

# Factcheck-Bench: Fine-Grained Evaluation Benchmark for Automatic Fact-Checkers

Anonymous ACL submission

## Abstract

The increased use of large language models (LLMs) across a variety of real-world applications calls for mechanisms to verify the factual accuracy of their outputs. In this work, we present Factcheck-Bench, a holistic end-to-end framework for annotating and evaluating the factuality of LLM-generated responses, which encompasses a multi-stage annotation scheme designed to yield detailed labels for fact-checking and correcting not just the final prediction, but also the intermediate steps that a fact-checking system might need to take. Based on this framework, we construct an open-domain factuality benchmark in three-levels of granularity: claim, sentence, and document. We further propose a system, Factcheck-GPT, which follows our framework, and we show that it outperforms several popular LLM fact-checkers. We make our annotation tool, annotated data, benchmark, and code available at <http://anonymous.for.review>

## 1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities to generate naturally sounding answers over a broad range of human inquiries (OpenAI, 2023). Yet, they often generate answers that contradict real-world facts (Bang et al., 2023; Borji, 2023; Guiven, 2023; Augenstein et al., 2023). This degrades LLM’s performance and undermines its reliability, representing a significant obstacle for deployment, especially in high-stake settings, e.g., clinical, legal, and financial (Weidinger et al., 2021; Bender et al., 2021; Kapoor et al., 2024).

Before LLMs, most prior work investigated hallucinations of conditional text generation for specific tasks, such as abstract summarization, dialogue generation, and machine translation (Ji et al., 2023). This prior research was either highly task-specific, with gold standard references, or focusing on short statements, in which automatic evaluation

by rule-based matching or semantic similarity measurements with references was feasible (Maynez et al., 2020; Martín et al., 2022). However, in the case of free-form LLM generations over open domains, there is no single gold standard reference answer that can be used to assess the factual correctness of a long free-text model output (Wang et al., 2024). This makes the factual evaluation of open-domain LLM responses non-trivial, either depending on manual verification or on automatic fact-checkers, e.g., FActScore and FacTool (Min et al., 2023; Chern et al., 2023).

Human assessment is costly (Nakov et al., 2021; Chen et al., 2023), while existing automatic fact-checking systems are often unreliable. How to evaluate and to improve the accuracy of automated fact-checkers is critical for producing dependable LLM factuality evaluations.

Recent work on fact-checking compared the system predictions to human annotations. This only assesses the final verification results, i.e., whether a claim or a document is true or false (Chern et al., 2023; Chen et al., 2023; Dhuliawala et al., 2023). Thus, the system is regarded as a black box. The lack of evaluation of the intermediate steps that the system has taken, makes it difficult to understand which specific system component(s) led to an erroneous factual judgement. Moreover, prior studies used disparate evaluations, rendering them hard to compare (Wang et al., 2024). See more detailed discussion in Appendix B.

To address these issues, we introduce **Factcheck-Bench**, a benchmark to evaluate systems for automatic fact-checking of LLM output, with each model response annotated with labels covering eight steps that might occur in the automatic detection and correction of factual errors (see Figure 2 in the Appendix). This enables the evaluation of the results of the intermediate component modules in various automatic fact-checkers in a unified framework. We make the following contributions:

- We propose a holistic and fine-grained framework for annotating the factuality of LLM outputs. It also serves as an end-to-end solution for detecting and correcting factual errors.
- We construct **Factcheck-Bench**, a document-level claim-based fact-checking benchmark for LLMs, covering both detection and revision, to facilitate the evaluation and the analysis of automatic fact-checking systems and their internal components. We also build and release a flexible annotation tool.
- We propose **Factcheck-GPT**, which follows our framework, and we show that it outperforms several popular LLM fact-checkers.

## 2 Fine-grained Annotation Framework

To evaluate all possible intermediate results of automatic fact-checking systems, the framework should cover as many fine-grained steps as possible. Based on our review of most prior systems and datasets, we frame the automated detection and correction of factual errors for outputs of LLMs into eight subtasks: (1) decomposition; (2) decontextualisation; (3) checkworthiness identification; (4) evidence retrieval and collection; (5) stance detection; (6) correction determination; (7) claim correction and (8) final response revision. Figure 2 presents the overview of the whole procedure, coupled with an example flowing through each subtask.

**(1) Decompose** Given a response  $R$  generated by a LLM, it is infeasible to fact-check the whole document at once, especially when it is long. The first step is to break down  $R$  into context-independent sentences. Then, we decompose each of the sentences into a set of atomic statements, as illustrated on the left side of Figure 2, with no information lost or distorted in this process (Fan et al., 2020; Wright et al., 2022; Chen et al., 2022). Decomposed statements should be checkable independently without preceding and following context.<sup>1</sup> See Appendix C.5 for detailed prompt.

**(2) Decontextualise** Sentences in a response might be context-dependent, with discourse and coreference relations existing between statements (Choi et al., 2021). For example, it is invalid to check the statement *It does not have a*

*king* before replacing “It” with “Canada” or “Constitutional monarchy” (see Figure 2). In addition to coreference relation, for the sentence S2, it is not reasonable to check the claim *Queen Elizabeth II is also the queen of 15 other Commonwealth realms*. Instead, the claim should be reframed to *Queen Elizabeth II is the queen of 16 Commonwealth realms (including Canada)* or *Queen Elizabeth II was the queen of 15 Commonwealth realms other than Canada*.

The concept of “context-independent” is straightforward, while the notion of “atomic” is subjective and ambiguous. This poses challenges: how to determine the granularity of *an atomic claim*? when and where to break down a response? For example, S1: *Canada is a constitutional monarchy, and as such, it does not have a king*, can be fact-checked as one statement, or be decomposed into two claims: *Canada is a constitutional monarchy* and *Canada does not have a king*. In our work, we first split a document into sentences, and then from sentence to claims, with each claim containing only one property or fact to verdict.

**(3) Identify Checkworthy Claims** Not all statements in a response require fact-checking, such as subjective opinions and actual commonsense, as obvious as *sun rises from the east*. Each statement in this framework will be identified whether it is checkworthy or not. However, checkworthiness is subjective to determine. Hassan et al. (2015) defined checkworthy claims as those for which the general public would be interested in knowing the truth. In the context of fact-checking LLMs outputs, we assume users who ask LLMs questions are interested in knowing the truth of all factual claims in the corresponding answer.

We specifically classify a statement into four categories: factual claim, opinion, not a claim (e.g. questions, exclamations, imperatives), and others (e.g. *As a language model, I cannot...*). Afterwards, a set of checkworthy factual claims needs to be verified by retrieving and collecting evidence.

Note that for checkworthiness, we not only take account of objective fact against subjective judgement, other aspects such as the role (importance) of the claim to the response is also a crucial criterion for its checkworthiness. For example, the sentence S1 needs more attention than the last sentence S4 in Figure 2. We label the importance level of both decomposed sentences and claims by labels: *most important*, *intermediate*, and *less important*.

<sup>1</sup>Statements are assumed to be checkable if relevant documents exist in publicly-available data sources.

179	<b>(4) Retrieve and Collect Evidence</b>	Evidence can be retrieved by a search engine like Google, or deep retrieval from a closed document collection such as Wikipedia, or using the parametric knowledge of a LLM. Search queries can be questions covering different aspects of the claim, entities in the claim, or even the claim itself (Dai et al., 2022). We used Google search, considering the quality and coverage.	229
180			230
181			231
182			232
183			233
184			234
185			235
186			236
187			237
188	<b>(5) Identify Stance of Evidence</b>	With retrieved evidence for a claim, how to identify the stance of the evidence against the claim. RARR (Gao et al., 2022) achieved this by assessing whether answers depending on the evidence and the claim are the same or not, given a query. If they are same, then the evidence supports the claim, otherwise refutes it. Previous work also employs natural language inference (NLI) model to classify whether the claim can be entailed by evidence, or is controversial against evidence, or is irrelevant (Martín et al., 2022; Jullien et al., 2023).	238
189			239
190			240
191			241
192			242
193			243
194			244
195			245
196			246
197			247
198			248
199			249
200		However, some evidence may neither refute nor fully support a claim. This mainly results from the fact that it is always possible that the evidence supports part of the claim. For example, for the claim <i>Elon Musk is the founder, CEO, and chief engineer of SpaceX</i> , evidence <i>Elon Musk is the CEO of SpaceX, Tesla, and Twitter</i> falls into this category. The evidence supports the factual statement of <i>Elon Musk is the CEO of SpaceX</i> , but it does not provide information regarding whether <i>Elon Musk is the founder and chief engineer of SpaceX</i> .	250
201			251
202			252
203			253
204			254
205			255
206			256
207			257
208			258
209			259
210			260
211		Therefore, we incorporate <i>partially support</i> in addition to <i>support</i> , <i>refute</i> and <i>irrelevant</i> (Kamoi et al., 2023). Concretely, <i>support</i> means that the evidence entails the claim. <i>Partial support</i> refers to the scenario where part of the information presented in a claim appears in the evidence. <i>Refute</i> means that the evidence mentions the same event as the claim, but a clear opposite fact contrasting to a part or the whole facts presented in a claim. <i>Irrelevant</i> refers to the situation that the evidence does not mention anything about the fact described in the claim, such that it neither supports nor refutes the claim.	261
212			262
213			263
214			264
215			265
216			266
217			267
218			268
219			269
220			270
221			271
222			272
223			273
224		Sometimes, it is ambiguous to distinguish between <i>refute</i> and <i>irrelevant</i> . We highlight that the evidence shows a clear opposite fact under <i>refute</i> stance, while the evidence does not include relevant facts mentioned in the claim under <i>irrelevant</i> .	274
225			275
226			276
227			277
228			278
	<b>(6) Determine Correction</b>	Given a claim, there will be more than one piece of related evidence. Most of the time, they hold consistent stances except for irrelevance, but sometimes, some support, some partially support while some refute (see Figure 8). How to aggregate conflicting stances and further decide how to make corrections to the claim is an open question. In practice, when evidence paragraphs conflict with each other, we will take the reliability of the evidence source into consideration and, meanwhile, retrieve extra information to judge which one is more dependable.	279
		A label often used is <i>not-enough-evidence</i> if there is insufficient information to make the veracity prediction, e.g., all retrieved evidence is irrelevant or intricate contradictory evidence (Atanasova et al., 2022). So we set three labels in terms of factuality: true, false, and not-enough-evidence.	280
	<b>(7) Edit Claims</b>	With the principle that revised claims should preserve the text’s original intent and style. Without adding or changing unnecessary additional information, we include edit operations: delete the whole claim, replace X with Y, and delete X, where X and Y are meta information in a claim. For other related works, see (Thorne and Vlachos, 2021; Schuster et al., 2021).	281
			282
	<b>(8) Revise Response</b>	After revision, we merge statements in the original order, including non-checkworthy statements, true claims, and revised claims. Finally, we delete reduplicative content if applicable, outputting a correct and fluent response.	283
		<b>Discussion:</b> our annotation framework splits the fact-checking pipeline into eight steps, more fine-grained than existing systems. This intends to incorporate all subtasks and attributes relevant to automated fact-checkers, so that the comprehensive labels can cover evaluations of a wide array of unit modules within fact-checking systems. Practical implementation of fact-checkers can merge some steps, e.g., decomposition and decontextualisation into one, and evaluate the results of context-independent claims.	284
		FELM (Chen et al., 2023) annotated sentence-level <i>true or false</i> labels without correction and showed that factual error detection performance tends to be improved when utilising claim-based segmentation methods compared with sentences. Therefore, we annotate a claim-based document-level fact-checking dataset in Section 3.	285

### 3 Dataset Construction

We annotate a dataset serving for a benchmark evaluating the effectiveness of approaches for LLM fact-checking subtasks or the whole pipeline, and few-shot demonstration examples.

#### 3.1 Data Collection

*What kind of LLM generations are we most concerned about?* In the context of detecting and correcting factual errors, we focus on generations in which the majority of statements are objective facts rather than subjective opinions whose veracity is not checkable. Additionally, we are more interested in questions where LLMs are prone to hallucinate or produce factual errors in responses. The whole annotation process is extremely time-consuming, about 15-30 minutes for an instance even if with the annotation tool to ease the procedure. This requests us to sample examples that highly satisfy two criteria — **fact-intensive** and **factually-false**.

**Sources** We start from hallucinations posted by ChatGPT<sup>2</sup> users on Twitter and further collect data by in-house brainstorming with preliminary verification, resulting in 45 examples. We further employ data from dolly-15k, which is brainstormed by thousands of *Databricks* employees with eight categories. 563 examples from closed QA and 528 from open QA are sampled by ChatGPT response length and the semantic similarity with gold answers, with 1,136 (question, response) pairs in total (see more in Appendix C.1).

**Data Selection** We select factually-false responses by estimating the percentage of incorrect claims in a response with four steps.

*Sentence and claim split:* given the whole response as the context and the first sentence (initialized by NLTK tokenizer), we instruct ChatGPT by three demonstration examples to guide it first breaking the input sentence into independent atomic claims, and then continue the decomposition of the next sentence until the end of the response (see the prompt in Appendix C.5). This is different from approaches in Kamoi et al. (2023); Min et al. (2023) despite all prompting LLMs.

