly when applied to commercial LLMs like GPT-4. Future work may consider cost-effective fine-tuning strategies or alternative methods to enhance evaluation performance without incurring significant resource demands.

3. Our study primarily focused on ego-centric bias in LLM-based evaluations. However, in future work, we plan to explore other types of biases, including cross-model biases, where different LLMs may exhibit preferential treatment toward explanations generated by certain other LLMs. This broader analysis will provide a more comprehensive understanding of biases in LLM-based evaluation systems and help develop fairer and more reliable evaluation methodologies.

Ethics Statement

This paper focuses on designing an automatic evaluator utilizing LLMs for explaniable AFC using a combination two benchmark datasets. Some errors might be induced in the evaluation of explanations as the evaluator is LLM based.

Acknowledgments

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Grant No. RGPIN-2022-04789, and by the Digital Research Alliance of Canada (https://www.alliancecan.ca/en). We would like to thank the annotators and the reviewers for their valuable feedback.

References

Shih-Chieh Dai, Yi-Li Hsu, Aiping Xiong, and Lun-Wei Ku. 2022. Ask to know more: Generating counterfactual explanations for fake claims. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2800–2810, New York, NY, USA. Association for Computing Machinery.

Islam Eldifrawi, Shengrui Wang, and Amine Trabelsi. 2024. Automated justification production for claim veracity in fact checking: A survey on architectures and approaches. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6679–6692.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Darius Feher, Abdullah Khered, Hao Zhang, Riza Batista-Navarro, and Viktor Schlegel. 2025. Learning to generate and evaluate fact-checking explanations with transformers. *Eng. Appl. Artif. Intell.*, 139(PA).

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6556–6576.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2023. Tigerscore: Towards building explainable metric for all text generation tasks. *Transactions on Machine Learning Research*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models, 2024. *URL https://arxiv. org/abs/2310.08491*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Neema Kotonya and Francesca Toni. 2020a. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Neema Kotonya and Francesca Toni. 2020b. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Neema Kotonya and Francesca Toni. 2024. Towards a framework for evaluating explanations in automated fact verification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16364–16377.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv e-prints*, pages arXiv–2404.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4215–4233, Singapore. Association for Computational Linguistics.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024a. Finesure: Fine-grained summarization evaluation using llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 906–922.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024b. FineSurE: Fine-grained summarization evaluation using LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 906–922, Bangkok, Thailand. Association for Computational Linguistics.

Budi Susanto, Ridi Ferdiana, and Teguh Bharata Adji. 2024. Performance of traditional and dense vector information retrieval models in code search. In *2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT)*, pages 52–57. IEEE.

Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. 2024. Foundational autoraters: Taming large language models for better automatic evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17086–17105.

Rui Xing, Timothy Baldwin, and Jey Han Lau. 2025. Evaluating evidence attribution in generated fact checking explanations. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5475–5496.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. Instructscore: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. In *Neurips Safe Generative AI Workshop 2024*.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023. Flask: Fine-grained language model evaluation based on alignment skill sets. *arXiv preprint arXiv:2307.10928*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Shiyue Zhang, David Wan, Arie Cattan, Ayal Klein, Ido Dagan, and Mohit Bansal. 2024. Qapyramid: Fine-grained evaluation of content selection for text summarization. *arXiv e-prints*, pages arXiv–2412.

A Experiment 1: Comparison with SOTA Evaluation Methods analysis

Regarding the error patterns, most errors occur in the evaluation of the links provided in the explanation because the model assumes the provided URL is functional, even when it is not. Here is an example:

- Claim: Catherine Zeta-Jones is a Pisces.
- Evidence:Catherine Zeta-Jones ,born 25 September 1969 is a Welsh actress.
- Explanation: If we were to say 'Catherine Zeta-Jones is a Pisces.' but say 'Catherine zeta-jones is a welsh actress.', the claim would be correct. link: https://memorydelta.fandom.com/wiki/Catherine_Zeta-Jones

- Error Segmentation phase result: [{'sentence': "Catherine Zeta-Jones is a Pisces.', 'reason': "Catherine Zeta-Jones was born on September 25, which makes her a Libra, not a Pisces.', 'correction': "Catherine Zeta-Jones is a Libra.'}]
- Explanation Evaluation Phase results:[{'error': "Catherine Zeta-Jones was born on September 25, which makes her a Libra, not a Pisces.', 'response': 'No', 'correction': 'Yes', 'supporting_links': 'Yes'}]
- Scores: error detection:0, error correction:2, supporting links: 2
- final score: 3
- human evalution: 2

Regarding failure cases, we tried to give the evaluator a true claim with its explanation. Actionability can be only measured according to its definition for False and partially true claims, however, the result when providing a true claim to FinGrAct was interesting. Here is an example of a true claim inputted with its explanation to FinGrAct:

- Claim: Moscow is in a country.
- Evidence: Moscow is the capital and most populous city of Russia , with 13.2 million residents within the city limits and 17.8 million within the urban area . Moscow has the status of a Russian federal city.
- Explanation: Moscow is indeed a city, but it is the capital city of the country called Russia.

