Truth or Mirage?  Towards End-To-End Factuality Evaluation with LLM-OASIS

Alessandro Scirè[1*], Andrei Stefan Bejgu[1*], Simone Tedeschi[1], Karim Ghonim[2], Federico Martelli[2], Roberto Navigli[1]

[1]Department of Computer, Control and Management Engineering, Babelscape & Sapienza University of Rome
scire@diag.uniroma1.it, bejgu@diag.uniroma1.it, tedeschi@diag.uniroma1.it, navigli@diag.uniroma1.it, info@babelscape.com
[2]Department of Computer, Control and Management Engineering, Sapienza University of Rome
ghonim@diag.uniroma1.it, martelli@diag.uniroma1.it

After the introduction of Large Language Models (LLMs), there have been substantial improvements in the performance of Natural Language Generation (NLG) tasks, including Text Summarization and Machine Translation. However, LLMs still produce outputs containing hallucinations, that is, content not grounded in factual information. Therefore, developing methods to assess the factuality of LLMs has become urgent. Indeed, resources for factuality evaluation have recently emerged. Although challenging, these resources face one or more of the following limitations: (i) they are tailored to a specific task or domain; (ii) they are limited in size, thereby preventing the training of new factuality evaluators; (iii) they are designed for simpler verification tasks, such as claim verification. To address these issues, we introduce LLM-OASIS, *to the best of our knowledge the largest resource for training end-to-end factuality evaluators.* LLM-OASIS *is constructed by extracting claims from Wikipedia, falsifying a subset of these claims, and generating pairs of factual and unfactual texts. We then rely on human annotators to both validate the quality of our dataset and to create a gold standard test set for benchmarking factuality evaluation systems. Our experiments demonstrate that* LLM-OASIS *presents a significant challenge for state-of-the-art LLMs, with GPT-4o achieving up to 60% accuracy in our proposed end-to-end factuality evaluation task, highlighting its potential to drive future research in the field.*

∗ Equal contribution.

Action Editor: Wei Gao. Submission received: 3 December 2024; revised version received: 8 August 2025; accepted for publication: 9 September 2025.

https://doi.org/10.1162/COLI.a.575

1. Introduction

In recent years, generative approaches in NLP have demonstrated remarkable results, achieving state-of-the-art performance across various tasks. This progress has been particularly notable with the advent of Large Language Models (LLMs), which have revolutionized the field, driving advancements in many tasks, including Text Summarization (Goyal, Li, and Durrett 2022; Pu, Gao, and Wan 2023; Zhang et al. 2023), Machine Translation (Alves et al. 2024; Zhang, Haddow, and Birch 2023; Wang et al. 2023), and Question Answering (Kamalloo et al. 2023; Rasool et al. 2024). However, a critical challenge remains as LLMs' outputs still contain hallucinations, i.e., content that cannot be grounded in any pre-existing knowledge (Tonmoy et al. 2024; Tam et al. 2022). Compounding the problem, LLMs generate highly fluent texts (Wang, Zhang, and Wang 2023), which may mislead users into trusting their factual accuracy. Therefore, developing modeling strategies to mitigate this issue and creating tools to detect and correct hallucinations has become urgent.

In this work, we focus on the problem of factuality evaluation, that is, the task of checking the factual accuracy of a machine-generated text. Previous research has proposed various resources to address this task. Although challenging, even for LLM-based factual reasoners, these resources are designed for specific settings, such as text summarization of news (Laban et al. 2021; Tang et al. 2023), books (Scirè, Ghonim, and Navigli 2024), and dialogues (Tang et al. 2024), among others. These benchmarks, while representative in their respective domains and tasks, often present peculiarities, which may lead to a lack of generalizability across different settings. A more general resource, pairing claims with evidence from Wikipedia is FEVER (Thorne et al. 2018); however, its applicability is limited by its focus on claim verification, which involves assessing the veracity of individual facts. This formulation is not well-suited to real-world scenarios, where texts typically contain multiple facts, thereby preventing the development of end-to-end factuality evaluation systems. These limitations highlight the need for a resource that is not restricted to a specific domain or task, offering broader applicability and enabling the design of complete factuality evaluation approaches.

In this context, we introduce LLM-OASIS, a large-scale resource for end-to-end factuality evaluation, created by extracting and falsifying information from Wikipedia pages. The overall process is depicted in Figure 1. As a result, we obtain 81k $\langle \text{factual}, \text{unfactual} \rangle$ pairs that are suitable for training end-to-end factuality evaluation systems.

Additionally, we set up a human annotation process to: (i) create a gold standard for the factuality evaluation task, useful for benchmarking LLMs, and (ii) validate the quality of the proposed data creation pipeline. Additionally, we issue two tasks to benchmark LLMs, namely, *end-to-end factuality evaluation* and *evidence-based claim verification*. Our experiments demonstrate that our resource is challenging even for state-of-the-art models, both in zero-shot and Retrieval Augmented Generation (Lewis et al. 2021, RAG) settings, with GPT-4o achieving an accuracy of 60% and 68%, respectively. In summary, our contributions are the following:

- We introduce LLM-OASIS, to the best of our knowledge the largest resource for end-to-end factuality evaluation, obtained by falsifying claims extracted from Wikipedia;
- Our resource enables two tasks to challenge current LLMs to detect factual inconsistencies in both short and long texts;

- We propose a gold standard benchmark, resulting from a human annotation process, to evaluate models on the proposed tasks;
- Our experiments demonstrate that our benchmark presents a significant challenge for LLMs, with smaller specialized models trained on LLM-OASIS achieving competitive performance.

Although we selected Wikipedia as the basis for our resource, we emphasize that our methodology can potentially be adapted to any other corpus in any domain or language, as the only requirement is access to a collection of raw texts. In the hope of fostering research in factuality evaluation, we release our resources at `https://github.com/Babelscape/LLM-Oasis`.