*Evidence collection:* given a claim, we first prompt ChatGPT to generate search queries, and then the Google search engine is used to get relevant web pages. Retrieved documents are split into passages by sliding windows, and a re-ranker

combining lexical and semantic similarity is used to identify the most relevant passages for the given query, in which Sentence-BERT (Reimers and Gurevych, 2019) serves for semantic embeddings. We aggregate evidence for all queries and select the top-5 evidences per atomic claim, similar to approaches presented by Laban et al. (2022) and Schuster et al. (2022).

*FActScore calculation:* FActScore is an automatic factuality metric, measuring the percentage of atomic claims supported by knowledge sources in a generation (Min et al., 2023). We use the gathered evidences as input, along with the claim, and an instruction-tuned LLM as the verifier to verdict.

*Example selection:* we keep all 45 pairs from the first source and dolly examples whose FActScore are less than 0.2, resulting in 33 closed QA pairs and 37 open questions, in total of 115 examples (see FActScore distribution in Figure 3). We remove a similar question, and four questions where the LLM did not provide helpful answers due to its inherent disability to access real-time data, eventually annotating 110 examples. For more details regarding data selection, see Appendix C.2.

#### 3.2 Annotation

Studies show that annotating a LLM factuality dataset is a highly challenging and time-consuming task (Chen et al., 2023; Li et al., 2023).

**Preliminary Trial Take-away** Manually annotating the whole process and typing results into a *json* file exposes three major difficulties: (1) retrieving supportive or contradictory evidence takes time and demands the annotator’s strong skills in searching for relevant and filtering out unreliable information, especially for non-common knowledge (e.g. *most popular bottled water brand in Israel*); (2) lengthy responses require good reading comprehension ability and patience; (3) certain domains such as genes and astronomy require domain knowledge, otherwise it is hard to search for valid evidence and determine whether it is true or false.

Taking the factors mentioned above into consideration, we design and build an annotation tool to support the efficient construction of the LLM factuality benchmark. Annotators can edit and assign labels based on intermediate outputs of automatic methods, click buttons instead of typing to copy-paste text, select, and download annotated results.<sup>3</sup>

<sup>3</sup>Without the annotation tool, on average, it takes ~1.5 hours to annotate a 120-word response with about five bullet

<sup>2</sup>ChatGPT refers to GPT-3.5 Turbo in this work.

**Annotation Tool** includes all subtasks and supports semi-auto annotation by incorporating the results of automatic methods, such as automatically-decomposed claims and automatically-retrieved evidence, to ease the annotation process and reduce the workload (see interfaces in Appendix H).

We perform the whole annotation in three steps: (1) decomposition, decontextualisation, and checkworthiness detection; (2) evidence stance identification and claim correction; (3) claim merge, deduplication, and response revision.

Between steps (1) and (2), we incorporate an automatic evidence retrieval system to provide annotators with a set of the most relevant snippets of documents and URLs, as evidence for each checkworthy atomic claim, generally five pieces. They are selected and ranked by semantic relevance degree against the claim throughout a large number of documents, similar to the evidence collection in the data selection above. Then, annotators determine the stance of each piece of evidence. With evidence from the automatic system, if annotators still cannot determine the factuality of a claim, they are requested to collect relevant evidence manually. This to some extent alleviates the system bias.

**Quality Control** To guarantee the annotation quality, instead of employing crowd-sourcing annotators, we perform an in-house labelling by ten annotators who are Master’s and PhD students, postdocs, and professors and are familiar with fact-checking.

Two annotators as a group are responsible for 22 model responses. For each step, annotators first independently finish individual annotations, and then consolidate their results with the group partner. In consolidation, partners discuss their disagreements until reaching a consensus. For cases where it is hard to reach an agreement even with the participation of the third rater, we discard it. Three steps are rigorously conducted serially. Annotators start the second step only after they finish the consolidation of the first step. Collecting evidence and judging stances is the most time- and patience-consuming step. To ensure quality, we incorporate the third rater when consolidating the second-step annotations in case of unintentional mistakes. For more details regarding data annota-

points, and more than 4 hours to annotate a 400-word response with ten bullet points, especially when the annotator is not familiar with details of an event (e.g., *What are some details that are public about the 2021 Capitol Hill riots*). With the tool, it takes 15-30 minutes to label a 50-150 words document.

	document	sent	cw_sent	claim	cw_claim	evid
size	94	311	277	678	661	3,305

Table 1: Statistics of the dataset. cw\_sent=checkworthy sentences, cw\_claim=checkworthy claims, evid=the total pieces of evidence, five for each cw\_claim.

Source	In-house	Closed-QA	Open-QA	All
Collected	45	33	35	115
Annotated	39	30	25	94
False	38	16	8	61

Table 2: False responses over three question sources.

tion, see Appendix C.3, and annotation guidelines are presented in the Appendix D.

### 3.3 Data Analysis

During annotation, we remove another 16 responses (see details in Appendix C.4), resulting in 94 instances. Statistics are shown in Table 1.

**Statistics** 277 sentences contain factual statements among 311. There are 678 atomic claims, where 661 claims are checkworthy, 16 are opinions and one is *not-a-claim*. For each checkworthy claim, five pieces of evidence are collected, resulting in 3,305 (claim, evidence, stance) triplets.

**How many examples are factually correct?** 61 examples contain factual errors, and 31 are factually correct, 2 without checkworthy claims. Amongst, 53 examples contain false claims, and 19 examples contain claims in which annotators cannot verify the statement due to insufficient evidence despite the manual search. Generally, one example contains 0-5 false claims. There are six examples with >5 incorrect claims. 16 sentences among 331 are deleted. 12 are total hallucinations, e.g., *Trump was the second black president*. 4 sentences are removed due to lacking enough evidence to support its factual correctness.

Table 2 shows that more incorrect responses appear in in-house collected questions, followed by dolly closed questions that require knowledge to obtain a unique correct answer. Fewer errors occur in dolly open questions, in which correct answers are not unique, e.g., *How do you play an E major chord on a guitar?* It has diverse correct answers requiring more general knowledge.

**Claims** Of 678 claims, 419 and 227 are labelled as the most and intermediate important claims, and only 32 fall into *not-important*, indicating that users

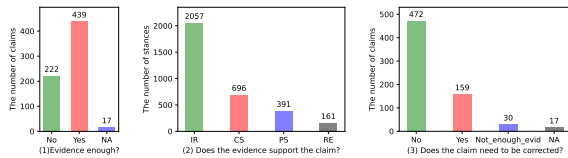


Figure 1: **Claim analysis:** (1) whether raters can determine the factuality of a claim depending on the automatically-collected evidence (*Yes/No*); (2) does the evidence support the claim (*CP*: completely support, *PS*: partially support, *RE*: refute, *IR*: irrelevant); (3) does the claim need to be corrected. NA (17) refers to 16 opinion-claims + 1 *not-a-claim*.

concern with almost the whole response given their importance. We analyse annotations of 661 check-worthy claims from two perspectives.

**Can raters determine the factuality of a claim depending on the automatically-collected evidence?** For 439 claims, annotators can determine *true or false* with automatic evidence, while 222 claims (one-third) need further manual retrieval to make judgements. Among the 222 claims, 125 true claims fall into domain knowledge and information that is less known by the external people given a country, region, company, or an individual. The other half are either factually-incorrect claims (76) or undetermined claims without sufficient evidence despite manual retrieval (21), shown in Figure 6.

This suggests the ineffectiveness of the automatic evidence retrieval methods on collecting rare knowledge and evidence conditioned on false premises (claims). However, it may also reply that not all facts have been presented by textual descriptions directly. Some facts are unknown by the public, and some require connecting and reasoning knowledge from multiple sources, e.g., *did Aristotle use a laptop?* (Geva et al., 2021).

**How many claims need to be corrected?** In Figure 1, about a quarter (159/661) of claims are factually incorrect and need to be corrected. 30 claims are undetermined due to inadequate related information and knowledge even with manual retrieval. It is hard to obtain reliable related information about these cases by searching publicly-available sources. They involve expert-level knowledge (e.g., gene, water memory, black hole) and private details of an individual, organisation, or country (personal awards and preferences, revenue of a company), which are only known by a small group of people, such as domain experts or internal individuals who are familiar with the event.

**Original vs. revised responses** We quantify the difference between the original responses and the human-revised responses over the 61 false responses, showing that the normalised edit distance is 0.354, word overlap is 0.715, while semantically, BERTScore-F1 is 0.955 and cosine similarity based on SimCSE (Roberta-large) is 0.912. This implies that the core content of LLM answers is mostly correct, but minor factual mistakes are easily made by LLMs in detail, leading to high semantic similarity but multiple lexical edits in small errors.

**Summary** The dataset consists of 94 ChatGPT (prompt, response) pairs. Each sample has detailed labels concerning the verification: elements of de-contextualised sentences, atomic claims, the importance degree of the sentence, claim to the response, five pieces of evidence for a claim, the relationship between a claim and evidence, factual label (*true or false*) and revised version of claims, sentences, and the response.

## 4 Unit Test for Fact-checkers

In this section, we compare the results of automatic methods that are commonly used in current fact-checking systems (e.g., RARR, FActScore, FacTool) for subtasks with human annotations. We first compare the automatic and human-annotated decomposition of atomic claim, and then evaluate five subtasks: (1) identify whether the sentence contains a factual statement; (2) detect the check-worthiness of a claim by categories of *factual*, *opinion*, *not a claim* and *other*; (3) judge the stance of a given evidence against a claim, whether it *supports*, *partially supports*, *refutes* or is *irrelevant* to the claim; (4) determine whether a claim is factually true or false, give a claim without “gold evidence”, if false, revise it into a correct one; (5) edit a list of originally-true or revised claims into a new response, given the original response, to correct the factual errors while preserving the linguistic features and style of the original.

Other steps are excluded because they are either relatively easy for current techniques (e.g., splitting a document into sentences), or results of automatic approaches have been compared against human annotations in data analysis, such as the relevance or quality of the automatically-retrieved evidence.

### 4.1 Automatic vs. Manual Decomposition

For 66/277 checkworthy sentences, the number of decomposed atomic claims is different between au-

544 automatic breaking-down by ChatGPT and manual  
 545 annotations. Amongst, more claims decomposed  
 546 by the automatic method than humans for 48 sen-  
 547 tences, and fewer claims for 18 sentences. This  
 548 exhibits that human annotators add extra claims  
 549 to only a small number of sentences. In most  
 550 cases, the automatic approach decomposes sen-  
 551 tences into an equal number of claims or even more  
 552 fine-grained than humans.

553 For the rest 211 sentences, human and ChatGPT  
 554 decompose the sentence into the same number of  
 555 claims, 521 claims are involved. This enables pair-  
 556 wise claim comparison between the human anno-  
 557 tation and automatic method. We calculate the  
 558 lexical similarity and distance: normalized edit dis-  
 559 tance=0.11, n-gram distance=0.11, and word over-  
 560 lap=0.88 demonstrating high agreement between  
 561 human annotation and ChatGPT decomposition.

## 562 4.2 Checkworthiness

563 We apply ChatGPT to identify if decomposed sen-  
 564 tences and claims are verifiable objective facts or  
 565 statements containing personal opinions.

566 **Subtask 1 and 2** We identify whether a sentence  
 567 contains a factual statement by a binary label (*yes*  
 568 or *no*) and whether a claim is checkworthy by  
 569 four labels (*factual claim*, *opinion*, *not-a-claim* and  
 570 *other*). The accuracy for subtask 1 by majority  
 571 guess (always checkworthy) will be  $277/311=0.891$   
 572 and the baseline for subtask 2: claim classification  
 573 is  $661/678 = 0.975$ . They are superior to using the  
 574 prompt based on ChatGPT: the accuracy is 0.814  
 575 and 0.932 respectively. However, this is mainly  
 576 attributed to the extremely-unbalanced data. Practi-  
 577 cally, our aim is to make distinctions. It’s critical to  
 578 consider recall: ChatGPT is much better than the  
 579 majority guess (see Table 3).

580 The confusion matrix in Figure 9 shows that  
 581 46 checkworthy sentences are identified as non-  
 582 checkworthy, accounting for 15%. Factual claims  
 583 could be recognized into any of the four labels, and  
 584 real opinions tend to be identified as factual claims,  
 585 even more than the opinion.

## 586 4.3 Verification

587 **Subtask 3** classifies whether the evidence fully  
 588 supports, partly supports, refutes, or is irrelevant  
 589 to the claim, given a (*claim*, *evidence*) pair. We  
 590 use zero-shot prompting based on ChatGPT and  
 591 LLaMA2 (7B), and find that LLaMA2 barely pre-  
 592 dicts *partial support* and always misclassifies as

Task	Method	Acc	Prec	Recall	F1-macro
1	Always-checkworthy	0.891	0.445	0.500	0.471
1	ChatGPT	0.814	0.637	0.740	0.660
2	Always-checkworthy	0.975	0.325	0.333	0.329
2	ChatGPT	0.932	0.314	0.534	0.319

Table 3: **Checkworthiness** detection by majority guess: Always-checkworthy vs. ChatGPT zero-shot prompt. *average*=“macro” is used in precision (Pred), recall and F1 calculation.

Method	Acc	Prec	Recall	F1-macro
<b>Four-label space</b>				
Random guess	0.255	0.258	0.264	0.215
LLaMA2-zeroshot	0.202	0.324	0.280	0.155
ChatGPT-zeroshot	0.365	0.402	0.439	0.332
<b>Three-label space</b>				
ChatGPT-zeroshot	0.567	0.506	0.588	0.483
LLaMA2-zeroshot	0.401	0.407	0.384	0.299
RoBERTa-large-mnli	<b>0.607</b>	<b>0.536</b>	<b>0.609</b>	<b>0.512</b>

Table 4: **Stance** detection by ChatGPT and LLaMA2 zero-shot prompt. Three-label space merges complete and partial support into one.