- Error Segmentation phase result: [{'sentence': 'Moscow is in a country.', 'reason': 'no error', 'correction': "}]
- Explanation Evaluation Phase results:[{'error': 'no error', 'response': 'No', 'correction': 'No', 'supporting_links': 'No'}]
- Scores: error detection:0, error correction:0, supporting links: 0
- final score: 0

This should not be considered a failure case, as actionability can only be measured in false or partially true claims. However, the evaluator scores must be different in true claims from false claims with zero degree of actionability

Regarding how the model handles partially true claims. It handles it the same way it handles false claims by identifying the parts that are not 100% true as errors. Here is an example:

- Claim: Novartis employees may have violated trial protocol in Japan.
- Evidence: Employees of Novartis Pharma K.K. (NPKK) transferred some data from research centers to a Tokyo hospital when that data should have been sent directly without first passing through Novartis hands, according to a report by Japanese broadcaster NHK that was picked up by the Wall Street Journal. "NPKK is currently investigating the allegations," Novartis said in a statement. The data was from a not yet fully enrolled 55-patient trial testing the Novartis cancer drug Tasigna, the company said. "Novartis Pharma K.K. is aware of the media report regarding a small investigator-initiated clinical study ... conducted to assess side effects in patients with chronic myelocytic leukemia," Novartis said in a statement. "NPKK has conducted employee trainings on proper protocol related to investigator-initiated clinical studies and believes that any involvement of our medical representatives in investigator-initiated clinical studies is inappropriate." The University of Tokyo Hospital said it was re-examining test results it had received but had uncovered no evidence that Novartis employees had manipulated any of the data during the transfers, according to the Wall Street Journal report.
- Explanation: Swiss drugmaker Novartis said on Friday it was looking into a report that employees of is Japanese unit may have violated clinical trial

protocol by handling data from a small independent study of one of its cancer drugs.

- The error segmentation phase results: [{’sentence’: ’Novartis employees may have violated trial protocol in Japan.’, ’reason’: ’The subclaim is vague and does not specify the nature of the potential protocol violation.’, ’correction’: ’Employees of Novartis Pharma K.K. transferred some data from research centers to a Tokyo hospital inappropriately, which may have violated trial protocol.’}]
- Explanation Evaluation Phase results: [{’error’: ’The subclaim is vague and does not specify the nature of the potential protocol violation.’, ’response’: ’Yes’, ’correction’: ’Yes’, ’supporting_links’: ’No’}]
- Scores: error detection:2, error correction:2, supporting links: 0
- final score: 3

B Experiment 2: Testing the External URL Content Retrieval Component analysis

It is worth noting that when UCR is implemented, FinGrAct tends to assign higher scores in error correction and detection. This is because it treats the retrieved web content as part of the explanation, given that the explanation references the URL containing the relevant information. This is the reason for 73% of the score mismatch with the human annotation. Here is an example:

- Claim: Usain Bolt is a whale.

- Evidence: Usain St Leo Bolt , born 21 August 1986 is a Jamaican sprinter.

- Explanation: If we were to say ’Usain Bolt is a whale.’ but say ’The whale is the name of the whale.’, the claim would be correct. https://en.wikipedia.org/wiki/Usain_Bol

- Error Segmentation phase result: [{’sentence’: ’Usain Bolt is a whale.’, ’reason’: ’The subclaim incorrectly categorizes Usain Bolt as a whale, which is a marine mammal, while the evidence clearly states that he is a Jamaican sprinter.’, ’correction’: ’Usain Bolt is a Jamaican sprinter.’}]

- Content retrieved: Bolt is the only sprinter to win Olympic 100m and 200m titles at three consecutive Olympics (2008, 2012, and 2016).

- Explanation Evaluation Phase results:[{’error’: ’The subclaim incorrectly categorizes Usain Bolt as a whale, which is a marine mammal, while the evidence clearly states that he is the only Jamaican sprinter to win Olympic 100m and 200m.’, ’response’: ’No’, ’correction’: ’Yes’, ’existing_links’: ’Yes’, ’related_links’: ’Yes’, ’supporting_links’: ’Yes’}]

- Scores: error detection:0, error correction:2, supporting links: 3

- final score: 4

-human evaluation: 2

	G-Eval		Prometheus		FinGrAct	
	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall
definition only annotation	0.444	0.343	0.264	0.215	**0.671**	**0.504**
annotation with score rubric	0.330	0.221	0.403	0.353	**0.564**	**0.458**

Table 4: Pearson and Kendall correlations of G-Eval, Promethuse, and FinGrAct across different human annotation styles. The highest correlation scores are in bold.