2. Related Work

Previous studies for factuality evaluation have focused on assessing **factual consistency**, i.e., the extent to which a generated text is grounded in a source document. Resources for this task typically include human annotations that indicate whether a

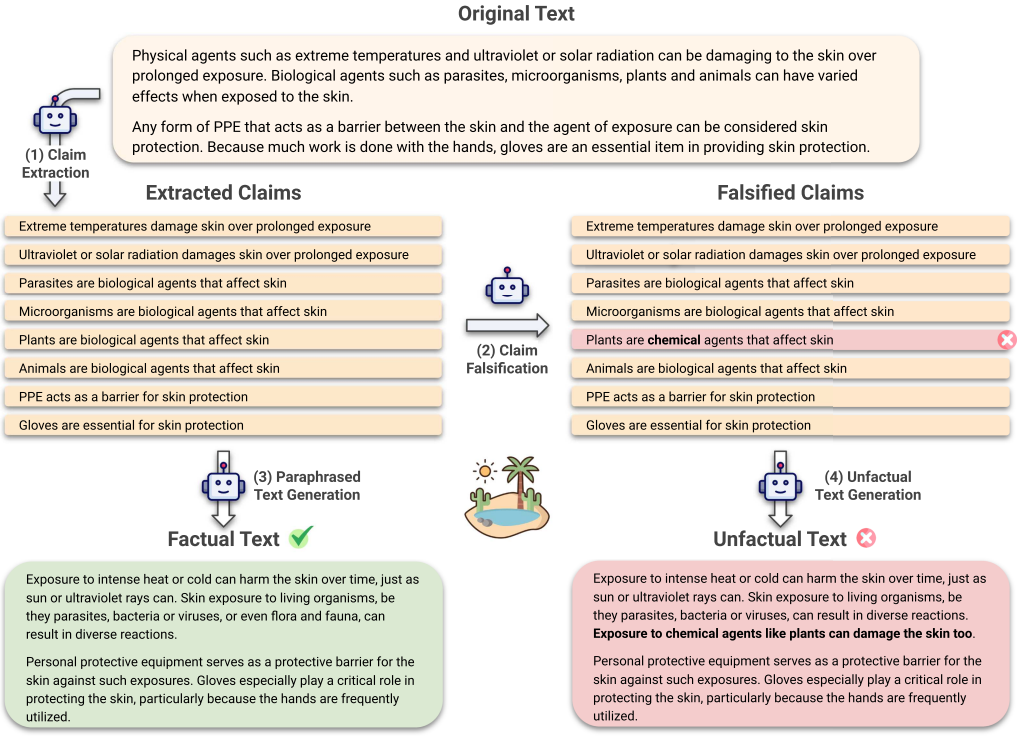

Figure 1
Pipeline for the creation of LLM-OASIS. Given a passage from a Wikipedia page (original text on top), we task an LLM to extract a list of atomic claims (1), falsify one of the extracted claims (2), and then, given the two sets of claims, produce a paraphrase of the original text (3), and an alternative version featuring the unfactual information (4).

generated text accurately reflects the original document's facts. However, many of these works are tailored for specific tasks and domains, such as the assessment of factual consistency in summaries of news (Fabbri et al. 2021; Tang et al. 2023; Pagnoni, Balachandran, and Tsvetkov 2021), books (Scirè, Ghonim, and Navigli 2024), and dialogues (Tang et al. 2024). Moreover, they are based on the assumption that the source of knowledge required for the verification is always available (e.g., the source document). This is not the case for the more general *factuality evaluation* task, in which a text in natural language must be verified regardless of the availability of the evidence, potentially requiring information retrieval techniques.

The first contribution towards general-purpose factuality evaluation dates back to Fact Extraction and VERification (Thorne et al. 2018, FEVER), which pairs claims with evidence retrieved from Wikipedia. The FEVER dataset comprises 185,445 human-generated claims, created by modifying sentences extracted from Wikipedia and subsequently verified without knowledge of the original sentences. The claims are classified as `Supported`, `Refuted`, or `NotEnoughInfo`, and, for the first two categories, annotators also recorded the sentence(s) forming the necessary evidence for their judgment. Although challenging, FEVER presents limitations due to its focus on fact verification, which involves checking the veracity of individual claims. This focus is hardly adaptable to real-world scenarios, where texts to verify usually feature multiple claims. Additionally, FEVER's annotation effort is limited to a relatively-small subset of 10k English Wikipedia pages.

Subsequent work has addressed the shortcomings of FEVER from complementary perspectives. First, HoVer (Jiang et al. 2020) preserves Wikipedia as its knowledge source and extends verification to multi-hop reasoning, yet it evaluates one claim at a time, ignoring the surrounding passage context and interactions among multiple claims. Moving the task into a specialized domain, SciFact (Wadden et al. 2020) pairs scientific claims with paper abstracts and sentence-level rationales, demonstrating that open retrieval can work beyond encyclopedic text; its focus, however, is restricted to biomedical and physics literature. To widen topical breadth, MultiFC (Augenstein et al. 2019) assembles about 35k real-world claims from 26 fact-checking outlets, yet each instance remains a stand-alone sentence, detached from its discourse context.

LLM-OASIS inherits the strengths of these resources while mitigating their weaknesses. It retains the multi-hop evidence requirement introduced by HoVer, provides multi-domain coverage comparable to MultiFC, and preserves the fine-grained claim–evidence alignment proposed in SciFact. Moreover, LLM-OASIS operates at the passage level, requiring systems to retrieve evidence, decompose the text into multiple interdependent claims, and verify them jointly, thereby capturing the full complexity of real-world factuality assessment. This design offers a large-scale, end-to-end benchmark that reflects the full complexity of real-world factuality assessment.

A recent line of research has focused on automatically generating synthetic benchmarks for factuality evaluation. Notably, Muhlgay et al. (2024) introduces FACTOR, a framework for generating factuality benchmarks by prompting an LLM to produce factual and unfactual completions given a prefix text. FACTOR includes 4,266 instances of ⟨prefix, completion⟩ pairs, each accompanied by a factuality label. Along the same lines, in proposing FELM, Chen et al. (2023) provide 847 LLM outputs focused on different types of knowledge, such as World Knowledge, Math, and Reasoning with human-made factuality annotations. While valuable for benchmarking LLMs, the limited size of these resources prevents them from being used to train new factuality evaluators.

With LLM-OASIS, we differentiate from previous studies by introducing a large-scale, task-agnostic resource covering a wide range of domains from Wikipedia.

Specifically, LLM-OASIS enables the task of end-to-end factuality evaluation, namely, the more realistic scenario that involves the verification of raw text in natural language. Notably, texts falling under this setting always go beyond individual sentences, inherently posing a more complex challenge to the systems. Additionally, to the best of our knowledge, it is the largest resource for this task, featuring 162,550 passages in natural language and 681,201 claims, which can be verified against knowledge from 81,275 Wikipedia pages covering a broad set of domains. Finally, we reserve a manually curated subset for this task, consisting of approximately 2k instances, and use it to benchmark several state-of-the-art models.