593 *irrelevant*, so we merge *complete support* and *par-*  
 594 *tial support* into a single label *support*. As results  
 595 shown in Table 4, three labels are easier for models  
 596 to predict with higher accuracy, but its absolute F1-  
 597 score is still less than 0.5, revealing the challenges  
 598 to distinguish the relationship between claim and  
 599 evidence by LLM in-context learning, especially  
 600 on the label of *refute*. Both LLaMA2 and ChatGPT  
 601 show around-0.1 F1 (see Table 10). We further use  
 602 a fine-tuned NLI model (*RoBERTa-large-mnli*) to  
 603 predict the stance, where entailment, contradiction,  
 604 and neutral correspond to labels of support, refute,  
 605 and irrelevant respectively. It performs better than  
 606 zero-shot ChatGPT, mainly being superior to pre-  
 607 dicting the label of *support*.

608 **Subtask 4** determines whether the claim is true  
 609 or false by leveraging the evidences retrieved from  
 610 external knowledge sources. We evaluate the ver-  
 611 ification methods used in FActScore (Min et al.,  
 612 2023) and FacTool (Chern et al., 2023), with vary-  
 613 ing evidence sources: Wikipedia (September 2023  
 614 dump) and web articles searched by Google. Com-  
 615 mercial verifier Perplexity.ai and the verifier im-  
 616 plemented with Google search + GPT-4 based on  
 617 the solution in this work (Factcheck-GPT) are also  
 618 evaluated.

619 Table 5 shows that false claims tend to be iden-  
 620 tified less accurately than true claims across all  
 621 approaches, implying that it is more difficult to

Verifier	Source	Label = True			Label = False		
		Prec	Recall	F1	Prec	Recall	F1
Random	NA	0.79	0.43	0.56	0.18	0.52	0.27
Always True	NA	0.81	1.00	0.88	0.00	0.00	0.00
Always False	NA	0.00	0.00	0.00	0.19	1.00	0.33
Inst-LLAMA	Wiki	0.87	0.74	0.80	0.34	0.56	0.42
Inst-LLAMA	Web	0.88	0.80	0.84	0.40	0.56	0.47
GPT-3.5-Turbo	Wiki	0.87	0.67	0.76	0.31	0.60	0.41
GPT-3.5-Turbo	Web	0.89	0.74	0.81	0.37	0.62	0.46
Perplexity.ai	Web	<b>0.93</b>	<b>0.73</b>	<b>0.83</b>	0.40	0.76	0.53
Factcheck-GPT	Web	0.90	0.71	0.79	<b>0.52</b>	<b>0.80</b>	<b>0.63</b>

Table 5: **Verification results** on our benchmark: judge whether a claim is factually true or false with external knowledge (Wikipedia or Web articles) as evidence.

detect factual errors than the correct statements. Factcheck-GPT performs the best on false claims with F1=0.63, and then Perplexity.ai by 0.53, followed by Instruction-LLaMA with web articles as evidence (F1=0.47/0.84), and verifying using GPT-3.5-Turbo exhibits slight declines. This reveals that current mainstreaming SOTA fact-checkers still have large room to improve on verification, particularly on false claims. Performance using Wikipedia as the source is inferior to using web articles, this is largely limited by the knowledge coverage of Wikipedia, esp. on open-domain benchmarks.

#### 4.4 Revision

**Subtask 5** Given the original false response, a list of revised true claims, ChatGPT and GPT-4 are prompted to revise the responses with/without the question, resulting in four revised responses.

Which revised response is better? We evaluate by human and the intrinsic metrics. BERTScore measures semantic preservation between gold reference answers and the edit-distance measures style preservation between original responses.

In human evaluation, we use the criteria: whether the revised response (1) contain factual errors? (2) keep the style feature of the original response as much as possible? (3) is it natural, coherent, and smooth as an answer? Criteria (1) is the most important, followed by (2) and (3). For instance, only *A* and *B* are factually correct, while *A* preserves more of the original response, thus *A* is better. If some responses are totally the same, raters can choose more than one. We collect 66 preference labels for 61 examples.

In case of personal preference bias from one or two raters, six raters are invited to choose their preferred response and provide a brief reason. We also shuffled four revisions and show by “revision\_x” (x=0,1,2,3), masking the real setting name to avoid possible inherent biases.

Prompt	model	Edit-dis↓	WO↑	BS-F1↑	STS↑	Human
no-ques	ChatGPT	<b>0.207</b>	<b>0.864</b>	0.953	0.937	10
no-ques	GPT-4	0.275	0.789	0.954	0.931	28
with-ques	ChatGPT	0.222	0.853	<b>0.956</b>	<b>0.941</b>	13
with-ques	GPT-4	0.286	0.776	0.953	0.935	15

Table 6: **Revision evaluation** by intrinsic metrics and human (how many responses are preferred). Edit distance (**Edit-dis**) and word overlap (**WO**) between revised and the original responses. BERTScore (**BS-F1**) and semantic textual similarity (**STS**) based on SimCSE between the revised responses and human annotations.

In Table 6, intrinsic metric results show that responses revised by ChatGPT (GPT-3.5-turbo) are better than GPT-4, which is against our experience and observation (see examples in Appendix G). Human assessment exhibits that 43 GPT-4 responses are preferred by raters and 23 from ChatGPT. Human is more satisfied with revisions without questions 38 vs. 28. This somewhat reflects the ineffectiveness of intrinsic evaluation metrics.

**Take-Away** ChatGPT shows promising results in atomic-claim decomposition, but low F1-score in checkworthiness detection. Also, verification remains challenging, especially when identifying false claims, even if it involves harnessing external knowledge. GPT-4 can generate sounding revised responses based on true statements. It’s still an open-question in terms of how to evaluate the quality of revised responses by intrinsic metrics.

## 5 Conclusion and Future Work

We proposed a fine-grained annotation framework and constructed Factcheck-Bench, a benchmark to evaluate automatic fact-checkers of LLM output. The benchmark contains 678 open-domain claims generated by LLMs, involving annotations of eight subtasks for detecting and correcting the factual errors in long documents. Human annotations show that LLMs are prone to make factual errors in expert-level knowledge and exclusive details known by a small group of people. Experiments show that current verifiers are struggling to identify open-domain false claims with the best F1=0.63 even if using external knowledge. Additionally, intrinsic measures based on edit distance and semantic similarity are ineffective for evaluating the edited responses against true evidence and the original response, misaligning with human preferences. We plan to explore this in future work.



## 698 Limitations

699 Three major limitations are identified in this work:

700 **Small-scale dataset** It consists of only 94 (ques- 745  
701 tion, response) pairs, we plan to scale up the dataset 746  
702 in English, Chinese, and Arabic in future work. It 747  
703 is worthwhile to note that our dataset contains fine- 748  
704 grained annotations of high quality for eight sub- 749  
705 tasks. Moreover, due to high cost (i.e., over 30-50 750  
706 USD on average to evaluate 100 LLM responses 751  
707 depending on its length), developers generally evalu- 752  
708 ate on less than 100 examples in the development 753  
709 iterations, sometimes even 10 examples to save 754  
710 costs. Practically, our dataset is enough to be used 755  
711 as a benchmark for the preliminary evaluation of 756  
712 automatic fact-checking systems. 757

713 **Inter-claim dependencies** This reflects at three 758  
714 challenges. First, current approaches including our 759  
715 solution are unable to check the overall logical cor- 760  
716 rectness of a procedure, such as how to cook, and 761  
717 whether some steps are out of order. Second, if 762  
718 the first claim is invalidated, maybe the entire text 763  
719 needs to be deleted. Third, it is hard to decontextu- 764  
720 alize implicit claims, e.g., “other 15 realms”, which 765  
721 means there are 16 realms. 766

722 **Quality of evidence** More than half of automati- 767  
723 cally retrieved evidences are irrelevant. Improving 768  
724 the relevance of retrieved evidence is critical to the 769  
725 accuracy of fact-checking. 770

## 726 Ethics and Broader Impact

727 We identify two major risks of the framework and 771  
728 benchmark: 772

729 **Biases:** The automatic atomic-claim decomposi- 773  
730 tion and evidence retrieval systems incorporated in 774  
731 the fact-checking annotations may introduce biases, 775  
732 which can affect the annotation results. 776

733 Besides, the dataset does not cover all types of 777  
734 claims. Limited scope and coverage may be more 778  
735 effective in certain domains, possibly leading to 779  
736 inaccurate or unfair assessments in certain domains 780  
737 for automatic fact-checkers. The responses gener- 781  
738 ated by LLMs might also inherit some biases 782  
739 present in the involved LLMs. 783

740 **The cost of making an error:** The goal of fact- 784  
741 checking is to improve the reliability of the LLM 785  
742 outputs, If post-hoc fact-checking methods under 786  
743 this framework always make errors, practitioners 787  
744 may lose faith in the accuracy of the fact-checking 788

745 results, which can affect efforts to maintain public 746  
747 trust in fact-checking systems. 748

749 **Broader impact:** The proposed framework is not 750  
751 limited to checking the output of LLMs; it is ap- 752  
753 plicable to check any kind of document, including 754  
755 human-written. 756  
757

## 758 References

- 759 Pepa Atanasova, Jakob Grue Simonsen, Christina Li- 760  
761 oma, and Isabelle Augenstein. 2020. [Generating fact 762  
763 checking explanations](#). In *Proceedings of the 58th 764  
765 Annual Meeting of the Association for Computational 766  
767 Linguistics*, pages 7352–7364, Online. Association 768  
769 for Computational Linguistics. 770
- 771 Pepa Atanasova, Jakob Grue Simonsen, Christina Li- 772  
773 oma, and Isabelle Augenstein. 2022. [Fact Checking 774  
775 with Insufficient Evidence](#). *Transactions of the Asso- 776  
777 ciation for Computational Linguistics*, 10:746–763. 778
- 779 Isabelle Augenstein. 2021. [Towards Explainable Fact 780  
781 Checking](#). Dr. Scient. thesis, University of Copen- 782  
783 hagen, Faculty of Science. 784
- 785 Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, 786  
787 Tanmoy Chakraborty, Giovanni Luca Ciampaglia, 788  
789 David Corney, Renee DiResta, Emilio Ferrara, Scott 790  
791 Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo 792  
793 Menczer, Ruben Miguez, Preslav Nakov, Dietram 793  
794 Scheufele, Shivam Sharma, and Giovanni Zagni. 794  
795 2023. [Factuality Challenges in the Era of Large Lan- 795  
796 guage Models](#). 796
- 797 Isabelle Augenstein, Christina Lioma, Dongsheng 797  
798 Wang, Lucas Chaves Lima, Casper Hansen, Chris- 798  
799 tian Hansen, and Jakob Grue Simonsen. 2019. [Multi- 799  
800 tiFC: A real-world multi-domain dataset for evidence- 800  
801 based fact checking of claims](#). In *Proceedings of 801  
802 the 2019 Conference on Empirical Methods in Natu- 802  
803 ral Language Processing and the 9th International 803  
804 Joint Conference on Natural Language Processing 804  
805 (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, 805  
806 China. Association for Computational Linguistics. 806
- 807 Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wen- 807  
808 liang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei 808  
809 Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, 809  
810 and Pascale Fung. 2023. [A multitask, multilingual, 810  
811 multimodal evaluation of chatgpt on reasoning, hal- 811  
812 lucination, and interactivity](#). *CoRR*, abs/2302.04023. 812
- 813 Oscar Barrera, Sergei Guriev, Emeric Henry, and Ekate- 813  
814 rina Zhuravskaya. 2020. [Facts, alternative facts, and 814  
815 fact checking in times of post-truth politics](#). *Journal 815  
816 of public economics*, 182:104123. 816
- 817 Emily M Bender, Timnit Gebru, Angelina McMillan- 817  
818 Major, and Shmargaret Shmitchell. 2021. [On the 818  
819 dangers of stochastic parrots: Can language models 819  
820 be too big?](#) In *Proceedings of the 2021 ACM confer- 820  
821 ence on fairness, accountability, and transparency*, 821  
822 pages 610–623. 822