C Experiment 5: Experimentation with data with different styles of human annotations

The aim of this experiment is to evaluate how different annotation instructions provided to human annotators affect the performance of FinGrAct and other state-of-the-art evaluators. This experiment uses a representative subset of 50 samples, rather than the entire dataset of 203 samples, since the goal is to obtain an approximate measure of the impact of varying instructions. The objective is not to create new datasets for each annotation style. Since this experiment uses only a subset of the dataset, the correlation results may be higher than those from the full dataset. This could be because the subset might contain some of the easier samples, and although the samples are randomly selected, this can lead to inflated correlation scores. What is most important is that the subset adequately represents all categories and degrees of actionability.

Setup: A different group of human annotators was tasked with providing actionability scores. To assess the impact of different data annotation approaches on the correlation between human-provided scores and those generated by FinGrAct, a representative random sample of 50 claims, along with their supporting evidence and explanations, was selected. Two distinct setups were used:

Setup 1: Annotators were provided only with the definition of actionability. They were to assign an actionability score to each of the 50 samples based on their understanding of this definition.

Actionability definition in explainable AFC: Actionability in AFC refers to providing factual corrections for identified errors in a non-factual claim, supported by sources and references.

Setup 2: Annotators received both the definition of actionability and a scoring rubric similar to the one used in the Prometheus prompt shown in Figure 9. They were asked to score the 50 samples, leveraging both the definition and the rubric.

The scoring rubric:

- "score1 description":"The model detects the error or misalihnment without correcting it. In addition sources are not mentioned",

- "score2 description":"The model corrects the error or misalignment, but doesn't point out where the error is. In addition sources are not mentioned"

- "score3 description":"The model typically detects the error or misalignment and explicitly mentions it. The model also provides correction of the error. In addition sources are not mentioned "

- "score4 description":"The model consistently detects the error or misalignment and explicitly mentions it. The model also provides correction of the error. In addition sources are mentioned"

- "score5 description":"The model excels in the detection of errors or misalignment and explicitly mentions it. The model also provides correction of the error. In addition sources are mentioned. The content of the sources or links supports the response."

Results: As shown in Table 4, the Pearson and Kendall correlation with FinGrAct is higher than any other method, indicating that the superiority of the FinGrAct is not because of a certain style of data annotation.

Analysis: Even with varying sets of annotator instructions, FinGrAct consistently outperforms other state-of-the-art (SOTA) baselines. As presented in Table 4, when human annotators were provided only with the definition for actionability, G-Eval demonstrated a stronger correlation with human judgments compared to Prometheus. This is expected, as G-Eval also relies solely on the actionability definition, while Prometheus uses a more detailed scoring rubric. However, FinGrAct achieves an even higher correlation than G-Eval, suggesting that its comprehensive, instruction-based approach aligns more closely with human evaluations than relying on GPT-4's prior knowledge alone.

When human annotators were provided with a scoring rubric similar to the one shown in Figure 9, G-Eval exhibited lower correlation with the human annotations compared to Prometheus. This outcome is expected since Prometheus directly follows the same rubric, rather than relying on GPT-4's prior knowledge. However, somewhat surprisingly, FinGrAct achieved the highest correlation.

A deeper analysis revealed that Prometheus sometimes struggles to strictly adhere to its provided scoring rubric, indicating that the task is quite complex. Specifically, Prometheus performs a multi-class classification task, assigning scores from 0 to 5 (six classes) for the explanation after assessing whether it detected errors in the claim, corrected those errors, and cited supporting sources.

In contrast, FinGrAct breaks the task down into three simpler classification subtasks: three classes 0 for none, 1 for partial and 2 for complete error detection, similarly three classes for error correction, and three classes for evaluating the cited supporting links. Additionally, the final score is computed outside of the underlying LLM and according to Algorithm 1. This suggests that large language models perform significantly better when given simpler, modular subtasks rather than one complex detailed task.

D Human annotation details

Three human annotators, MSc students familiar with NLP tasks, aged between 22 and 30 years, were tasked with evaluating the actionability of 203 samples, each containing four explanations. They were provided with detailed instructions, clear examples and a video demo to ensure consistency in their evaluations (see Figure 4).

The annotation process followed an iterative approach: initially, a subset of the dataset was assigned for evaluation, and annotators provided feedback on any ambiguities. Based on their input, the instructions were refined and improved to enhance clarity. Significant disagreements were addressed through discussions and successive refinements of the guidelines, ensuring a more consistent and reliable evaluation process.