3. LLM-OASIS

In this section, we outline the steps required to generate LLM-OASIS. We start by selecting Wikipedia as our source of factual data due to its coverage of a wide range of topics and its frequent revisions, which help maintain accurate and up-to-date information. Moreover, to guarantee the quality of our data in terms of well-established and widely referenced information, we retain the most popular English Wikipedia pages.[1] Each page is then divided into passages of K sentences using a sliding window with stride of s sentences, forming our initial corpus.[2] Given a passage, as outlined in Figure 1, we task an LLM[3] to: (i) extract a list of atomic claims (**Claim Extraction**, Section 3.1); (ii) falsify one of the extracted claims (**Claim Falsification**, Section 3.2); and (iii) generate a paraphrase of the original passage, grounded on the extracted claims, along with an unfactual version incorporating the information from the falsified claim (**Factual and Unfactual Text Generation**, Section 3.3).

In the remainder of this section, for the sake of clarity, we describe the above-mentioned steps individually, but we disclose in advance that the step-specific outputs are obtained by means of a general, unified prompt containing the instructions for all the steps. The overall prompt is provided in Table 1.

3.1 Claim Extraction

The first step in creating LLM-OASIS involves extracting claims from an input passage t. We randomly sample one passage from each Wikipedia page and then extract a list of claims from each of the passages (cf. Step 1 in Figure 1).

Following Liu et al. (2023), we use the term **claim** to denote an atomic fact, i.e., an elementary information unit found in a text, that does not require further subdivision. We frame the claim extraction task as an end-to-end autoregressive generation problem. Let $\mathcal{M}$ be our generative model. Given an input passage t, we task $\mathcal{M}$ to extract the claims using the prompt $P_1(t)$ (cf. Step 1 in Table 1):

$$(c_1, \ldots, c_n) = \mathcal{M}(P_1(t)) \tag{1}$$

1 We select the 80k most visited pages in 2023.
2 In creating our resource, we set $K = 5$ and $s = 1$.
3 We used the GPT-4 API. More details in Appendix D.

Table 1
Prompt for the generation of data in LLM-OASIS.

Input: Wikipedia Passage of K sentences

Instructions: Execute the following steps:

Step 1 - Claim extraction: From the input passage, extract a comprehensive set of claims. These claims must be atomic, i.e., semantically coherent pieces of text that do not require further subdivision, and self-contained, i.e., not requiring additional context to be understood. Note that each claim must be short, using 15 words at most. Do not use "..." to truncate them. The ordering of the extracted claims must follow the logical flow expressed in the original text. Use a noun as the subject in the claim (avoid pronouns). All claims present in the input text must be included in the list.

Step 2 - Claim falsification: From the output of Step 1, subtly alter one claim, in order to introduce a critical factual inaccuracy. This claim to be altered must be the most relevant for the input text. It is forbidden to change dates, years, numbers, and person/location/organization/etc. names. It is also forbidden to provide naive negations of verbs, e.g., was -> was not, did -> did not. This step, i.e., Step 2, returns a pair containing the falsified claim along with the original one.

Step 3 - Factual text generation: From the output of Step 1, generate a text. Note that this text must be a paraphrase of the original provided text, i.e., a new text that should overlap as little as possible with the original, while preserving the meaning. The generated text must follow the same logical flow as the ordering of the extracted claims.

Step 4 - Unfactual text generation: Generate a text from the union of all extracted claims (including the falsified one) and the paraphrase, i.e., the output of Step 3. Therefore this text features a subtly unfactual piece of information. The generated text must follow the same logical flow as the ordering of the claims. The output text must be as similar as possible to the output of Step 3, except the unfactual part.

Output format: Return the output in a JSON with the following format: {'step_1': List[str], 'step_2': Tuple[str, str], 'step_3': str, 'step_4': str}. The output must be a valid JSON, thus try to avoid special characters like ' and " inside the JSON values, unless you escape them with a \. Do not include any marker for the falsified claim inside the JSON values, e.g., # this is the falsified claim. Please do not provide any preamble to your response, just provide the JSON.

where $(c_1, \ldots, c_n)$ represents the sequence of the generated claims. With the prompt $P_1(t)$, we aim at obtaining atomic[4] and self-contained claims, i.e., elementary units of information that do not require additional context in order to be verified. Specifically, we explicitly require the model to adhere to such formal definition, and, additionally, constrain it to generate short texts and avoid the usage of pronouns as subjects. For instance, given the **input passage**:

> *"The Amazon Rainforest, also known as Amazonia, is a moist broadleaf forest in the Amazon biome that covers most of the Amazon basin of South America. This region includes territory belonging to nine nations, with Brazil containing 60% of the rainforest."*

4 Liu et al. (2023) define a **claim** as an Atomic Content Unit (ACU), that is, an elementary unit of information found in a text that does not require further subdivision for the purpose of reducing ambiguity.

the model $\mathcal{M}$ returns the following list of **claims**:

1. The Amazon Rainforest is also known as Amazonia.
2. It is a moist broadleaf forest in the Amazon biome.
3. The Amazon Rainforest covers most of the Amazon basin of South America.
4. The region includes territory belonging to nine nations.
5. Brazil contains 60% of the rainforest.

Further examples of extracted claims can be found in Appendix A.

3.2 Claim Falsification

With the aim of producing an unfactual version of the original text, we introduce a critical factual error into one of the extracted claims. Formally, given the set of claims $C = (c_1, \ldots, c_n)$, i.e., the output of Equation (1), we task the model to falsify one of the claims[5] as follows:

$$(c_i, \bar{c}_i) = \mathcal{M}(P_2(C)) \tag{2}$$

where $P_2(C)$ is the prompt comprising the instructions for claim falsification, $\bar{c}_i$ the falsified claim, and c_i the corresponding factual one. We ask the model to provide the factual claim as well, thus enabling the investigation of the model's behavior.

As outlined in Table 1 (Step 2), we instruct the model to falsify only one of the extracted claims by introducing a critical yet subtle error, which makes it potentially challenging to detect. Moreover, inspired by findings from previous works about the manual creation of Natural Language Inference (NLI) resources (Parrish et al. 2021; Hu et al. 2020), we design the prompt with instructions to discourage the generation of naive contradicting instances, e.g., trivial negations of verbs. Continuing the example introduced in the previous section, given the extracted set of **claims**:

1. The Amazon Rainforest is also known as Amazonia.
2. The Amazon Rainforest is a moist broadleaf forest in the Amazon biome.
3. The Amazon Rainforest covers most of the Amazon basin of South America.
4. The region includes territory belonging to nine nations.
5. Brazil contains 60% of the rainforest. (c_i)

5 Our choice of falsifying only one claim per passage was intentional in order to increase the difficulty of the end-to-end factuality evaluation task. Identifying a text as "non-factual" when it contains multiple hallucinations is inherently easier than doing so when only a single, subtle falsehood is present.

the model $\mathcal{M}$ produces the following **falsified claim**:

The majority of the forest is contained within Peru. ($\overline{c_i}$)

In this example, the model replaces "Brazil" with "Peru", another country partially covered by the Amazon rainforest, making the falsification subtle and contextually plausible. Compared with FEVER (Thorne et al. 2018), which features manipulations ranging in difficulty from simple negations to semantically related entity substitutions, our approach focuses on context-aware falsifications that are deliberately more subtle. These are crafted to remain factually plausible and linguistically natural, making them significantly more challenging for language models to detect. Further examples of ⟨factual, unfactual⟩ pairs of claims can be found in Appendix A.