799	Ali Borji. 2023. <a href="#">A categorical archive of chatgpt failures</a> . <i>CoRR</i> , abs/2302.03494.	<i>Association for Computational Linguistics</i> , 9:346–361.	855
800			856
801	Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. <a href="#">Generating literal and implied sub-questions to fact-check complex claims</a> . In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing</i> , pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.	Guiven. 2023. Llm failure archive (chatgpt and beyond). <a href="https://github.com/giiven95/chatgpt-failures">https://github.com/giiven95/chatgpt-failures</a> .	857
802			858
803			859
804		Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. <a href="#">A survey on automated fact-checking</a> . <i>Transactions of the Association for Computational Linguistics</i> , 10:178–206.	860
805			861
806			862
807			863
808	Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. <a href="#">Felm: Benchmarking factuality evaluation of large language models</a> . <i>arXiv preprint arXiv:2310.00741</i> .	Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. <a href="#">Detecting check-worthy factual claims in presidential debates</a> . In <i>Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015</i> , pages 1835–1838. ACM.	864
809			865
810			866
811			867
812	I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. <a href="#">Factool: Factuality detection in generative AI - A tool augmented framework for multi-task and multi-domain scenarios</a> . <i>CoRR</i> , abs/2307.13528.	Robert Iv, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. 2022. <a href="#">FRUIT: Faithfully reflecting updated information in text</a> . In <i>Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 3670–3686, Seattle, United States. Association for Computational Linguistics.	871
813			872
814			873
815			874
816			875
817			876
818	Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. <a href="#">Decontextualization: Making sentences stand-alone</a> . <i>Transactions of the Association for Computational Linguistics</i> , 9:447–461.	Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. <a href="#">Survey of hallucination in natural language generation</a> . <i>ACM Comput. Surv.</i> , 55(12):248:1–248:38.	877
819			878
820			879
821			880
822			881
823	Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. <a href="#">Promptagator: Few-shot dense retrieval from 8 examples</a> . <i>arXiv preprint arXiv:2209.11755</i> .	Maël Jullien, Marco Valentino, Hannah Frost, Paul O’regan, Donal Landers, and André Freitas. 2023. <a href="#">SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data</a> . In <i>Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)</i> , pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.	882
824			883
825			884
826			885
827			886
828	Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. <a href="#">Chain-of-verification reduces hallucination in large language models</a> . <i>arXiv preprint arXiv:2309.11495</i> .	Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. <a href="#">WiCE: Real-world entailment for claims in Wikipedia</a> . In <i>Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing</i> , pages 7561–7583, Singapore. Association for Computational Linguistics.	887
829			888
830			889
831			890
832			891
833	Thomas Diggelmann, Jordan L. Boyd-Graber, Jan-nis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. <a href="#">CLIMATE-FEVER: A dataset for verification of real-world climate claims</a> . <i>CoRR</i> , abs/2012.00614.	Sayash Kapoor, Peter Henderson, and Arvind Narayanan. 2024. <a href="#">Promises and pitfalls of artificial intelligence for legal applications</a> . <i>arXiv preprint arXiv:2402.01656</i> .	892
834			893
835			894
836			895
837			896
838	Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. <a href="#">Generating fact checking briefs</a> . In <i>Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)</i> , pages 7147–7161, Online. Association for Computational Linguistics.	Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. <a href="#">SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization</a> . <i>Transactions of the Association for Computational Linguistics</i> , 10:163–177.	901
839			902
840			903
841			904
842			905
843			906
844			907
845	Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2022. <a href="#">Attributed text generation via post-hoc research and revision</a> . <i>arXiv preprint arXiv:2210.08726</i> .	Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. <a href="#">Halueval: A large-scale hallucination evaluation benchmark for large language models</a> . <i>CoRR</i> , abs/2305.11747.	908
846			909
847			
848			
849			
850			
851	Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. <a href="#">Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies</a> . <i>Transactions of the</i>		
852			
853			
854			

910	Tianyu Liu, Xin Zheng, Baobao Chang, and Zhifang Sui. 2021. <a href="#">Towards faithfulness in open domain table-to-text generation from an entity-centric view</a> . In <i>Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021</i> , pages 13415–13423. AAAI Press.	<i>Canada, 19-27 August 2021</i> , pages 4551–4558. <a href="#">ij-cai.org</a> .	967 968
920	Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. <a href="#">Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation</a> . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.	Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. <a href="#">Combining fact extraction and verification with neural semantic matching networks</a> . In <i>The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019</i> , pages 6859–6866. AAAI Press.	969 970 971 972 973 974 975 976 977 978
929	Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. <a href="#">Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models</a> . <i>CoRR</i> , abs/2303.08896.	OpenAI. 2023. <a href="#">Gpt-4 technical report</a> . <i>ArXiv</i> , abs/2303.08774.	979 980
933	Alejandro Martín, Javier Huertas-Tato, Álvaro Huertas-García, Guillermo Villar-Rodríguez, and David Camacho. 2022. <a href="#">Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference</a> . <i>Knowledge-Based Systems</i> , 251:109265.	Nicolas Pröllochs. 2022. <a href="#">Community-based fact-checking on twitter’s birdwatch platform</a> . In <i>Proceedings of the Sixteenth International AAAI Conference on Web and Social Media, ICWSM 2022, Atlanta, Georgia, USA, June 6-9, 2022</i> , pages 794–805. AAAI Press.	981 982 983 984 985 986
939	Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. <a href="#">On faithfulness and factuality in abstractive summarization</a> . In <i>Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics</i> , pages 1906–1919, Online. Association for Computational Linguistics.	Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. <a href="#">The curious case of hallucinations in neural machine translation</a> . In <i>Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 1172–1183, Online. Association for Computational Linguistics.	987 988 989 990 991 992 993
945	Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. <a href="#">Factscore: Fine-grained atomic evaluation of factual precision in long form text generation</a> . <i>CoRR</i> , abs/2305.14251.	Nils Reimers and Iryna Gurevych. 2019. <a href="#">Sentence-bert: Sentence embeddings using siamese bert-networks</a> . In <i>Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing</i> . Association for Computational Linguistics.	994 995 996 997 998
951	Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2023. <a href="#">Generating benchmarks for factuality evaluation of language models</a> . <i>CoRR</i> , abs/2307.06908.	Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022. <a href="#">Stretching sentence-pair NLI models to reason over long documents and clusters</a> . In <i>Findings of the Association for Computational Linguistics: EMNLP 2022</i> , pages 394–412, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.	999 1000 1001 1002 1003 1004 1005
956	Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin T. Vechev. 2023. <a href="#">Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation</a> . <i>CoRR</i> , abs/2305.15852.	Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. <a href="#">Get your vitamin C! robust fact verification with contrastive evidence</a> . In <i>Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 624–643, Online. Association for Computational Linguistics.	1006 1007 1008 1009 1010 1011 1012
960	Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. <a href="#">Automated fact-checking for assisting human fact-checkers</a> . In <i>Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal,</i>	James Thorne and Andreas Vlachos. 2021. <a href="#">Evidence-based factual error correction</a> . In <i>Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)</i> , pages 3298–3309, Online. Association for Computational Linguistics.	1013 1014 1015 1016 1017 1018 1019
966		James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. <a href="#">FEVER: a large-scale dataset for fact extraction</a>	1020 1021 1022

- 1023 and VERification. In *Proceedings of the 2018*  
1024 *Conference of the North American Chapter of*  
1025 *the Association for Computational Linguistics:*  
1026 *Human Language Technologies, Volume 1 (Long*  
1027 *Papers)*, pages 809–819, New Orleans, Louisiana.  
1028 Association for Computational Linguistics.
- 1029 Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry  
1030 Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny  
1031 Zhou, Quoc Le, et al. 2023. [Freshllms: Refreshing](#)  
1032 [large language models with search engine augmenta-](#)  
1033 [tion](#). *arXiv preprint arXiv:2310.03214*.
- 1034 David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu  
1035 Wang, Madeleine van Zuylen, Arman Cohan, and  
1036 Hannaneh Hajishirzi. 2020. [Fact or fiction: Verifying](#)  
1037 [scientific claims](#). In *Proceedings of the 2020 Con-*  
1038 *ference on Empirical Methods in Natural Language*  
1039 *Processing (EMNLP)*, pages 7534–7550, Online. As-  
1040 sociation for Computational Linguistics.
- 1041 Yuxia Wang, Minghan Wang, Muhammad Arslan Man-  
1042 zoor, Georgi Georgiev, Rocktim Jyoti Das, and  
1043 Preslav Nakov. 2024. [Factuality of large language](#)  
1044 [models in the year 2024](#). *CoRR*, abs/2402.02420.
- 1045 Laura Weidinger, John Mellor, Maribeth Rauh, Conor  
1046 Griffin, Jonathan Uesato, Po-Sen Huang, Myra  
1047 Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh,  
1048 et al. 2021. [Ethical and social risks of harm from](#)  
1049 [language models](#). *arXiv preprint arXiv:2112.04359*.
- 1050 Dustin Wright and Isabelle Augenstein. 2020. [Claim](#)  
1051 [Check-Worthiness Detection as Positive Unlabelled](#)  
1052 [Learning](#). In *Findings of the Association for Compu-*  
1053 *tational Linguistics: EMNLP 2020*, pages 476–488,  
1054 Online. Association for Computational Linguistics.
- 1055 Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl,  
1056 Arman Cohan, Isabelle Augenstein, and Lucy Lu  
1057 Wang. 2022. [Generating scientific claims for zero-](#)  
1058 [shot scientific fact checking](#). In *Proceedings of the*  
1059 *60th Annual Meeting of the Association for Computa-*  
1060 *tional Linguistics (Volume 1: Long Papers)*, pages  
1061 2448–2460, Dublin, Ireland. Association for Compu-  
1062 tational Linguistics.
- 1063 Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu,  
1064 Xipeng Qiu, and Xuanjing Huang. 2023. [Do large](#)  
1065 [language models know what they don’t know?](#) In  
1066 *Findings of the Association for Computational Lin-*  
1067 *guistics: ACL 2023*, pages 8653–8665, Toronto,  
1068 Canada. Association for Computational Linguistics.
- 1069 Muru Zhang, Ofir Press, William Merrill, Alisa Liu,  
1070 and Noah A. Smith. 2023a. [How language model](#)  
1071 [hallucinations can snowball](#). *CoRR*, abs/2305.13534.
- 1072 Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu,  
1073 Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang,  
1074 Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei  
1075 Bi, Freda Shi, and Shuming Shi. 2023b. [Siren’s song](#)  
1076 [in the AI ocean: A survey on hallucination in large](#)  
1077 [language models](#). *CoRR*, abs/2309.01219.

A Pipeline Overview

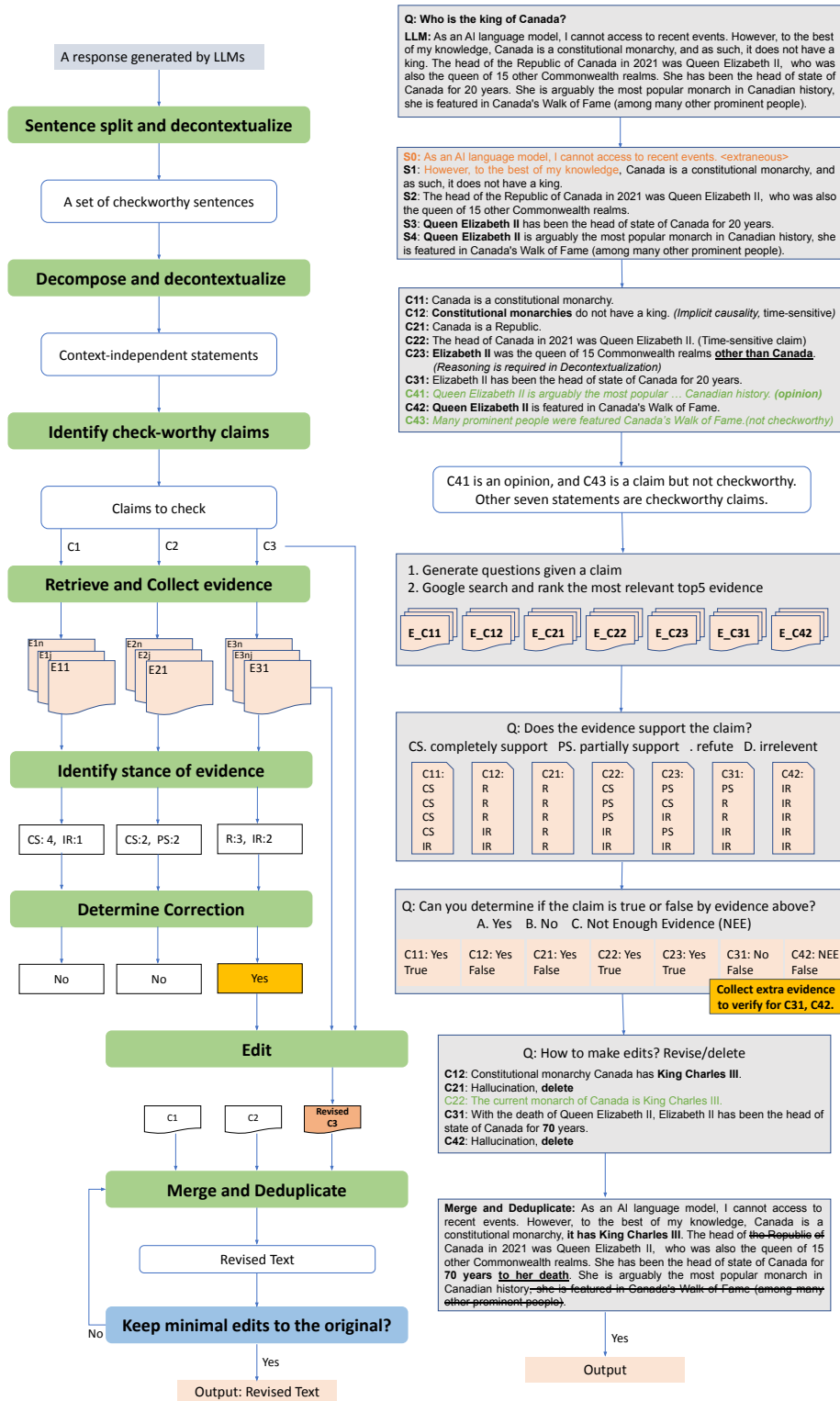


Figure 2: **Left:** Fact-checking pipeline for a response generated by LLMs. **Right:** An example workflow.