We used Krippendorff's alpha to measure interannotator agreement, as it is well-suited for Likert-scale data with multiple annotators. The resulting Krippendorff's alpha of 0.863 reflects the high level of agreement, which is expected due to the following factors:

1. **Clear Annotation Guidelines** – annotators have strict, well-defined rules. In addition, the process is iterative, fine-grained, and a video demonstration is provided to the annotators.

2. **Objective or Easy-to-Classify Data** – Tasks with minimal ambiguity (e.g., labeling with '0,1 or 2' labels like "none, partial and full") often lead to high agreement.

3. **Trained Annotators** tend to agree more than crowd-sourced or untrained annotators.

The annotators volunteered to evaluate the actionability of the explanations for the claims. Given this, we can assert that their annotations were conducted solely based on their understanding of the provided instructions.

E Explanations generated by the LLMs of the SOTA evaluators

To study ego-centric bias, we prompted the generative LLMs that serve as the foundation for our state-of-the-art (SOTA) evaluators to generate actionable explanations. These explanations are then assessed by both the same SOTA evaluators and human annotators, enabling a comparative analysis to detect potential biases.

This process consists of two steps to ensure diversity in explanations, including both those with and without supporting links.

1. We generate explanations for all claims using the prompt shown in Figure 5.

2. We prompt the LLMs to generate supporting links for **some** claims while leaving other explanations without links. We use the prompt in Figure 6. This approach ensures a balanced dataset with explanations both with and without links.

F G-Eval Adaptation to Measure Actionability

G-Eval is widely used to assess various important properties in the summarization domain, but it has never been applied to measure actionability. This is because actionability is critical in explainable fact-checking, rather than in summarization.

Typically, G-Eval takes a transcript and a summary as inputs for evaluation. However, in this work, we adapt it for actionability assessment by changing the inputs to the claim, evidence, label, and the explanation to be evaluated.

The customizability of G-Eval makes it a go-to evaluator and baseline for researchers. Its concept is straightforward: provide the LLM with the definition of the property to be evaluated, give it general guidelines or instructions for the evaluation process, and then let the LLM act as a judge, determining the score based on the given definition and instructions.

This approach gives the LLM significant flexibility in its assessments. As a result, the strength of the LLM-as-a-judge plays a crucial role in the quality and reliability of G-Eval's evaluations.

We used two prompts with G-Eval:

1. **Actionability Evaluation Prompt** – This prompt includes a clear definition of actionability along with detailed evaluation instructions. This prompt is shown in Figure 7.

2. **Actionability Evaluation with URL Content Retrieval** – This prompt is identical to the first but incorporates the retrieved web content from all links mentioned in the explanation. This ensures that the evaluation considers external sources rather than relying solely on the LLM's prior knowledge. This prompt is shown in Figure 8.

G Prometheus Adaptation to Measure Actionability

Prometheus is designed to assess customized properties in a transcript, but it has never been applied to measure actionability. It utilizes Mistral-7B and follows a strictly structured prompt template, where the evaluator inputs the definition of the property to be assessed, along with explicit instructions for evaluating that property—similar to G-Eval.

What differentiates Prometheus is its rigid scoring framework. Unlike G-Eval, which allows the LLM more freedom in judgment, Prometheus requires a detailed scoring rubric, specifying exactly when to assign a score of 1, 2, 3, 4, or 5. Additionally, it requests feedback to provide insights into the reasoning behind the assigned score.

Typically, Prometheus evaluates summaries by taking a transcript, a summary, and a detailed scoring rubric as input. In this work, however, we adapt

You have a claim that needs to be fact-checked, evidence which is basically the trustworthy information that we can rely on to check if the claim is factual or not, a label that shows the prediction of our model on whether the claim is true or false, and the model explanation for its predicted label. If the claim aligns with the evidence, then its label will be true and vice versa.
The explanation should be evaluated based on the following criteria:

1. The number of factual errors that were detected in the claim and pointed out in the explanation compared to the number of all the factual errors on a scale from 0 to 2. This means that if no factual errors in the claim were detected then the score is zero, if some of the factual errors were detected then the score is one, and if **all** the factual errors were detected then the score is 2.

2. The number of factual errors that were corrected faithfully in the explanation on a scale from 0 to 2. This means that if no factual errors in the claim were corrected then the score is zero, if some of the factual errors were corrected then the score is one, and if **all** the factual errors were corrected then the score is 2.