3.3 Factual and Unfactual Text Generation

Based on the claims extracted in the previous steps (cf. Sections 3.1 and 3.2), we now generate pairs of ⟨factual, unfactual⟩ texts, which populate our resource for factuality evaluation, thereby enabling the training and the benchmarking of factual reasoners.

Factual Text Generation. To make the factuality evaluation task more challenging, instead of using the original passages from Wikipedia as our factual texts, we leverage paraphrase generation. This approach produces texts that convey the same meaning as the original ones but with different surface forms, thereby making the verification task difficult for LLMs in both zero-shot settings—as the original texts could have been seen during pretraining—and RAG settings, which might retrieve the exact passages from Wikipedia. Formally, given the set of claims C resulting from Equation (1), we task the model to generate a factual text $\mathcal{F}$ grounded on such claims:

$$\mathcal{F} = \mathcal{M}(P_3(C)) \tag{3}$$

where $P_3(C)$ is the prompt with the instructions for obtaining a factual text through paraphrasing. As described in Table 1 (Step 3) we explicitly require $\mathcal{M}$ to follow the sequence of extracted claims to encourage a full coverage of the facts expressed in the original text. For instance, given the following **claims**:

1. The Amazon Rainforest is also known as Amazonia.
2. The Amazon Rainforest is a moist broadleaf forest in the Amazon biome.
3. The Amazon Rainforest covers most of the Amazon basin of South America.
4. The region includes territory belonging to nine nations.
5. Brazil contains 60% of the rainforest.

the model $\mathcal{M}$ generates the following **factual text**:

> *"Amazonia, widely known as the Amazon Rainforest, is a damp broadleaf forest located within the Amazon biome, covering a significant portion of the Amazon basin in South America. This vast region spans across nine countries, with Brazil housing 60% of the rainforest."*

See Appendix A for more examples of generated factual texts.

Unfactual Text Generation. Finally, the unfactual texts are generated through an analogous process, this time grounded on the set of claims that includes the unfactual one (from Equation (2)), namely, $\overline{C} = (c_1, \ldots, \bar{c}_i, \ldots, c_n)$. We obtain the unfactual text $\mathcal{U}$ with the generation process defined with the following:

$$\mathcal{U} = \mathcal{M}(P_4(\overline{C}, \mathcal{F})) \tag{4}$$

where $P_4(\overline{C}, \mathcal{F})$ is the prompt containing the guidelines for unfactual text generation and $\mathcal{F}$ is the factual text obtained from Equation (3). In particular, as specified in Table 1 (Step 4), we instruct $\mathcal{M}$ to produce a text identical to $\mathcal{F}$ except for the segment containing the factual error in order to ensure that the only confounding factor for the verification task is the unfactual portion of the text. This approach helps isolate the effect of the factual inaccuracy, preventing the model from introducing further inaccuracies. For example, given the **claims** in $\overline{C}$:

1. The Amazon Rainforest is also known as Amazonia.
2. The Amazon Rainforest is a moist broadleaf forest in the Amazon biome.
3. The Amazon Rainforest covers most of the Amazon basin of South America.
4. The region includes territory belonging to nine nations.
5. The majority of the forest is contained within Peru.

the model $\mathcal{M}$ generates the following **unfactual text**:

> *"Amazonia, widely known as the Amazon Rainforest, is a damp broadleaf forest located within the Amazon biome, covering a significant portion of the Amazon basin in South America. This vast region spans across nine countries, and the majority of the forest is contained within Peru."*

As shown in this example, the falsification is seamlessly embedded within a factually accurate and natural-sounding passage. This introduces an additional layer of complexity compared with FEVER, where claims are presented in isolation for verification. Here, models must not only assess the factuality of individual statements but also distinguish between verifiable facts and misinformation carefully woven into coherent, credible narratives. Additional examples of unfactual texts can be found in Appendix A. Finally, statistics about claim extraction, claim falsification, and factual and unfactual text generation process can be found in Table 2.

Table 2
Summary statistics for the creation of LLM-OASIS.

Claim Extraction	
# Pages	81,275
# Passages	81,275
Avg. Tokens per Passage	99.7
# Claims	681,201
Avg. Claims per Passage	8.381
Avg. Tokens per Claim	8.6
Claim Falsification	
# Unfactual Claims	81,275
Avg. Tokens per Unfactual Claim	9.0
Factual Text Generation	
# Factual Texts	81,275
Avg. Tokens per Factual Text	82.9
Unfactual Text Generation	
# Unfactual Texts	81,275
Avg. Tokens per Unfactual Text	86.5

4. The LLM-OASIS Benchmark

As a result of the steps described in Section 3, we obtained a large resource consisting of claims and texts (both factual and unfactual) that can be used to train end-to-end factuality evaluation systems. However, due to the automated nature of the proposed approach, it is crucial to both evaluate the quality of the produced data—by accurately evaluating the individual steps of our pipeline—and introduce a gold-standard benchmark for the task.

4.1 Human Evaluation

To assess the quality of our dataset and enable a rigorous evaluation of our procedure, we asked $M = 5$ expert linguists to validate a portion of $N = 1{,}750$ instances for each task in our pipeline (cf. Section 3 and Figure 1). Each annotator curated $(N/M) + K$ instances for each task with each of the M subsets having an overlap of $K = 100$ instances shared among all annotators. For the final benchmark, we resolved instances with disagreements through majority voting. We paid the annotators according to the standard salaries for their geographical location and provided them with task-specific guidelines, annotation examples, and a simple interface for each task. More details are provided in Appendix E.

Claim Extraction. For the claim extraction task, annotators received Wikipedia passages $(t_1, \ldots, t_N)$, each accompanied by a list of claims extracted by the model $\mathcal{M}$ as described in Section 3.1. The annotators' task was to verify whether each claim was appropriately represented in the corresponding passage (i.e., with the same semantics) and assess its atomicity.[6]

6 We chose to prioritize a precision-oriented evaluation for two key reasons: first, low coverage does not affect our proposed claim verification task (see Task 2, Section 4.2); and second, evaluating coverage would have required annotators to read the entire passage, making the annotation process more time-consuming and costly.