## B Fact-checking Background

### B.1 What is Fact-checking?

Fact-checking is the task of assessing whether claims made in writing are manipulated or true. This is typically broken down into the stages of claim detection, evidence retrieval, verdict prediction, and optionally justification prediction (Guo et al., 2022; Augenstein, 2021).

Claim detection is to identify claims that require verification, which commonly relies on the concept of check-worthiness. In the context of human-written documents, checkworthy claims are regarded as those for which the general public would be interested in knowing the truth (Hassan et al., 2015; Wright and Augenstein, 2020). However, this may not be adaptable to machine-generated texts. Plausible hallucinations of LLMs make it difficult for general individuals to distinguish whether it is true or false, thus their outputs become less trustworthy than the statements made by humans. Current methods tend to check all factual claims of LLM generations (Chern et al., 2023).

Evidence retrieval aims to find sources supporting or refuting the claim. Claim verification is expected to assess the veracity of the claim and produce justification based on the retrieved evidence. That is, claims are assigned truthfulness labels, and explanations for verdicts are produced. A basic form of justification is to highlight the pieces of evidence used to reach a verdict (Guo et al., 2022).

Method	D	R	Granularity	Knowledge source	Datasets	Task	How_collect
Factcheck-GPT	✓	✓	claim	Google search	✓	Instruction	prompt ChatGPT and human annotation
FacTool (Chern et al., 2023)	✓	✓	article metadata	Google scholar	✓	Generate literature review	prompt ChatGPT
FacTool (Chern et al., 2023)	✓	✓	claim (gold)	Parsed Google search	RoSE/FactPrompts	Summarisation-eval/QA	human annotation: RoSE (Liu et al., 2023)
RARR (Gao et al., 2022)	✓	✓	document	Bing search	NQ,StrategyQA,QReCC	QA	human annotation
CoVe (Dhuliawala et al., 2023)	✓	✓	document	parametric knowledge	CoVe corpus	QA, instruction	human annotation
FELM (Chen et al., 2023)	✓	✗	segment	Google search	✓	Instruction	prompt ChatGPT and human annotate factuality
Self-contradictory (Mündler et al., 2023)	✓	✗	sentence	parametric knowledge	✓	Instruction	prompt ChatGPT.GPT-4 for contradictory sentence
SelfCheckGPT (Manakul et al., 2023)	✓	✗	sentence	parametric knowledge	✓	Generate Wikibio passage	prompt GPT3 and human annotate 3 factual labels
FACTOR (Muhlgay et al., 2023)	✓	✗	sentence	parametric knowledge	✓	Multichoice QA	prompt <i>davinci-003</i> for non-factual completions
HaluEval (Li et al., 2023)	✓	✗	document	parametric knowledge	✓	QA, summarise, dialogue	prompt ChatGPT to generate hallucinated answers
HaluEval (Li et al., 2023)	✓	✗	document	parametric knowledge	✓	Instruction	prompt ChatGPT, human annotate false segments
FActScore (Min et al., 2023)	✓	✗	claim	Wiki Bio Generation	✓	Instruction	prompt ChatGPT to generate biography
FRESHQA (Vu et al., 2023)	✓	✗	facts in answer	parametric knowledge	✓	QA	collect questions with time-changing answers
Snowball (Zhang et al., 2023a)	✓	✗	Yes/No answer	parametric knowledge	✓	QA	human annotation
SelfAware (Yin et al., 2023)	✓	✗	document	reference generations	✓	QA	collect unanswerable questions and prompt ChatGPT

Table 7: **Methods and benchmarks for hallucination Detection (D) and Revision (R)**. FacTool: article metadata is a tuple (paper title, year, authors). CoVe=Chain-of-Verification, CoVe corpus includes four existing datasets: Wikidata, Wiki-category, MultiSpanQA, and biographic. 3 labels in SelfCheckGPT: major/minor inaccurate and accurate. Unanswerable questions: the model should express uncertainty instead of delivering conclusive responses. FRESHQA collect four types of questions: false premise, answers never change, change slowly and fast over time.

### B.2 Conventional Fact-checking

Previous works either focus on hallucinations in task-specific generations with references (to detect whether the generated output contradicts the source content), such as abstractive summarization (Maynez et al., 2020), machine translation (Raunak et al., 2021) and data-to-text generation (Liu et al., 2021), or concentrate on specific topics e.g. Covid-19 (Augenstein et al., 2019), politics (Barrera et al., 2020), climate (Diggelmann et al., 2020), and specific domains such as journalism, news, social media (e.g. Twitter (Pröllochs, 2022)) and Wikipedia (FEVER: Thorne et al. (2018)). In contrast, we set target for text generation tasks without references such as generative question answering and dialogue systems in open domain and open topic, where the source is the world knowledge.

Moreover, most early studies only perform one or two subtasks in the factual error detection and correction, instead of the whole process. For example, many models estimate a label indicating whether the statement is supported or refuted by the evidence, given a (statement, evidence) pair as input (Thorne et al., 2018; Nie et al., 2019; Augenstein et al., 2019; Wadden et al., 2020). To adapt to situations where relevant evidence for a statement is not readily available, some works explored how to automatically retrieve evidence that may help support or refute a statement (Fan et al., 2020; Nakov et al., 2021; Gao et al., 2022).

More recent work has also explored how to correct claims based on retrieved evidence (Thorne and Vlachos, 2021; Schuster et al., 2021; Iv et al., 2022) and how to generate justification/explanation for

verdicts on claims (Atanasova et al., 2020). However, most factual correction used human-authored edits from FEVER (Thorne et al., 2018) as both their training and automatic evaluation data. FEVER’s claims were extracted from Wikipedia. This limits the generalizability of these fact-checking models over generations of LLMs across various tasks and diverse domains.

Our goal is to establish a holistic framework, evaluating systems that automatically detect and correct factual errors end to end for open-domain factual knowledge hallucinations.

### B.3 LLM Fact-checking

The phenomenon that LLMs produce outputs that are seemingly plausible while deviating from the user input, previously generated context, or factual knowledge, is commonly referred to as hallucination (Zhang et al., 2023b). Based on the timing of the LLM life cycle, LLM hallucinations can be addressed during pretraining by automatically selecting reliable data or filtering out noisy data to mitigate hallucinations, in supervised fine-tuning by curating a small volume of high-quality training data, in reinforcement learning from human feedback (RLHF), and during inference by decoding strategies (Zhang et al., 2023b). We focus on approaches applied after inference.

**Methods** For post-processing approaches to alleviating LLM hallucinations, recent studies can be roughly classified into two categories depending on whether they rectify errors: (1) detecting and correcting factual errors for free-form text; and (2) only detecting whether a text contain hallucinations (*Yes* or *No*). Both of them resort to either external knowledge or parametric knowledge to identify and rectify factually-incorrect statements (Gao et al., 2022; Chern et al., 2023; Manakul et al., 2023; Dhuliawala et al., 2023). We used external knowledge retrieved from Google.

Our work puts efforts into facilitating the first category. Though Self-contradictory (Mündler et al., 2023) involves revision, they aim to remove the conflicting information between the original sentence and the synthetically-generated contradictory sentence, instead of correcting factual errors in the original sentences. We classify it into the second category: detection only. RARR (Gao et al., 2022), FacTool (Chern et al., 2023) and CoVe (Dhuliawala et al., 2023) are three most relevant work to ours.

Given a LLM response, RARR and CoVe first generate a series of questions covering different aspects of the response, which serve as queries in the evidence retrieval, and then edit the whole response to correct factual errors. Such coarse granularity verification may miss out incorrect statements, particularly over long documents, and also makes it difficult to spot false spans precisely, thus disabling fine-grained (e.g., correct only a false number in a statement) and flexible edits (e.g., delete a completely-hallucinated sentence). Additionally, revising the whole document tends to result in poor preservation of the original input (e.g., style, vocabulary, and structure), introducing irrelevant descriptions and even new hallucinations. Claim-level editing empowers precise correction and good preservation.

FacTool performs fact-checking over claims. However, gold claims are required as input for the system. That is, users must first decompose an output from a LLM into a list of checkable atomic claims by themselves before using FacTool to check, which complicates the fact-checking process. Moreover, it is expensive to use FacTool to check a piece of text, since the whole checking process calls APIs including OpenAI (\$0.06/1K tokens), Serper (\$1.00/1k queries), and Scraper.<sup>4</sup> This also challenges the evaluation where online API is not allowed to call with the consideration of internal data protection.

We attempt to alleviate these issues in our framework. We decompose the fact-checking task into eight subtasks. The design of decomposing and decontextualising a long document into independent sentences and then into atomic claims allows inputs of any granularity: document, sentence, or claim. The pipeline equipped with check-worthiness selection also naturally endows the flexibly-customised verification, such as skipping subjective statement, commonsense and the knowledge is well-known by the individual.

**Datasets** From the perspective of the evaluated benchmarks, as shown in Table 7, studies of the first category generally evaluate their methods on existing QA datasets, or revise hallucinations in a specific topic such as literature review and biographic generations (Chern et al., 2023; Dhuliawala et al., 2023). These topics may not be frequently requested by general users in real-world scenarios.

<sup>4</sup><https://www.scraperaapi.com/pricing/>

Dataset	Granularity	Factual label	Revision	Length	Size
HaluEval	document	✓	✗	82.0	4,507
FELM-WK	segment	✓	✗	51.1	184
FactPrompts	claim	✓	✗	41.8	50
Factcheck-GPT	claim	✓	✓	73.1	94

Table 8: Statistics of world-knowledge factuality evaluation benchmarks. Length=the average number of words of LLM responses.

1162 Studies of the second category contribute a spectrum of benchmarks to detect diverse hallucinations,  
1163 such as synthetically-generated contradictory sentences (Mündler et al., 2023), deliberately-generated  
1164 hallucinated answers (Li et al., 2023) and non-factual completions given a prefix context (Muhlgay et al.,  
1165 2023). Manakul et al. (2023) manually annotate factual labels (major/minor inaccurate and accurate)  
1166 given a sentence in the generated Wikibio passage.

1167 Interestingly, Yin et al. (2023) collected 1,032 unanswerable questions from five diverse categories  
1168 *no scientific consensus, imagination, completely subjective, too many variables, philosophical*, and their  
1169 2,337 answerable counterparts. Unanswerable questions refer to questions where the model should express  
1170 uncertainty instead of delivering conclusive responses. Zhang et al. (2023a) collected three datasets, with  
1171 500 questions (all *No* or all *Yes* answers) for each. One focuses on one type of question, including whether  
1172 a number is a prime, senator search (whether a US city has a specific university), and whether there is a  
1173 flight from one city to another given a graph connection.

1174 However, these datasets are either only applicable in detection, or originate from a single task like  
1175 biography writing (Min et al., 2023), without accounting for variations across different generations.  
1176 HaluEval’s annotation over Alpaca 5K responses of various instructions, which is one of the most similar  
1177 works to ours. They ask human annotators to label whether the response contains hallucinated information  
1178 (*Yes* or *No*) and list the corresponding spans if there exist errors (Li et al., 2023).<sup>5</sup> FELM with 184  
1179 world-knowledge questions is labelled in the granularity of segments, while ours are over fine-grained  
1180 claims to locate factual errors more precisely. Moreover, our annotations not only include factual labels  
1181 of each claim, but the revised text and labels of all involved subtasks as well, e.g., decomposition of a  
1182 sentence into a list of independent claims, check-worthiness of a sentence/claim, evidence stance and so  
1183 on.

1184 FacTool evaluate over a knowledge-based QA dataset FactPrompts consisting of 50 (prompt, response)  
1185 pairs. It is annotated by authors over atomic claims and their factual labels (true/false), but the responses  
1186 tend to be short, instead of long documents (see Table 8). Overall, our dataset offers both factual labels  
1187 and the revised text in three-level granularity — atomic claims, decontextualized sentences, and responses,  
1188 for LLM answers, with an emphasis on long documents.

1189 FELM (Chen et al., 2023) is the most relevant concurrent work with ours, but only annotated sentence-  
1190 level *true* or *false* labels (no correction). We construct a new dataset that collects (question, ChatGPT  
1191 response) pairs in real conversations. Annotators identify and edit factual errors for each atomic claim  
1192 decomposed and decontextualized from the original long-form responses. This is expected to serve as a  
1193 benchmark to evaluate the performance of fact-checkers.

<sup>5</sup>The hallucination is considered from the following three aspects: unverifiable, non-factual, and irrelevant.