3. The correctness of the resources/references, their relevance to the evidence, and their alignment with the evidence on a scale from 0 to 3. This means if there were no references mentioned then the score is zero, if the references are mentioned and the links are correct then the score is one,if the references are mentioned and are relevant to the evidence then the score is 2 and if the references are mentioned and are **both relevant and aligned with the evidence** then it gets a score of 3.

Example 1:
Claim: Adrienne Bailon is an accountant.
Evidence: Adrienne Eliza Houghton Bailon ; born October 24 , 1983 is an American singer-songwriter , recording artist , actress , dancer and television personality .
Label: FALSE
Explanation: Adrienne Bailon is an American singer-songwriter, recording artist, actress, dancer and television personality. Evaluation:

- percentage of factual errors detected in explanation: 0 out of 2 “no factual errors from the claim are mentioned explicitly in the explanation”

- percentage of factual errors corrected in explanation: 2 out of 2 “The error ‘accountant’ in the claim was corrected in the explanation’

- The correctness of the resources/references and their relevance: 0 out of 3 No references were mentioned

Example 2:
Claim: Junun is a book.
Evidence: Junun is a 2015 documentary film directed by Paul Thomas Anderson . Junun premièred at the 2015 New York Film Festival and was released on the MUBI film streaming service on October 9 and on iTunes on November 20, 2015 . Greenwood previously composed soundtracks for several Anderson films.
Label: FALSE
Explanation: If we were to say ’Junun is a 2015 documentary film directed by paul thomas anderson’ instead of ’book’, the claim would be correct. https:en.wikipedia.orgwikiJunun_(film)

- percentage of factual errors detected in explanation: 2 out of 2“ factual errors from the claim are mentioned in the explanation after the part ‘instead of’ ” and the error is explicitly obvious in the word “book”

- percentage of factual errors corrected in explanation: 2 out of 2“The error ‘book’ in the claim was corrected in the explanation to the word “film”

- The correctness of the resources/references and their relevance: 3 out of 3 “references mentioned are correct, aligned and relevant as I have opened the link and checked”

Example 3:
Claim: Earth is flat and green.
Evidence: Nasa images show that Earth is a huge blue round planet.
Label: FALSE
Explanation: The NASA images show that the Earth is huge and not green. https:science.nasa.govgalleryjupiter
Evaluation:

- percentage of factual errors detected in explanation: 1 out of 2 “ factual errors from the claim are mentioned in the explanation in part ‘not green’ ”. However, the error of Earth being ‘flat’ wasn’t mentioned

- percentage of factual errors corrected in explanation: 1 out of 2 “The error ‘green’ in the claim was corrected in the explanation, but it didn’t say that it was blue. The other error ‘flat’ was not corrected

- The correctness of the resourcesreferences and their relevance:1 out of 3 “references mentioned are correct but not relevant as I have opened the link and found out that it shows Jupiter, not Earth”

Figure 4: The instructions given to the three human annotators for evaluating the actionability

You will be given a claim, some credible information called the evidence, a label that shows whether the claim is true or false.

Your task is to generate an actionable explanation for the label of the claim based on the evidence.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Explanation Criteria:

Actionability - misinformation detection and factual correction backed up by supporting sources. The explanation should provide an indication of which parts of the claim include misalignment with the evidence. In addition, the explanation should provide a corrected version of the erroneous claim.

Evaluation Steps:

1. Read the claim, evidence and the explanation carefully.
2. Compare the claim with the evidence and identify the errors or misalignment parts between the claim and the evidence.
3. Generate an explanation that clearly mentions the errors detected in the claim and corrects these errors based on the evidence.
4. Don't respond with any information outside the provided evidence. Your are restricted to answer from the evidence only.

Claim:
{claim}

Evidence:
{evidence}

Label:
{label}

Figure 5: General prompt used on all the generative LLMs of the SOTA evaluators to generate actionable explanations. These explanations are used to measure the ego-centric bias of evaluators based on these underlying LLMs.

You will be given some information called the explanation.

Your task is to generate a correct and working web link for a source supporting the explanation.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Steps:

1. The web link provided should be correct and working.
2. The web link should open a page that has information relevant to the explanation.
3. The web link should open a page that has information supporting and in alignment with the explanation.

Explanation:
{explanation}

Figure 6: General prompt used on all the generative LLMs of the SOTA evaluators to generate supporting links to their explanations for some samples

You will be given a claim, some credible information called the evidence, a label that shows whether the claim is true or false, and an explanation for the label.

Your task is to evaluate the explanation on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Actionability (0-5) - misinformation detection and factual correction backed up by sources. The explanation should provide an indication of which parts of the claim include misalignment with the evidence.In addition, the explanation should provide a corrected version of the erroneous claim.