We evaluated the LLM's performance on this task by counting the human-annotated errors, yielding an accuracy of 96.78%. Additionally, we measured inter-annotator agreement, resulting in a Fleiss' κ score of 0.81. These results underscore both the high quality of the generated $\langle$text, claims$\rangle$ pairs and the strong agreement among the annotators.

Among the few errors produced by the LLM, we observed some occasional incorrect claims in the context of conditional clauses, where the model interpreted conditional or hypothetical statements as if they were factual claims. For instance, given the text: *In contrast, **if** interest rates were the main motive for international investment, FDI **would** include many industries within fewer countries. [...]*, the following incorrect claims were extracted: *Interest rates motivate international investment* and *Interest rates lead to FDI in multiple industries*, thus misrepresenting the original text which, instead, indicates a hypothetical scenario.

Claim Falsification. For this task, annotators received pairs of claims $\langle c_i, \overline{c_i} \rangle$ with c_i being one of the original claims selected from $(c_1, \ldots, c_n)$ and $\overline{c_i}$ the corresponding falsified claim produced by the model $\mathcal{M}$. The annotators' task was to verify whether each claim was appropriately falsified (i.e. with contradicting semantics). This required them to determine if $\overline{c_i}$ meaningfully diverged from c_i in terms of content and truthfulness, effectively capturing the model's ability to produce altered, incorrect versions of the original claims. Again, the model achieved a very high accuracy (98.55%). We measured a Fleiss' κ score of 0.84, showing almost perfect agreement between the annotators.

In this case, one of the most frequent error categories concerns instances where attempts at falsification manifest through minimal lexical variation, specifically by altering a single word. In these cases, such minor substitutions do not always yield a valid falsification. For example, consider the following claims: *Michael Ausiello **authored** the exclusive piece* and *Michael Ausiello **wrote** the exclusive piece*. As we can see, despite the substitution of the verb, the semantic congruence between the two claims is maintained, rendering the falsification attempt ineffective. An additional instance of this type is represented by the claims: *Washington, D.C. has **milder** winter weather than New York* and *Washington, D.C. has **warmer** winter weather than New York*.

Factual and Unfactual Text Generation. For these two tasks, we used a common format. Annotators received lists of original (or falsified) claims C (or $\overline{C}$) and the associated factual (or unfactual) texts produced by the model $\mathcal{M}$. The annotators' task was to verify whether each claim was correctly represented in the generated text. In the context of factual text generation, we additionally check whether the texts feature the same semantics as the claims but using a different wording. For the factual text generation step, we measured an accuracy of 90.36% and a Fleiss' κ score of 0.73. Similarly, for the unfactual text generation, we measured an accuracy of 89.2% and a Fleiss' κ score of 0.72.

In the factual text generation task, we occasionally observed omissions of details present in the extracted claims. For instance, the month "May" is omitted in the factual rewriting of the claim "Russian President Yeltsin formed the Russian Armed Forces in May 1992":

> Originally, the Armed Forces of the Russian Socialist Federative Soviet Republic, also acknowledged as the Red Army, served both the Russian SFSR and Soviet Union. [...] In 1992, Boris Yeltsin, the then Russian President, initiated the formation of the Russian Armed Forces, integrating a significant part of the Soviet Armed Forces.

Table 3
Performance of the chosen LLM $\mathcal{M}$ in the data generation process according to human evaluation (Accuracy), and the corresponding inter-annotator agreement (Fleiss' κ).

Task	Accuracy (%)	Fleiss' κ
Claim Extraction	96.78	0.81
Claim Falsification	98.55	0.84
Factual Text Generation	90.36	0.73
Unfactual Text Generation	89.20	0.72

We also found similar omissions in some unfactual texts, where a factual claim extracted from the original passage is not included in the generated unfactual version. In both cases, we stress that these occasional omissions do not compromise the factuality labels of the generated texts. Our manual validation process confirmed that the omitted content was not critical for determining the factual status of the passage in all cases. However, when constructing our gold benchmark (Section 4.2), we prioritize precision by discarding all generated texts, both factual and unfactual, that any annotator marks as containing an omission.

Overall, the reported detailed evaluations summarized in Table 3 show the efficacy and robustness of the proposed methodology for producing training data for the task.

4.2 Gold Benchmark

In this section, we describe the construction of our benchmark, along with the factuality-oriented tasks we propose. Specifically, we exploit the human annotations (cf. Section 4.1) to construct a gold-standard benchmark for model evaluation. To ensure the high quality of our data, we only retain the instances that were not marked as error by any of our annotators in any annotation stage (cf. Section 4.1). We use this data to propose two evaluation tasks, which we describe below.

Task 1: End-to-End Factuality Evaluation. The first task is to determine whether a given text contains any factual inaccuracies. Formally, given an input passage t, the model must output a binary label $y \in \{\text{True}, \text{False}\}$, where True indicates that the text is factually accurate and False indicates the presence of factual inaccuracies.

For this setting, we rely only on factual and unfactual texts as input passages, and discard the original texts, as the latter might have already been seen during the pretraining of LLMs. Specifically, to further ensure the high quality of our benchmark, we only retain the correct paraphrases that are generated from a valid set of claims (cf. *Factual and Unfactual Text Generation* and *Claim Extraction* in Section 4.1). Concerning the valid unfactual texts, instead, we only keep the ones that are: (i) generated, again, from a set of valid claims, and (ii) properly falsified and paraphrased (cf. *Claim Falsification* and *Factual and Unfactual Text Generation* in Section 4.1). We then label all the resulting factual and unfactual texts with *True*, and *False*, respectively.

In this setting, we aim at evaluating models on discerning true from fake texts (i.e., "Truth" from "Mirage"). This formulation enables the assessment of both plain LLMs and more complex RAG models. We deem this task to be particularly challenging as the falsification may involve even a single word occurring in one of the many claims featured in a text, in the spirit of recent works highlighting how LLMs struggle to deal

with subtle nuances in a large input text (Kamradt 2023; Hsieh et al. 2024; Laban et al. 2024; Wang et al. 2024)

Task 2: Evidence-based Claim Verification. In this setting, the task is to classify individual claims as factual or unfactual using a given piece of evidence. This formulation is essential for isolating and evaluating the models' reading comprehension abilities in factuality verification. By providing only the gold-evidence passage—rather than the full retrieval output—we focus solely on the models' capacity to interpret the evidence and verify a claim, thereby eliminating confounding factors related to retrieval quality. Formally, given an input claim c and a corresponding evidence passage e, the model must output a binary label $y \in \{\text{True}, \text{False}\}$, where True indicates that the claim is supported by the evidence and False indicates that the claim is not supported by the evidence.