<b>C Dataset</b>	1194
<b>C.1 Sources</b>	1195
<b>Twitter posts and in-house brainstorming:</b> We first collect (question, response) pairs from ChatGPT/GPT-4 failures found on social media, in Web articles, and in related papers. <sup>6</sup> The query should satisfy the criteria that the corresponding response must have factual errors, rather than failures regarding reasoning, math, coding, bias, and so on; (query, response) also should be independent of a dialog. This results in 23 examples. We additionally brainstorm a spectrum of questions depending on individual usage experience of ChatGPT and then select 22 questions whose responses contain factually-false content by manually verifying suspicious facts.	1196 1197 1198 1199 1200 1201 1202
<b>Dolly-15k</b> It consists of 15,011 examples, with eight categories ranging from closed, open, and general QA, to creative writing, brainstorming, information extraction, summarisation and classification. <sup>7</sup> Since we pay attention to open-domain generations and responses with more factual statements, closed and open-question answering pairs are chosen to be the database.	1203 1204 1205 1206
We first generate ChatGPT responses for 1,773 closed QA pairs without using context information (a paragraph extracted from Wikipedia relevant to the question), and 3,700 open QA pairs. After filtering questions that cannot be answered without context as well as questions ChatGPT does not answer, we further filtered responses with fewer than 200 characters. Taking human answers as the gold reference, we assume that if machine generations are semantically far from human answers, they may contain false information. So we keep the examples where the cosine similarity $\leq 0.5$ between human answer and machine response based on SimCSE sentence embedding. Finally, we select 563 examples from closed QA and 528 from open QA, thus 1,136 (question, response) pairs in total with 45 from the first source.	1207 1208 1209 1210 1211 1212 1213 1214
<b>C.2 Data Selection</b>	1215
The whole annotation process is extremely time-consuming, about 15-30 minutes, for an instance, even if with intermediate results from automatic methods to ease the procedure. This requests us to cherry-pick examples that highly satisfy two criteria — fact-intensive and factually-false. Therefore, we leverage FActScore to filter cases with the following four steps.	1216 1217 1218 1219
<b>Sentence split and atomic claims breaking-down</b> We first split a document into sentences using the NLTK tokenizer. The most straightforward way is to prompt ChatGPT to split a sentence into claims given the response as context. However, processing sentences one by one consumes both time and API tokens.	1220 1221 1222 1223
Therefore, given the whole response as the context and the first sentence of the response, we ask ChatGPT to break the input sentence into independent atomic claims, and also continue the decomposition of the next sentence of the response (see the prompt in Section C.5). Specifically, ChatGPT is given three demonstration examples, so that it can follow the instruction to first break down the input sentence into atomic claims, and then sequentially find the next sentence and make the splits. Over 90% examples follow the instruction, breaking down the whole response. 105 out of 1,136 examples only decompose the first sentence, on which we process sentence by sentence based on the NLTK sentence splits.	1224 1225 1226 1227 1228 1229 1230
Another reason why we ask ChatGPT to re-split the response into single sentences is that we observed that some sentences are incorrectly split into smaller units by NLTK, such as decomposing a paper reference into a set of metadata, while ChatGPT can remain the citation reference as a whole. <sup>8</sup> A weakness of ChatGPT outputs compared with traditional models is that it is sometimes non-trivial to parse the results from the text-free responses when ChatGPT does not follow the output format as the instruction. In such cases, we have to process examples specifically.	1231 1232 1233 1234 1235 1236
<b>Discussion:</b> One may argue that why not directly decompose the whole response into atomic claims, but through single sentences and then to atomic claims? There are two reasons.	1237 1238

<sup>6</sup><https://github.com/giiven95/chatgpt-failures>

<sup>7</sup>Its use is subject to the *CC BY-SA 3.0* license.

<sup>8</sup>In our dataset, we prioritise sentence splits by ChatGPT, using NLTK results for unsuccessfully-parsed instances. The prompt is initialised with the first sentence split by NLTK.

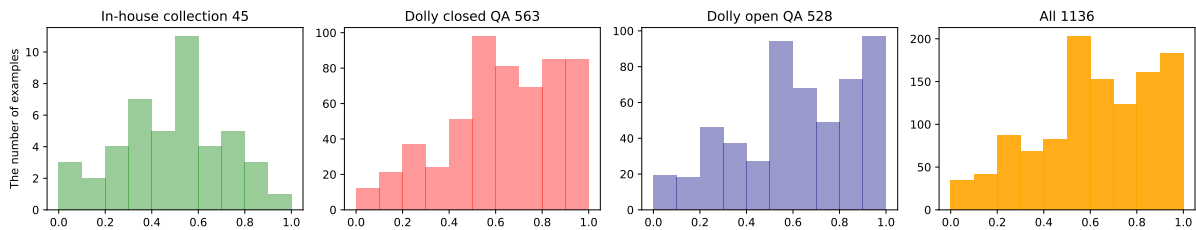


Figure 3: FActScore distribution for three component sources and their combination.

- *Avoid distortion*: atomic claims decomposed and decontextualized from a whole response by models such as ChatGPT tend to either lose or hallucinate information compared to the original response, while the quality of atomic claims of a single sentence is much better;
- *Improve annotation quality*: sentences as the intermediate state, it is easier for annotators to go through 1-5 atomic claims for a sentence as one annotation unit, instead of >5 claims for a whole response (particularly long documents), which helps annotators to pay attention and improve the annotation quality.

**Evidence collection for atomic claims** Given an atomic claim, following Gao et al. (2022), we first prompt ChatGPT to generate search queries for the claim, and then Google Search is used to get relevant web pages. We further split the retrieved documents into passages by sliding windows, and use a re-ranker combining lexical and semantic similarity to identify the most relevant passages for the given query, in which Sentence-BERT (Reimers and Gurevych, 2019) serves for semantic embeddings.<sup>9</sup> Finally, we aggregate evidence for all queries and select the top-5 evidences per atomic claim.

**FActScore calculation** FActScore (Min et al., 2023) is an automatic metric for fine-grained evaluation of the factuality of long-form generations. Given a generation, FActScore is calculated as the percentage of atomic claims within the generation that are supported by a knowledge source. For verifying the claim, we use the gathered evidences as input, along with the claim, and an instruction-tuned model as the verifier.

**Example selection** Figure 3 shows the FActScore distribution of three component sources and the whole data set. We keep all 45 pairs from the first source, and Dolly examples whose FActScore is less than 0.2, resulting in 33 closed question-answering pairs and 37 open questions, in total of 115 examples. We remove a similar question (7 and 13 are similar), and four questions where the LLM did not provide helpful answers due to its inherent disability to access real-time data. For example, the LLM cannot browse the internet and does not have access to the latest information (“which paper got the most citations in the question generation area?” and “which large language model contains the most parameters?”), or up-to-date data and event-specific details (“who was the general chair of COLING 2023”), or individual information (“what are the awards that Prof. William Yang Wang have?”). We eventually annotated 110 examples in our first annotation stage, and more cases would be annotated in the next stage.

**Discussion** One may concern that selecting questions where LLMs are prone to hallucinate or produce factual errors in the response may introduce bias into the dataset towards hallucinating errors. However, our benchmark is intended not to evaluate the factuality ability of the LLM, but to measure the verification accuracy of automatic fact-checkers. Hence, we chose questions that are selectively hard prompts, so that the LLM responses contain challenging claims which distinguish the performance of different automatic fact-checking approaches. We argue that if most claims in the benchmark are true and easy to verify, the benchmark would not be of much use to evaluate fact-checker’s performance.

Another issue is how Google’s use of LLM in their queries might impact the quality of the retrieved and gathered evidence. In our view, we expect to have access to the web articles from the traditional web

<sup>9</sup>cross-encoder/ms-marco-MiniLM-L-6-v2: <https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2>

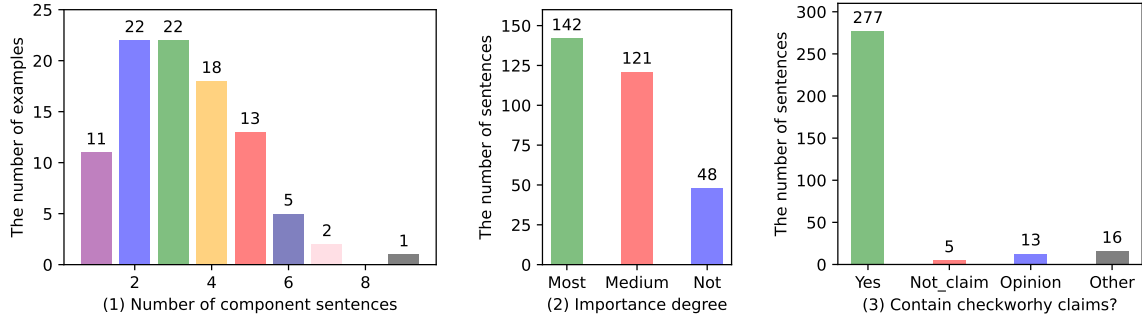


Figure 4: **Sentence analysis:** (1) Distribution of the number of sentences for each response; (2) Importance degree of sentences to answer the question (The distribution of the most important sentences to answer the question, intermediate important and not important); (3) The number of sentences across four types in terms of whether the sentence contains statements requiring fact-checking, Not\_claim refers to *not a claim*, such as a question.

search engine. It would be worthwhile to see whether the LLM answers to queries can be directly used as evidence for verification, given concerns of hallucinations even in retrieval-augmented generation settings.

### C.3 Annotation

As many studies illustrated, annotating a LLM factuality benchmark is a highly challenging task (Chen et al., 2023; Li et al., 2023). Our preliminary trials, in which authors manually annotate labels of all subtasks, empirically confirm the pain.

**Preliminary Trial** Based on the annotation guideline (see Appendix D), we first conduct an in-house annotation for ten examples, each example has two annotators. We annotate the whole process for all steps and manually type results into a *json* file as the pre-defined format. This attempt exposes three issues.

First, it is extremely time-consuming. It takes more than four hours for a fully-focused annotator to annotate a document of  $\sim 400$  words with about 20 sentences, in which evidence collection takes the most time and effort, particularly for topics with which the annotator is not familiar. Second, it is ineffective to extract relevant evidence passages by human eyes and basic string matching from retrieved Google search documents. This not only takes time but, most importantly, takes the risk of missing the most relevant evidence due to limited traversal. It is impractical for humans to go through all relevant Web articles and select the most semantically-relevant and reliable ones in a limited time. Humans are good at judging or making decisions, while machines are good at traversing. Lastly, it is hard to reach a high agreement between annotators, especially for subtasks of decomposition, evidence collection, and stance identification.

### C.4 Data Analysis

During annotation, we remove another 16 examples because there is no standard gold answer for these questions, such as seven involving a flow of procedures, six non-factual questions, one tricky riddle-like question, one broken generated answer, and one highly-disagreed case, resulting in 94 instances.

From the perspective of LLM users, we may expect to assess any answers and identify whether they are true and reliable, including the cases deleted in our setting. It should be highlighted that the questions involving a flow of procedures, tricky riddles, or non-factual questions need to be verified, while they are just out of the verification scope of the current fact-checking pipelines that only concern facts. The causality and the global logic behind the whole answer are under-explored.

**Sentences:** Most responses contain 2-5 sentences, with the longest response encompassing 9 sentences as shown in Figure 4. 142 sentences are considered to be the most important sentences, 121 and 48 fall into intermediate and not important. 278 sentences contain checkworthy statements, 16, 12 and 5 are categorised into *other*, *opinion*, and *not a claim*.

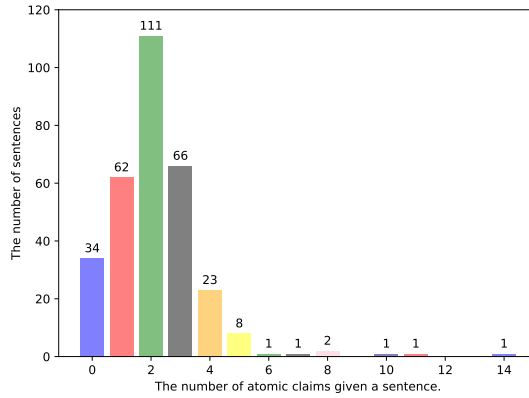


Figure 5: The distribution of component atomic claims amount given a sentence.

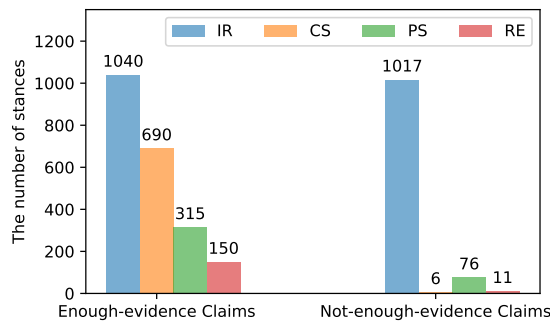


Figure 6: **Stance distribution** of claims with enough automatically-retrieved evidence to determine the factuality vs. claims without enough evidence (*CP*: completely support, *PS*: partially support, *RE*: refute, *IR*: irrelevant).

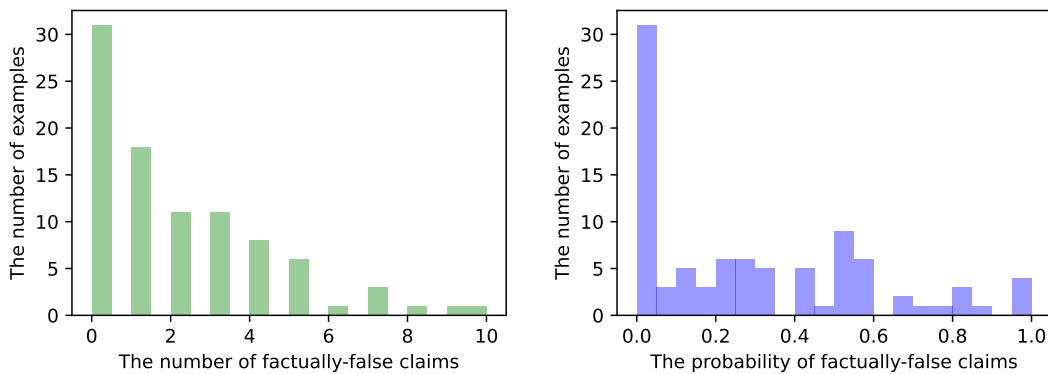


Figure 7: The number of false claims given an example.