Evaluation Steps:

1. Read the claim, evidence and the explanations carefully.
2. Compare the claim with the evidence and identify the errors or misalignment parts between the claim and the evidence.
3. Assess how well the explanations cover the errors detected, the supporting sources provided, and the degree of error correction of the claim in each explanation.
4.If there is a link in the explanation, assess if the content of the link -based on your knowledge- is both relevant and supporting to the explanation.
5. Assign a score from 1 to 5 to the metric.
6. Your output should be scores only.

Claim:
{claim}

Evidence:
{evidence}

Label:
{label}

Explanation:
{explanation}

Evaluation Form (scores ONLY):
- Actionability:

Figure 7: G-Eval Prompt for evaluating actionability

it for actionability assessment by modifying the inputs to include the claim, evidence, label, explanation to be evaluated, and a structured scoring rubric.

You will be given a claim, some credible information called the evidence, a label that shows whether the claim is true or false, and an explanation for the label. The explanation might have a link. If this is true, the content of the link will be provided as well.

Your task is to evaluate the explanation on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Actionability (0-5) - misinformation detection and factual correction backed up by supporting sources. The explanation should provide an indication of which parts of the claim include misalignment with the evidence.In addition, the explanation should provide a corrected version of the erroneous claim.

Evaluation Steps:

1. Read the claim, evidence and the explanation carefully.
2. Compare the claim with the evidence and identify the errors or misalignment parts between the claim and the evidence.
3. Assess how well the explanation covers the errors detected and the degree of error correction of the claim in the explanation.
4. If there is a link in the explanation, assess if the content of the link provided is both relevant and supporting to the explanation.
5. Assign a score from 1 to 5 to the metric.
6. Your output should be scores only.

Claim:
{claim}

Evidence:
{evidence}

Label:
{label}

Explanation:
{explanation}

link content:
{link_content}

Evaluation Form (scores ONLY):
- Actionability:

Figure 8: G-Eval Prompt for evaluating actionability with URL content retriever

This approach reduces the LLM's flexibility in scoring but ensures greater consistency and reliability. As a result, Mistral-7B as an LLM-as-a-judge has outperformed G-Eval (GPT-4) in several benchmarks, making Prometheus the current state-of-the-art (SOTA) evaluator.

We used two prompts with Prometheus:

1. **Actionability Evaluation Prompt** – This prompt includes a clear definition of actionability along with detailed evaluation instructions and scoring ruberic. This prompt is shown in Figure 9.

2. **Actionability Evaluation with URL Content Retrieval** – This prompt is identical to the first but incorporates the retrieved web content from all links mentioned in the explanation. This ensures that the evaluation considers external sources rather than relying solely on the LLM's prior knowledge. This prompt is shown in Figure 10.

H FinGrAct and Fine-Grained Actionability Evaluation

The systematic and fine-grained evaluation approach in FinGrAct minimizes GPT-4's reliance on its own knowledge by breaking down the evaluation into small, manageable tasks. This structured methodology ensures a more objective and transparent assessment of actionability.

FinGrAct's Error Segmentation operates in three stages:

1. **Claim Breakdown** – The claim is broken down into atomic claims, making the evaluation more granular and precise.

2. **Error Detection** – Each atomic claim is examined for factual inaccuracies.

3. **Correction Proposal** – For every detected error, a correction is generated, ensuring that the explanation provides accurate and actionable insights.

By structuring the evaluation in this way, FinGrAct reduces subjectivity and enhances the reliability of actionability assessments.

An instruction (might include an Input inside it), a response to evaluate, and a score rubric representing a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of the response strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. The output format should look as follows: F̈eedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)¨
4. Please do not generate any other opening, closing, and explanations.

###The instruction to evaluate: You will be given a claim, some credible information called the evidence, a label that shows whether the claim is true or false, and an response that explains the label.
Evaluate the actionability of the response by examining misinformation detection and factual correction backed up by supporting sources. The response should provide an indication of which parts of the claim include misalignment with the evidence. In addition, the response should provide a corrected version of the erroneous claim and a web link or a source that it relies on.

###Claim:
{claim}

###Evidence:
{evidence}

###Label:
{label}

###Response to evaluate:
{response}

###Score Rubrics:

"criteria":"Is the model proficient in detecting and correcting the error or misalgnment between the response and the evidence and also providing supporting sources",

"**score1_description**":"The model detects the error or misalihnment without correcting it. In addition sources are not mentioned",
"**score2_description**":"The model corrects the error or misalignment, but doesn't point out where the error is. In addition sources are not mentioned",
"**score3_description**":"The model typically detects the error or misalignment and explicitly mentions it. The model also provides correction of the error. In addition sources are not mentioned ",
"**score4_description**":"The model consistently detects the error or misalignment and explicitly mentions it. The model also provides correction of the error. In addition sources are mentioned",
"**score5_description**":"The model excels in the detection of errors or misalignment and explicitly mentions it. The model also provides correction of the error. In addition, sources/links that have relevant and supporting content are included in the explanation."