For this setting, we focus on the extracted claims and their corresponding unfactual version, and use the factual text as evidence. We discard both the original and unfactual texts as the former might have already been seen during the pre-training of LLMs, while the latter contradicts real-world knowledge and, therefore, the internal knowledge of LLMs, possibly leading to unfair evaluations.

Additionally, to guarantee the high precision of our data, we focus on the claims that are both atomic and reflecting the same semantics of the original text (cf. *Claim Extraction* Section 4.1). Then, we only keep the claims that have been appropriately falsified (cf. *Claim Falsification* in Section 4.1), along with their unfactual counterparts. Finally, we apply the same quality checks described in Task 1 to retain only the valid factual texts.

At this stage, we classify the $\langle c_i, \mathcal{F} \rangle$ pairs with the label True, while we label $\langle \overline{c_i}, \mathcal{F} \rangle$ as False, with c_i and $\overline{c_i}$ being the original claim and its falsified version, respectively.

5. End-to-end Factuality Evaluation with LLM-OASIS

In this section, we showcase how LLM-OASIS can be leveraged to build an end-to-end factuality evaluation system. In the spirit of Min et al. (2023), we decompose the task of evaluating the factuality of a given text into three simpler tasks, namely, Claim Extraction, Evidence Retrieval, and Claim Verification. The process begins with extracting a set of atomic facts (cf. **Claim Extraction**, Section 5.1) from the text to be verified. These extracted claims are then used to retrieve relevant evidence from a reliable knowledge base (cf. **Evidence Retrieval**, Section 5.2). After this, the factual accuracy of each claim is evaluated by comparing it against the retrieved evidence (cf. **Claim Verification**, Section 5.3). Finally, the results of these individual evaluations are aggregated to determine the overall factuality of the entire text.

5.1 Claim Extraction

Our approach starts by extracting atomic claims from a given input text t. With the aim of training a claim extractor, we leverage LLM-OASIS to create a dataset of $\langle t, C \rangle$ tuples, where t is an original text from Wikipedia and $C = (c_1, \ldots, c_n)$ the corresponding automatically extracted claims by our chosen LLM $\mathcal{M}$ (Section 3.1). We then fine-tune a smaller sequence-to-sequence model $\mathcal{G}$ on this data, thus distilling the claim extraction capabilities of $\mathcal{M}$.

We frame the training process as a text generation task; more formally, we fine-tune $\mathcal{G}$ to generate the claims autoregressively:

$$P(y \mid t) = \prod_{k=1}^{|y|} P\left(y_k \mid y_{0:k-1}, t\right) \tag{5}$$

where y is the sequence obtained by concatenating the claims in C and y_k is a token in this sequence.

5.2 Evidence Retrieval

At this stage, given the claims extracted by $\mathcal{G}$, we require a system capable of retrieving relevant passages from a knowledge corpus to serve as evidence to verify those claims. Again, we leverage LLM-OASIS to create a training dataset for our retriever; in particular, given each generic claim $c_j \in C$ extracted from the original text t, we construct the following training pairs:

$$\langle c_j, t\rangle, \langle c_j, \mathcal{F}\rangle, \langle c_j, \mathcal{U}\rangle, \forall c_j \in C$$

where $\mathcal{U}$ and $\mathcal{F}$ are the generated factual and unfactual texts (cf. Section 3.3).

We then augment this set by pairing the factual and unfactual texts with the falsified claim $\overline{c_i}$ (cf. Section 3.2), thus obtaining the following additional training instances:

$$\langle \overline{c_i}, t\rangle, \langle \overline{c_i}, \mathcal{F}\rangle, \langle \overline{c_i}, \mathcal{U}\rangle$$

In this way, we include all possible pairs of ⟨claim, passage⟩ in LLM-OASIS in our training set. This strategy is aimed at increasing the generalization capabilities of our retriever: notably, given a claim, the retriever is trained to both provide the passages to support it along with the ones that are useful to contradict it.

Following the methodology outlined in Dense Passage Retrieval (Karpukhin et al. 2020, DPR), we define our retriever $\mathcal{E}$ as a Transformer-based encoder, which produces dense representations of both claims and passages. Starting from an input claim c and a knowledge corpus $\mathcal{D}$, we use $\mathcal{E}$ to compute a vector representation v_c for c, and v_p for every passage $\{p_1, p_2, \ldots, p_N\} \in \mathcal{D}$. Then, we use the dot product $v_c \cdot v_p$ to rank all the passages in $\mathcal{D}$ and, finally, extract the top k among these. We denote by $R_k(c, \mathcal{D})$ the retrieval function that, given a claim c and a corpus of passages $\mathcal{D}$, returns the top-k passages ranked by similarity. We minimize the DPR loss $\mathcal{L}$ to train $\mathcal{E}$:

$$\mathcal{L} = -\sum_{i=1}^{N} \log \frac{e^{v_{c_i} \cdot v_{p_i^+}}}{e^{v_{c_i} \cdot v_{p_i^+}} + \sum_{j \neq i} e^{v_{c_i} \cdot v_{p_j^-}}} \tag{6}$$

where N is the batch size, v_{c_i} is the vector representation of the i-th claim in the batch, $v_{p_i^+}$ is the vector representation of the corresponding gold passage for the i-th claim, and $v_{p_j^-}$ represents the vector representations of all the other passages in the batch, serving as in-batch negatives. This formulation ensures that the model learns to score the correct

passage higher than the other ones within each batch, which has been shown to be an effective strategy for training retrieval models (Yih et al. 2011; Gillick et al. 2019).

5.3 Claim Verification

The final step of our factuality evaluation methodology involves verifying each claim c generated by our claim extractor from the text t, by comparing it against the corresponding passages $R_k(c, \mathcal{D})$ retrieved from our corpus. Inspired by previous work on consistency evaluation (Zha et al. 2023; Chen and Eger 2023; Scirè, Ghonim, and Navigli 2024), we ground our verification approach in the NLI formulation. NLI is a task that determines the logical relationship between two texts: a *premise* and a *hypothesis*. Formally, given a premise *pre* and a hypothesis *hyp*: $\text{NLI}(pre, hyp) = Y \in \{\text{ENT}, \text{NEUT}, \text{CONTR}\}$, where Y is a label indicating whether *pre* entails (ENT), is neutral about (NEUT), or contradicts (CONTR) *hyp*.