1308  
1309  
1310  
1311  
1312  
1313

**How does the evidence support the claim?** Two-thirds pieces of irrelevant evidence (2057/3305). We compare the stance distribution of claims in which automatically-retrieved evidence is enough to determine its factuality and the claims that cannot be determined by automatic evidence in Figure 6. Though the majority of evidence are irrelevant for both groups, there are only 17 strong-position stances (“completely support”: *CS* and “refute”: *RE*) in the latter, compared with 690 *CS* and 150 *RE* in the former.

## C.5 Prompt to Generate Atomic Claims

Table 9: **Prompt** used to decompose and decontextualize a sentence into a set of independent atomic claims. We use three examples as demonstrations to elicit ChatGPT follow the instructions, break the response into sentences, as well as break a sentence into atomic claims.

Field	Content
<b>Prompt</b>	<p>Depending the context, please breakdown the following sentence into independent facts.</p> <p><b>Context:</b> The United States has had two black presidents: Barack Obama, who served two terms from 2009 to 2017, and Donald Trump, who served one term from 2017 to 2021. Obama was the first black president in the history of the United States. He was born in Honolulu, Hawaii, to a mother from Kansas and a father from Kenya. Trump was the second black president. He was born in New York City and previously served as a businessman and reality television personality.</p> <p><b>The sentence is:</b> The United States has had two black presidents: Barack Obama, who served two terms from 2009 to 2017, and Donald Trump, who served one term from 2017 to 2021. <b>Atomic facts for this sentence are:</b></p> <p>[          "The United States has had two black presidents: Barack Obama and Donald Trump.",          "Black president Barack Obama served two terms from 2009 to 2017.",          "Black president Donald Trump served one term from 2017 to 2021."          ]</p> <p><b>The sentence is:</b> Obama was the first black president in the history of the United States. <b>Atomic facts for this sentence are:</b></p> <p>[          "Obama was the first black president in the history of the United States."          ]</p> <p><b>The sentence is:</b> He was born in Honolulu, Hawaii, to a mother from Kansas and a father from Kenya. <b>Atomic facts for this sentence are:</b></p> <p>[          "Barack Obama was born in Honolulu, Hawaii.",          "Barack Obama mother was from Kansas.",          "Barack Obama father was from Kenya."          ]</p> <p><b>The sentence is:</b> Trump was the second black president.  <b>Atomic facts for this sentence are:</b></p> <p>[          "Trump was the second black president."          ]</p> <p><b>The sentence is:</b> He was born in New York City and previously served as a businessman and reality television personality.  <b>Atomic facts for this sentence are:</b></p> <p>[          "Donald Trump was born in New York City.",          "Donald Trump previously served as a businessman",          "Donald Trump previously served as a reality television personality."          ]</p>

Field	Content
	<p><b>Context:</b> In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas. He was born on October 16, 1898, and served on the Supreme Court from 1939 until his retirement in 1975. Therefore, in 1980, Justice Douglas was still alive and would have been the oldest serving justice on the Court at that time.</p> <p><b>The sentence is:</b> In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas.</p> <p><b>Atomic facts for this sentence are:</b></p> <p>[      "In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas."      ]</p> <p><b>The sentence is:</b> He was born on October 16, 1898, and served on the Supreme Court from 1939 until his retirement in 1975.</p> <p><b>Atomic facts for this sentence are:</b></p> <p>[      "Justice William O. Douglas was born on October 16, 1898."      "Justice William O. Douglas served on the Supreme Court from 1939 until his retirement in 1975."      ]</p> <p><b>The sentence is:</b> Therefore, in 1980, Justice Douglas was still alive and would have been the oldest serving justice on the Court at that time.</p> <p><b>Atomic facts for this sentence are:</b></p> <p>[      "Therefore, in 1980, Justice Douglas was still alive."      "Justice William O. Douglas would have been the oldest serving justice on the Court in 1980."      ]</p> <p><b>Context:</b> There have been only four female presidents of the United States in the country's history, so it is difficult to determine an average height for this group. The four female presidents were:      1.Abigail Adams (1797-1801)      2.Marilyn Carlson Nelson (2009-2013)      3.Luci Baines Johnson (1973-1977)      4.Hillary Clinton (2017-2021)</p> <p><b>The sentence is:</b> There have been only four female presidents of the United States in the country's history, so it is difficult to determine an average height for this group.</p> <p><b>Atomic facts for this sentence are:</b></p> <p>[      "There have been only four female presidents of the United States in the country's history.",      "It is difficult to determine an average height for four female presidents of the United States."      ]</p> <p><b>The sentence is:</b> The four female presidents were: 1.Abigail Adams (1797-1801) 2.Marilyn Carlson Nelson (2009-2013) 3.Luci Baines Johnson (1973-1977) 4.Hillary Clinton (2017-2021)</p> <p><b>Atomic facts for this sentence are:</b></p> <p>[      "Abigail Adams (1797-1801) is a female president of the United States.",      "Marilyn Carlson Nelson (2009-2013) is a female president of the United States.",      "Luci Baines Johnson (1973-1977) is a female president of the United States.",      "Hillary Clinton (2017-2021) is a female president of the United States."      ]</p> <p><b>Context:</b> {}  <b>The sentence is:</b> {}  <b>Atomic facts for this sentence are:</b></p>

Field	Content
Input	<p><b>Context:</b> It is difficult to say exactly how many stars are formed in the Milky Way each year, as the rate of star formation can vary over time. However, it is generally thought that the Milky Way produces about one new star per year on average. This may seem like a small number, but keep in mind that the Milky Way is a vast galaxy with billions of stars, so even a small number of new stars added each year can add up quickly.</p> <p><b>First sentence of the document:</b> It is difficult to say exactly how many stars are formed in the Milky Way each year, as the rate of star formation can vary over time.</p>
Output	<p><b>Atomic facts for this sentence are:</b></p> <p>[          "It is difficult to say exactly how many stars are formed in the Milky Way each year.",          "The rate of star formation in the Milky Way can vary over time."          ]</p> <p><b>The sentence is:</b> However, it is generally thought that the Milky Way produces about one new star per year on average.</p> <p><b>Atomic facts for this sentence are:</b></p> <p>[          "The Milky Way produces about one new star per year on average."          ]</p> <p><b>The sentence is:</b> However, This may seem like a small number, but keep in mind that the Milky Way is a vast galaxy with billions of stars, so even a small number of new stars added each year can add up quickly.</p> <p><b>Atomic facts for this sentence are:</b></p> <p>[          "The Milky Way is a vast galaxy with billions of stars.",          "Even a small number of new stars added each year can add up quickly in the Milky Way."          ]</p>

## D Annotation Guidelines

Based on the annotation framework, we aim to construct a benchmark for evaluation, assessing whether a fact-checking pipeline is effective in detecting and correcting factual errors in texts. The benchmark comprises annotations for the whole fact-checking process spanning seven subtasks for 110 (question, response) pairs, in which most responses are generated by ChatGPT, and some are by GPT-4. This section introduces the annotation guidelines, and Section 3 and 3.2 provide details of data collection and annotation.

For each example, annotators are given a pair of (question, response). A response is either an answer generated by LLMs responding to users' question, or a document returned by LLMs according to users' request. Annotators are required to give outputs of each step shown in Figure 2. We describe how to annotate for component subtasks throughout the pipeline, particularly clarifying how to deal with possible ambiguous scenarios.

### D.1 Decompose

It is subjective to decide the granularity of decomposition. We may aim to break down a long document into a set of atomic claims, while the definition of an atomic claim varies. Here, we practically apply the following strategy:

- Start by decomposing into single sentences.
- If the sentence contains too much information, break it into several components, but annotators do not overdo it, e.g., decomposing *Capitol Hill riots happened on January 6, 2021* to one claim for a year and one for the day.
- If several pieces of information are strongly dependent on each other, they are expected to co-occur in one snippet of evidence text, no more breaking-down is needed.

### D.2 Decontextualise

The criteria of decontextualisation are to ensure that all separated statements fully preserve semantics presented in the original context. For example, a statement that *it happened on Jan 6, 2021* loses information in decomposition, which makes it uncheckable. In such cases, annotators should replace pronouns, such as *it, they, those, these, this, that*, with specific entities or events after decomposition. Decontextualisation is mostly needed over cases with coreference relation. For complex relations, such as two sentences are strongly dependent on each other, we encourage to go back to the step of decomposition and keep the original text without breaking-down.

### D.3 Identify checkworthy claim

We consider two aspects in check-worthiness identification:

- If a statement presents subjective opinions, then it is not checkworthy.
- If the objective facts presented in a statement are commonsense, as obvious as *sun rises from the east*, it is not worth checking.

Therefore, we regard a statement as checkworthy claim when it presents objective facts, and these facts are not apparent commonsense. There is a special case. If the objective facts presented in a statement are not publicly available information. Namely, we cannot collect any evidence over web pages related to the claim, such as personal experience. They are regarded as uncheckable claims.

Specifically, for each statement, annotators are asked to answer two questions. Which category does this claim fall into? (1) factual claim; (2) subjective opinion; (3) not a claim; and (4) other. Is this statement worth checking? (1) Yes; and (2) No.



<b>D.4 Retrieve and collect evidence</b>	1357
Given a checkworthy claim, annotators are asked to search and collect the five most relevant snippets of text as evidence based on general web pages (including Wikipedia pages). Annotators are allowed to use any form of queries in retrieval, e.g. questions covering some aspects of the claim, or entities in the claim, and they need to record all queries and indicate those used for searching the most relevant evidence.	1358 1359 1360 1361
Note that five pieces of evidence is not a hard criterion. If less than five (even only one) pieces of evidence are sufficient to verify the input claim, and they are from reliable sources, annotators are allowed to collect <5 results. Meanwhile, if a claim involves a controversial topic, annotators are also encouraged to collect more than five results.	1362 1363 1364 1365
For each piece of evidence, record meta-data including (1) corresponding query, (2) citation (URL) of the web page from which this piece of evidence is extracted, (3) judgement of whether the source of evidence is reliable or not, <sup>10</sup> and (4) indicator whether this individual evidence is sufficient to verify the input claim.	1366 1367 1368 1369
The aforementioned guidelines are applicable to claims for which there exists evidence over web pages. However, there are situations where there is not any information on public web pages, e.g. personal experience. They are objective facts, but are not extensively known by the public. Put differently, they are uncheckable. Annotators can give empty list of evidence for uncheckable claims.	1370 1371 1372 1373
<b>D.5 Identify evidence stance</b>	1374
Given a claim and five pieces of most relevant evidence, annotators judge whether the evidence supports, partially supports, refutes or is irrelevant to the claim (see definition of stance in Section 2).	1375 1376
<b>D.6 Determine correction</b>	1377
For a claim, there will be K snippets of text (evidence), corresponding stance vectors $[s_1, s_2, \dots, s_K]$ and source reliability values $[r_1, r_2, \dots, r_K]$ . We skip all irrelevant evidence and follow the criteria below to determine whether edits are needed for a claim.	1378 1379 1380
<ul style="list-style-type: none"> <li>• If the claim is completely supported by evidence, no edit.</li> <li>• If the claim is completely refuted by evidence, check the evidence and make edits accordingly one by one.</li> <li>• If some evidence supports the claim and some refute it, this means there are conflicts between evidence (see Appendix E for example). In such a scenario, we consider both the source reliability and the number of evidence falling into each stance. If the voice of “refute” is stronger than “support”, we edit, otherwise remains the original text.</li> <li>• If some refute and some partially support, there are two possible situations depending on whether the supported partition is the same as the refuted partition: (1) if what is supported and what is refuted are the same partition, there are conflicts between evidence, follow the steps above; and (2) if they support and refute different partition of the claim, edit the refuted partition.</li> </ul>	1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391
<b>D.7 Edit, Merge and Deduplicate</b>	1392
In correction, we keep the principle of making minimal edits against the original text to correct factual errors. Annotators do not add extra information provided by evidence that is not directly targeted at factual errors. No extra deletion, insertion or addition. Finally, annotators merge all statements, either revised or original ones, in order, and deduplicate repeated information with the principle of minimal edits.	1393 1394 1395 1396

---

<sup>10</sup>Source reliability can also automatically be collected from MBFC/AllSides/Politifact/, but they apply for a small number of sources.

## E Conflicting Evidence Example

### 2. "land" can be found on Earth, Jupiter and Mars but not on Neptune and the Sun.

#### Evidence 1

For land on astronomical objects, see Planetary surface . For other uses, see Land (disambiguation) . Land , also known as dry land , ground , or earth , is the solid terrestrial surface of Earth not submerged by the ocean or another body of water . It makes up 29.2% of Earth's surface and includes all continents and islands . Earth's land surface is almost entirely covered by regolith , a layer of rock , soil , and minerals that forms the outer part of the crust .

Link: <https://en.wikipedia.org/wiki/Land>

Does this evidence support the claim?

Completely support  Partially support  Refute  Irrelevant

#### Evidence 2

Has there been any lore revealed how exactly a city can exist on Neptune? Yes, technically Neptune has a surface, but the pressure and temperatures are so high as to make a settlement existing there (even if it be a ship that crash landed) rather... unlikely. Read more More posts you may like

Link: [https://www.reddit.com/r/DestinyTheGame/comments/117tvbj/neptunes\\_surface/](https://www.reddit.com/r/DestinyTheGame/comments/117tvbj/neptunes_surface/)

Does this evidence support the claim?