Figure 9: Prometheus prompt for evaluating actionability

###Task Description:

An instruction (might include an Input inside it), a response to evaluate, and a score rubric representing a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of the response strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. The output format should look as follows: F̈eedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)¨
4. Please do not generate any other opening, closing, and explanations.

###The instruction to evaluate:

You will be given a claim, some credible information called the evidence, a label that shows whether the claim is true or false, and an response that explains the label. If the response contains a link, the content of the link will be provided as well. Evaluate the actionability of the response by examining misinformation detection and factual correction backed up by supporting sources. The response should provide an indication of which parts of the claim include misalignment with the evidence. In addition, the response should provide a corrected version of the erroneous claim and a web link or a source that it relies on. The content of the link should support the response.

###Claim:
{claim}

###Evidence:
{evidence}

###Label:
{label}

###Response to evaluate:
{response}

###link content:
{link_content}

###Score Rubrics:

"criteria":"Is the model proficient in detecting and correcting the error or misalgnment between the response and the evidence and also providing supporting sources",
"score1_description":"The model detects the error or misalihnment without correcting it. In addition sources are not mentioned",
"score2_description":"The model corrects the error or misalignment, but doesn't point out where the error is. In addition sources are not mentioned",
"score3_description":"The model typically detects the error or misalignment and explicitly mentions it. The model also provides correction of the error. In addition sources are not mentioned ",
"score4_description":"The model consistently detects the error or misalignment and explicitly mentions it. The model also provides correction of the error. In addition sources are mentioned",
"score5_description":"The model excels in the detection of errors or misalignment and explicitly mentions it. The model also provides correction of the error. In addition faithful sources are mentioned. The content of the sources or links supports the response."

Figure 10: Prometheus prompt for evaluating actionability after adding the URL content retriever

You will receive a claim, and some trustworthy reliable information called evidence. Your task is to divide the claim into multiple smaller subclaims/ atomic claims , then assess the factuality of each subclaim sentence based on the information provided in the evidence:

- no error: the subclaim aligns explicitly with the content of the evidence and is factually consistent with it.
- factuality error: the subclaim contains any factuality error.

Instruction:
First, compare each subclaim sentence with the evidence. Second, provide a single sentence explaining the factuality error in the subclaim and how to correct it based on the evidence.

Provide your answer in JSON format. Your answer should strictly be a list of dictionaries whose keys are "sentence", "reason" and "correction". An example of your output should be:
[{"sentence": "first subclaim", "reason": "your reason", "correction": "your correction"},
{"sentence": "second subclaim", "reason": "your reason", "correction": "your correction"}]

Claim:
{claim}
Evidence:
{evidence}

Figure 11: FinGrAct prompt for error segmentation. The output should include false atomic claims, their factual errors, and GPT-4's proposed corrections. The explanation will then be evaluated to check if it explicitly covers all detected errors and corrections.

Here is an example:

claim: Earth is flat and red.

Evidence: Nasa images shows that Eart is a blue marble shaped planet.

Explanation: The claim has two errors in earth's description. The errors are in the words 'flat' and 'red'. The correct version of the claim is : "Earth is round and blue". Check NASA images at: Check NASA images at https://explorer1.jpl.nasa.gov/galleries/earth-from-space

output of the error segmentation and correction stage: [{'sentence': 'Earth is flat', 'reason': 'The evidence explicitly states that Earth is a marble shaped planet, not flat.', 'correction': 'Earth is round.'}, {'sentence': 'Earth is red', 'reason': 'As per the evidence, Earth is blue, 'correction': 'Earth is blue'}]

The prompt for this is shown in Figure 11.

After detecting errors and generating corrections, the next step is to verify whether these elements are explicitly inferred from the explanation. This involves answering "yes" or "no" for each detected error and proposed correction to assess alignment with the explanation.

The goal is to determine whether:

1. The number of false atomic claims matches the number of errors mentioned in the explanation.
2. The number of detected corrections corresponds to the number of corrections provided in the explanations.

This process helps establish whether error detection and correction were partial or complete, which ultimately influences the final actionability score.