Training a Claim Verifier on LLM-Oasis. In this section, we show how LLM-OASIS can be utilized to train a model for the claim verification task. Complying with the NLI formulation, we require a strategy to assess whether each claim extracted from a text is entailed, contradicted, or neutral with respect to a set of the retrieved passages. With this purpose, we construct a training dataset by deriving the following ⟨claim, passage, label⟩ triplets from LLM-OASIS:

$$\langle c_j, t, \text{ENT} \rangle, \langle c_j, \mathcal{F}, \text{ENT} \rangle, \forall c_j \in C$$

where $c_j \in C$ is a claim extracted by the LLM from the original text t (cf. Section 3.1), and $\mathcal{F}$ and $\mathcal{U}$ are the factual and unfactual texts outlined in Section 3.3.

We expand our training dataset for NLI with the following triplets:

$$\langle \overline{c_i}, t, \text{CONTR} \rangle, \langle \overline{c_i}, \mathcal{F}, \text{CONTR} \rangle, \langle \overline{c_i}, \mathcal{U}, \text{ENT} \rangle, \langle c_i, \mathcal{U}, \text{CONTR} \rangle$$

where $\overline{c}_i$ is the falsified version of the extracted claim c_i (cf. Section 3.2).[7] To obtain a complete NLI dataset, we require a strategy to generate neutral triplets as well. To achieve this, we first pair each claim c_j in C (Section 3.1) with the passages p_i of the Wikipedia page W from which the original text t was extracted. Then, we select the passage p^* as the one that maximizes the neutrality probability when fed to an NLI model[8] Ψ along with c_j:

$$p^* = \underset{p_i \in W}{\text{argmax}}\, \mathbb{P}_\Psi(\text{NEUT} \mid p_i, c_j)$$

and augment our dataset with the neutral pairs $\langle c, p^*, \text{NEUT} \rangle$. This approach increases the likelihood that the selected passages are semantically related to the claim, as they

7 While edge cases exist where certain instances might be misclassified as contradictions, the low error rate observed in the claim falsification process (1.45%) supports our decision to include these samples in the training dataset.

8 We used a DeBERTa-v3-large model fine-tuned on several NLI datasets. For more information: `https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli`.

Algorithm 1 Algorithm for Claim Verification.

Require: claim c, top-k retrieved passages $\{p_1, p_2, \ldots, p_k\}$, NLI model Φ
for each passage p_i in $\{p_1, p_2, \ldots, p_k\}$ **do**
  $\hat{y} \leftarrow \Phi(c, p_i)$
  if $\hat{y} ==$ ENT **then**
    return True
  else if $\hat{y} ==$ CONTR **then**
    return False
  end if
  {The output of the model is NEUT, i.e., neutrality. Continue to the next passage}
end for
return True {All NLI outputs are neutral, c is deemed verified}

come from the same Wikipedia page, while still being neutral. This is preferable to randomly selecting neutral examples, as it tends to provide more meaningful contrasts.

Finally, we fine-tune a Cross-Encoder model on this data; as a result of this process, we obtain our claim verification model Φ. More information about the training setup can be found in Section 6.1.

Claim Verification Algorithm. In Algorithm 1 we outline how we leverage Φ to assess the factuality of a claim. Our procedure takes as input a claim, a set of *top-k* retrieved passages, and a claim verification model. For each ⟨passage, claim⟩ pair we obtain a label $\hat{y}$ by applying Φ:

$$\hat{y} = \Phi(p_i, c) = \underset{y \in \{\text{ENT}, \text{NEUT}, \text{CONTR}\}}{\text{argmax}} P(y \mid p_i, c) \tag{7}$$

where p_i is a retrieved passage and c is a claim, which are fed to the NLI model Φ as the premise and hypothesis, respectively. As described in Algorithm 1, the algorithm proceeds by checking the output of this model for each passage in the ranking order. If Φ outputs ENT for a passage, the claim is deemed verified (i.e., return True). Conversely, if Φ outputs CONTR, the claim is deemed unfactual (i.e., return False). Finally, if Φ outputs NEUT for all passages, the claim is deemed verified (i.e., return True), as there is no contradicting evidence available.

In practice, given an input text t, we use our claim verifier to assign a factuality label to the claims generated by our claim extractor, using the passages returned by our retriever as sources of evidence.

The final factuality prediction for the text t is an aggregation of the claim-level factuality labels. Specifically, the text t is considered factual if all of its extracted claims are verified, unfactual otherwise.

6. Experimental Setup

In this section, we provide details about the models and data involved in our experiments. To train our components for the end-to-end factuality evaluation task, we leverage the synthetic data from LLM-OASIS (cf. Section 3, Figure 1). Specifically, we randomly split the passages in an 80/20 proportion to build the train and validation

datasets, respectively. When splitting, we ensure that all the claims, as well as the factual and unfactual text generated from the same passage, will end up in the same split.

We evaluate both our modular architecture (cf. Section 6.1) and several LLM-based baselines (cf. Section 6.2), showing the effectiveness of our benchmark in challenging factuality evaluation systems. To assess their performance, we rely on the LLM-OASIS gold-standard benchmark (Section 4.2). Models are evaluated across the two proposed tasks (i.e., *end-to-end verification* and *evidence-based claim verification*), and we use balanced accuracy (Brodersen et al. 2010) as our evaluation metric. All fine-tuning experiments and inference for models up to 8B parameters are conducted on a single NVIDIA GeForce RTX 3090 GPU. For larger models, specifically Phi-4 and Llama-3.3-70B-Instruct, we utilize an HPC cluster node equipped with 4 NVIDIA A100 GPUs.

6.1 Our Model

Here, we provide the training details for each module of our proposed solution for end-to-end factuality evaluation (cf. Section 5).

Claim Extractor. As described in Section 5.1, we build our claim extractor dataset with the ⟨text, claims⟩ tuples in the training split of LLM-OASIS. We split the resulting dataset into ∼67k passage-claims pairs for training, and ∼4k passage-claims pairs for validation. Statistics about the claim extraction dataset can be found in Table 2.

We fine-tune a $T5_{base}$ (Raffel et al. 2019) model on this data to generate the sequence of claims given an input passage. We train the model for a total of 1M steps, with Adafactor (Shazeer and Stern 2018) as optimizer with a learning rate of $1e^{-}5$.