Completely support  Partially support  Refute  Irrelevant

Figure 8: A claim with conflicting stance evidence: *partially support* and *refute*.

## F Confusion Matrix of Subtasks

1398

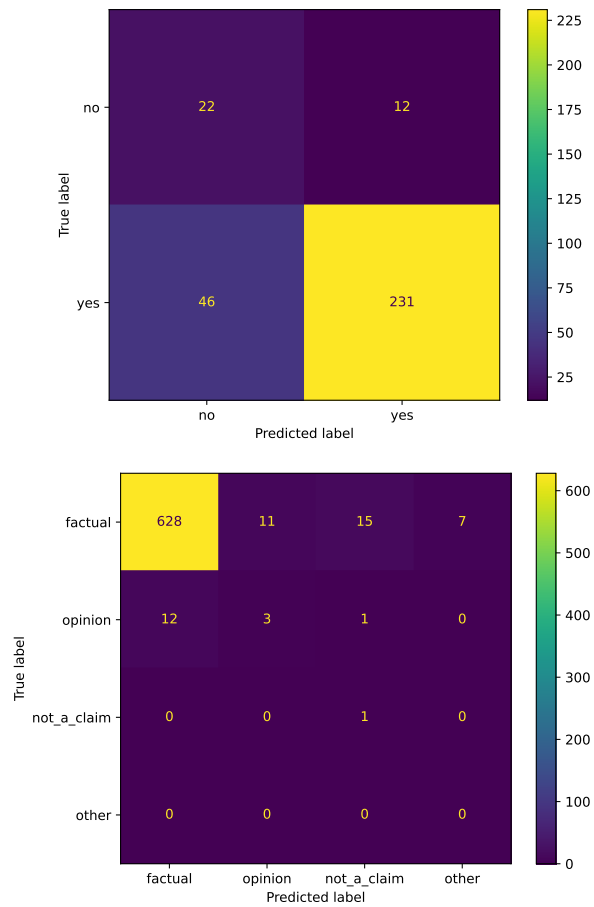


Figure 9: **ChatGPT checkworthiness** detection confusion matrix: sentence (top) and claim (bottom)

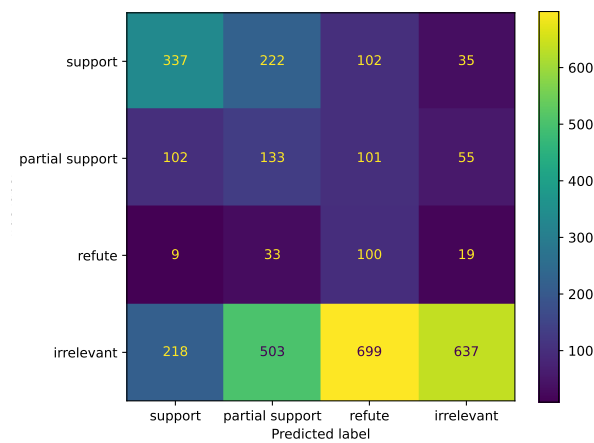


Figure 10: **(claim, evidence) Stance** detection confusion matrix based on ChatGPT with four labels: completely support (support in short in the figure), partial support, refute and irrelevant.

label	<b>LLaMA2</b>			<b>ChatGPT</b>			<b>RoBERTa-large-mnli</b>		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
irrelevant	0.69	0.54	0.61	0.85	0.47	0.60	0.71	0.52	0.60
refute	0.06	0.49	0.10	0.14	0.53	0.22	0.15	0.65	0.24
support	0.48	0.12	0.19	0.53	0.76	0.63	0.75	0.65	0.69

Table 10: **Stance** detection by prompting zero-shot ChatGPT, LLaMA2 and employing a fine-tuned NLI model in three label space: support, refute and irrelevant.

## G Example of Subtask 5

1399

<b>Prompt1 without question</b>	Given a document containing factual errors, please correct the errors in the document depending on a corresponding list of factually true claims. Note that preserve the linguistic features and style of the original document, just correct factual errors. document: {response} true claims: {correct claims} revised document:
<b>Prompt2 with question</b>	Given a question, and an answer containing factual errors, please correct the errors in the document depending on a corresponding list of factually true claims. Note that preserve the linguistic features and style of the original document, just correct factual errors. question: {prompt} document: {response} true claims: {claims} revised document:
<b>Question Response</b>	Who was the oldest justice on the US supreme court in 1980? In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas. He was born on October 16, 1898, and served on the Supreme Court from 1939 until his retirement in 1975. Therefore, in 1980, Justice Douglas was still alive and would have been the oldest serving justice on the Court at that time.
<b>Correct claims</b>	In 1980, the oldest justice on the United States Supreme Court was Justice William J. Brennan Jr. Justice William O. Douglas was born on October 16, 1898. Justice William O. Douglas served on the United States Supreme Court from 1939 until his retirement in 1975. In 1980, Justice William O. Douglas died. Justice William O. Douglas was the oldest serving justice in the history United States Supreme Court, sat on the Court from 1939 to 1975 and died in 1980.
<b>Human revision</b>	In 1980, the oldest justice on the United States Supreme Court was Justice William J. Brennan Jr.
<b>Prompt1, GPT-3.5-turbo</b>	In 1980, the oldest justice on the United States Supreme Court was Justice William J. Brennan Jr. He was born on October 16, 1898, and served on the Supreme Court from 1939 until his retirement in 1975. Therefore, in 1980, Justice Douglas had already passed away and would not have been the oldest serving justice on the Court at that time.
<b>Prompt1, GPT-4</b>	In 1980, the oldest justice on the United States Supreme Court was not Justice William O. Douglas, but rather Justice William J. Brennan Jr. Justice Douglas, who was born on October 16, 1898, notably served on the Supreme Court from 1939 until his retirement in 1975. However, by 1980, Justice Douglas was no longer alive, having passed away that year. Previously, he had held record as the oldest serving justice in the history of the United States Supreme Court.
<b>Prompt2, GPT-3.5-turbo</b>	In 1980, the oldest justice on the United States Supreme Court was Justice William J. Brennan Jr. He was born on October 16, 1898 and served on the Supreme Court from 1939 until his retirement in 1975. Therefore, in 1980, Justice Douglas had already passed away and Justice Brennan would have been the oldest serving justice on the Court at that time.
<b>Prompt2, GPT-4</b>	In 1980, the oldest justice on the United States Supreme Court was Justice William J. Brennan Jr., not Justice William O. Douglas. While Justice Douglas was indeed born on October 16, 1898, and served on the Supreme Court from 1939 until his retirement in 1975, in 1980, he was no longer alive, contrary to the original claim. It's true that Justice Douglas was considered the oldest serving justice in the history of the United States Supreme Court, his tenure spanning from 1939 to 1975, but by 1980, he had passed away. Thus, the distinction of being the oldest justice on the Supreme Court in 1980 goes to Justice Brennan Jr.

Table 11: Outputs (revised response) comparison using different prompts and models (GPT-3.5-turbo and GPT-4). The response by Prompt1 using GPT-4 is preferred.

## H Annotation Interfaces

### H.1 Decomposition, Decontextualization and Check-worthiness detection

**Example 1**

Who was the oldest justice on the US supreme court in 1980?

1. In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas.
2. He was born on October 16, 1898, and served on the Supreme Court from 1939 until his retirement in 1975.
3. Therefore, in 1980, Justice Douglas was still alive and would have been the oldest serving justice on the Court at that time.

Please upload a generated result json.

UPLOAD

Please select a sentence by clicking it on the left.

1. In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas.

Decontextualized sentence

COPY

Is this sentence the most important one to answer the question?

Yes  No, intermediate  No, not important

Does the sentence contain statements requiring fact-checking (checkworthy)?

Yes  No

---

Please add claims if you see any.

**Statement 1**

In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas.

In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas. COPY

What type of claim is this statement?

Factual Claim  
 Opinion  
 Not a Claim (e.g. a question)  
 Other (e.g. I'm a language model.)

REMOVE

---

Which statement is the most important statement in the sentence:

ADD CLAIM
SAVE

Figure 11: Screenshot of the first annotation interface for Decomposition, Decontextualization, and Check-worthiness detection.

## H.2 Evidence stance identification and Claim correction

1402

The screenshot displays a web-based annotation interface. On the left, a yellow bar contains the claim: "In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas." Below this, a "NEXT EVIDENCE" button is visible. A "Please upload a generated result json" section shows a file named "screen2.json" with an "UNLOAD" button.

The main area on the right is titled "Please select a claim by clicking it on the left." It lists five evidence items, each with a "NEXT EVIDENCE" button and a "Does this evidence support the claim?" question. The evidence items are:

- Evidence 1:** A chronologically ordered list of Supreme Court justices from the earliest to the most recent. The state the justice took the judicial oath is used as the beginning date of service. Chief justices are denoted by italic text. John Jay (1789-95), James Wilson (1789-98), John Rutledge (1789-91), William Cushing (1790-93), John Blair (1793-94), James Iredell (1790-99), Thomas Johnson (1762-93), William Paterson (1763-1806), Samuel Chase (1766-1811), Oliver Ellsworth (1795-1802), Bushrod Washington (1795-1829), Alfred Moore (1800-04), John Marshall (1801-35), William Johnson (1804-34).  
Support options:  Completely support,  Partially support,  Refute,  Irrelevant
- Evidence 2:** United States Senate. This article incorporates public domain material from Justice TSB to Present. United States Supreme Court. before Justice Cort. disclosed as improvionly granted Grant, waite, remond Shadow discler In chambers opinion Current members Chief Justice : John Roberts Associate justices :  
Support options:  Completely support,  Partially support,  Refute,  Irrelevant
- Evidence 3:** The Supreme Court of the United States is the final court of appeal and final exporator of the Constitution of the United States. The justices are appointed by the president of the United States and must be confirmed by the U.S. Senate. The chief justice of the United States is also appointed by the president and must be confirmed by the Senate. The number of justices fluctuated in the 18th and 19th centuries until 1869, when it was set at nine. A plan by Peck...  
Support options:  Completely support,  Partially support,  Refute,  Irrelevant
- Evidence 4:** The Supreme Court of the United States is the highest ranking judicial body in the United States. Established by Article III of the Constitution, the detailed structure of the Court was laid down by the 1st United States Congress in 1789. Congress specified the Court's original and appellate jurisdiction, created 13 judicial districts, and fixed the number of justices at six - one chief justice and five associate justices. The number of justices on the Supreme Court changed six times before settling at the present total of nine in 1869. [1] The following tables detail the succession of justices of the Supreme Court of the United States by seat.  
Support options:  Completely support,  Partially support,  Refute,  Irrelevant
- Evidence 5:** www.supremecourt.gov. Supreme Court of the United States. Archived from the original on December 7, 2020. Retrieved January 21, 2018. ^  
Support options:  Completely support,  Partially support,  Refute,  Irrelevant

Below the evidence items, there is a section "Can you determine whether the claim is true or false by evidence above?" with radio buttons for "Yes" (selected) and "No".

A text box contains the following text: "William J. Brennan (1906 - 1997) was the oldest justice on the United States Supreme Court in 1980. From table information from Wikipedia: https://en.wikipedia.org/wiki/List\_of\_Justices\_of\_the\_Supreme\_Court\_of\_the\_United\_States. We can see that in 1980, William J. Brennan is the oldest alive justice."

Below the text box, there is a "Does this claim need to be corrected?" section with radio buttons for "Yes" (selected) and "No".

At the bottom, there is a "COPY" button and a "self retrieved" label. A "SAVE" button is also present, along with a "DOWNLOAD COMMENTED JSON" button.

Figure 12: Screenshot of the second annotation interface: Evidence stance identification and Claim correction.

### H.3 Claim Merge and Deduplication

Who was the oldest justice on the US supreme court in 1980?

1. In 1980, the oldest justice on the United States Supreme Court was Justice William O. Douglas.
2. He was born on October 16, 1898, and served on the Supreme Court from 1939 until his retirement in 1975.
3. Therefore, in 1980, Justice Douglas was still alive and would have been the oldest serving justice on the Court at that time.
  - a. Therefore, in 1980, Justice Douglas was still alive.
  - b. William J. Brennan would have been the oldest serving justice on the Court in 1980.

Please upload a generated result json.

UPLOAD 0\_screen3.json

Please select a sentence by clicking it on the left.

**Sentence 1:**

In 1980, the oldest justice on the United States Supreme Court was William J. Brennan.

**Sentence 2:**

Justice William O. Douglas was born on October 16, 1898 and served on the Supreme Court from 1939 until his retirement in 1975.

**Sentence 3:**

Therefore, in 1980, Justice Douglas was still alive, but William J. Brennan would have been the oldest serving justice on the Court in 1980.

Please Merge All Revised Sentences, duplicate and make coherent:

In 1980, the oldest justice on the United States Supreme Court was William J. Brennan. Justice William O. Douglas was born on October 16, 1898 and served on the Supreme Court from 1939 until his retirement in 1975. Therefore, in 1980, Justice Douglas was still alive, but William J. Brennan would have been the oldest serving justice on the Court in 1980.

MERGE ALL

SAVE

DOWNLOAD COMBINED JSON

Figure 13: Screenshot of the third annotation interface: Claim Merge and Deduplication.