Continuing on the previous example of the claim that Earth is flat and red. Here is an example of the output: [{'error': 'The evidence explicitly states that Earth is a marble shaped planet, not flat', 'response': 'Yes', 'correction': 'Yes', 'supporting_links': 'Yes'}, {'error': 'As per the evidence, Earth is blue.', 'response': 'Yes', 'correction': 'Yes', 'supporting_links': 'Yes'}]

The "error" key in the JSON output represents a specific factual error identified during the error segmentation and correction stage. The "response" key is a boolean indicating whether the explanation explicitly mentions the identified error. The "correction" key is another boolean that shows whether the explanation includes the corresponding correction from the error segmentation and correction process. Finally, the "supporting_link" key is a boolean that signifies whether there is at least one link in the explanation with content that supports the correction, as assessed based on the LLM's prior knowledge.

Additionally, each correction in the explanation—mapped to a correction identified during the error segmentation and correction phase in FinGrAct—is verified to ensure it has at least one supporting link with relevant web content. An explanation is deemed fully actionable only if at least one link supports all corrections.

Sanity checks were implemented to prevent impossible scenarios. For example, the "related_links" key cannot be "yes" if "existing_link" is not, and the "supporting_link" boolean cannot be "yes" unless both "existing_links" and "related_links" are also "yes".

There are two methods for checking link content:

1. LLM's Prior Knowledge: The evaluator relies on the LLM's pre-existing knowledge from training to assess whether the link content aligns with the corrections. The corresponding prompt is shown in Figure 12.

2. URL Content Retrieval (UCR): The URL content retriever fetches the text content from the web page, which is then checked for align-

You will receive a list of json objects called the input. The input contains sentences with errors, the reason why they have errors and their corrections. In addition, you will receive a transcript called the ’explanation’. Your task is to assess if each of the errors can be inferred from the explanation, and if the corrections can be inferred from the explanation as well. In addition, check if the explanation has web links that support the corrections.

Instruction:

First, compare each error reason with the explanation.
Second, check if the error reason is inferred from the explanation and then response "Yes" or "No" for each error explicitly mentioned in the explanation.
Third, compare each correction with the explanation.
Fourth, check if the correction is inferred from the explanation and then respond with "Yes" or "No" for each correction.
Fifth, check if there are working web links in the explanation. The links content will mention if the link is working or not, and then respond with "Yes" or "No".
Sixth, Provide your output in JSON format. The output should be a list of json objects whose keys are "error", "response", "correction", and "supporting_links".
An example of your output: [{"error": "error reason of first sentence", "response": "Yes", "correction": "Yes", "existing_links": "Yes", "related_links": "Yes", "supporting_links": "Yes"},
{"error": "error reason of second sentence", "response": "No", "correction": "Yes", "existing_links": "Yes", "related_links": "No", "supporting_links": "No"},
{"error": "error reason of third sentence", "response": "Yes", "correction": "No", "existing_links": "Yes", "related_links": "Yes", "supporting_links": "No"}]

Json input:
{input_json}

Explanation:
{explanation}

Figure 12: FinGrAct Prompt for evaluating the actionability aspects. This prompt represents the actionability evaluation stage and the source evaluation stage.

ment with the corrections. The corresponding prompt is shown in Figure 13.

You will receive a list of json objects called the input. The input contains sentences with errors, the reason why they have errors and their corrections. In addition, you will receive a transcript called the ’explanation’ and another transcript called web links content. Your task is to assess if each of the errors can be inferred from the explanation, and if the corrections can be inferred from the explanation as well. In addition, check if the explanation has web links that support the corrections.

Instruction:

First, compare each error reason with the explanation.
Second, check if the error reason is inferred from the explanation and then response "Yes" or "No" for each error explicitly mentioned in the explanation.
Third, compare each correction with the explanation.
Fourth, check if the correction is inferred from the explanation and then respond with "Yes" or "No" for each correction.
Fifth, check if there are working web links in the explanation. The links content will mention if the link is working or not, and then respond with "Yes" or "No".
Sixth, check the provided content of these web links, if the content is related, and then respond with "Yes" or "No" for each related link.
Seventh, check the provided content of these web links, if the content supports the explanation, and then respond with "Yes" or "No" for each related link.

Provide your output in JSON format. The output should be a list of json objects whose keys are "error", "response", "correction", and "supporting_links".
An example of your output: [{"error": "error reason of first sentence", "response": "Yes", "correction": "Yes", "existing_links": "Yes", "related_links": "Yes", "supporting_links": "Yes"},
{"error": "error reason of second sentence", "response": "No", "correction": "Yes", "existing_links": "Yes", "related_links": "No", "supporting_links": "No"},
{"error": "error reason of third sentence", "response": "Yes", "correction": "No", "existing_links": "Yes", "related_links": "Yes", "supporting_links": "No"}]

Json input:
{input_json}

Explanation:
{explanation}

Links content:
{links_content}

Figure 13: FinGrAct Prompt for evaluating the actionability aspects. This prompt represents the actionability evaluation stage and the source evaluation stage.