Following Scirè, Ghonim, and Navigli (2024), we rely on the easiness_{F1} metric (Zhang and Bansal 2021) for model selection. Let C represent the set of generated claims for a given text and C^* the corresponding set of gold claims. To compute the easiness_P score, as defined by Zhang and Bansal (2021), we first calculate the ROUGE-1[9] score for each generated claim $c \in C$ by comparing it to every gold claim $c^* \in C^*$, and then select the maximum score. The final easiness_P score is obtained by averaging these maximum scores over all generated claims:

$$\text{easiness}_P(C, C^*) = \frac{\sum_{c \in C} \max_{c^* \in C^*} \text{R1}(c, c^*)}{|C|} \tag{8}$$

Similarly, we compute the easiness_R score by selecting the maximum ROUGE-1 score for each gold claim c^* with respect to all generated claims:

$$\text{easiness}_R(C, C^*) = \frac{\sum_{c^* \in C^*} \max_{c \in C} \text{R1}(c, c^*)}{|C^*|} \tag{9}$$

Finally, we combine easiness_P and easiness_R to calculate the easiness_{F1} score, and select the model that achieves the highest easiness_{F1} on our validation set.

Evidence Retriever. The training dataset of our retriever comprises ∼3.2M ⟨claim, evidence⟩ pairs. At validation/test time we construct the knowledge corpus with the

9 We consider ROUGE-1 to be a suitable basis for our easiness metric due to the high extractiveness of the claim extraction task.

original texts in our validation split and gold benchmark, respectively. To make the evaluation more realistic and challenging, we expand the corpus with passages from the same Wikipedia page. This approach results in our corpus $\mathcal{D}$ comprising a total of 2.5M passages.

We use the pre-trained Transformer-based architecture $E5_{base}$ (Wang et al. 2022) as our encoder $\mathcal{E}$. To generate embeddings for both claims and passages, we apply mean pooling over the output of $\mathcal{E}$. The model is trained with a batch size of 20 input texts for 300,000 steps, using AdamW (Loshchilov and Hutter 2019) as the optimizer. We use a learning rate of $1 \cdot 10^{-6}$, with a 20% warm-up phase.

Claim Verifier. As outlined in Section 5.3, we formalize the claim verification task as an NLI problem and construct a dataset of ∼3.5M ⟨premise, hypothesis, label⟩ triplets from LLM-OASIS. We devote 3.2M instances for training our claim verification model and the remaining 300k for validation. We fine-tune DeBERTa-v3$_{large}$ (He et al. 2021) for a total of 1M steps on this data, using Adafactor.

6.2 Evaluated LLMs

We provide a comprehensive evaluation of a set of LLMs on the LLM-OASIS benchmark. We evaluate a closed-source model from the GPT family—specifically, GPT-4o (OpenAI et al. 2024)—alongside open-weight LLMs such as Qwen-2.5 (Qwen et al. 2025), Llama 3 (Grattafiori et al. 2024), Mistral (Jiang et al. 2023), Phi-4 (Abdin et al. 2024), and Phi-4-mini (Microsoft et al. 2025), as well as Falcon-Mamba (Zuo et al. 2024), which serves as a representative of non-Transformer-based architectures. These models are selected owing to their widespread use in the literature and their demonstrated high performance on standard evaluation benchmarks. By analyzing systems with parameter counts ranging from 4B to 70B, we can assess how different architectural approaches perform on factuality evaluation tasks.

6.3 Evaluation Settings

Following standard practice in LLM evaluation, we assess model performance across multiple prompting strategies. Specifically, we consider four settings: Zero-Shot (ZS), Few-Shot (FS), Explain-Then-Answer (EX), and Retrieval-Augmented Generation (RAG).

Zero-Shot (ZS). In this setting, the LLMs are prompted with the instructions and the input without any additional guidance. This setting serves as a baseline for assesing the model's inherent ability to evaluate factuality.

Few-Shot (FS). To guide the model in performing the task, we utilize a few-shot learning approach by including a set of 5 manually labeled held-out examples within the prompt.

Explain-then-Answer (EX). In this setting, the model is required to generate an explanation before providing a factuality label. This structured response format encourages the model to engage in explicit reasoning, potentially making its decision process more interpretable and accurate.

Retrieval-Augmented Generation (RAG). As part of the end-to-end task evaluation, we ablate the impact of providing the LLMs with external knowledge, that is, in the RAG

Table 4
Easiness metrics of our claim extraction model on the LLM-OASIS gold benchmark (Section 4.1), computed with two backbone similarity measures (ROUGE-1 and BERTScore). We report easiness_P, easiness_R, and easiness_{F1}.

Metric	ROUGE-1	BERTScore
easiness_P	0.6805	0.9398
easiness_R	0.6408	0.9335
easiness_{F1}	0.6601	0.9367

setting. To experiment with this, we include the top-K passages[10] returned by our retriever (cf. Section 5.2) in the prompts. We extend the input by appending the retrieved passages after the text to be verified and a separator.

All the prompts used in the various settings can be found in Appendix B.

7. Results

7.1 Task 1: End-to-End Factuality Evaluation

In this section we present the results obtained in the end-to-end factuality evaluation task (cf. Section 4.2). First of all, we examine the performance of the claim extraction and evidence retrieval modules, which directly affects the end-to-end factuality evaluation process.

Claim Extractor. We evaluate our claim extraction by computing the easiness_{F1} score—the harmonic mean of easiness_R and easiness_P defined in Equations (8) and (9)—over the manually validated claims in our gold benchmark (Section 4.1). Table 4 reports the resulting easiness_{F1} when using ROUGE-1 and BERTScore (Zhang et al. 2020) as underlying similarity metrics.

We observe strong performance on our claim extraction benchmark, with easiness_{F1} reaching 0.66 under ROUGE-1 and 0.94 under BERTScore (Table 4). These results indicate that the extracted claims are both lexically and semantically well-aligned with the gold annotations. Given ROUGE-1's sensitivity to surface-level phrasing, an easiness_{F1} of 0.66 reflects high alignment, especially considering the inherent variability in how valid atomic claims can be expressed.

Evidence Retriever. We evaluate the performance of the evidence retrieval module using the Recall at k (R@k) metric, which quantifies the proportion of relevant documents retrieved in the top k results. Formally, it is defined as:

$$\text{R@k} = \frac{|\{\text{relevant } D\} \cap \{\text{top } k \text{ retrieved } D\}|}{|\{\text{relevant } D\}|} \tag{10}$$

This metric allows us to assess the ability of our retriever to identify relevant passages for factuality verification within the top-k ranked results. Higher values of k generally yield higher recall, as more documents are considered, but also introduce the risk of increasing irrelevant retrievals.

10 We selected $K = 30$ based on the analysis of our retriever's performance at different values of K (cf. Section 7.1) conducted on the validation set.

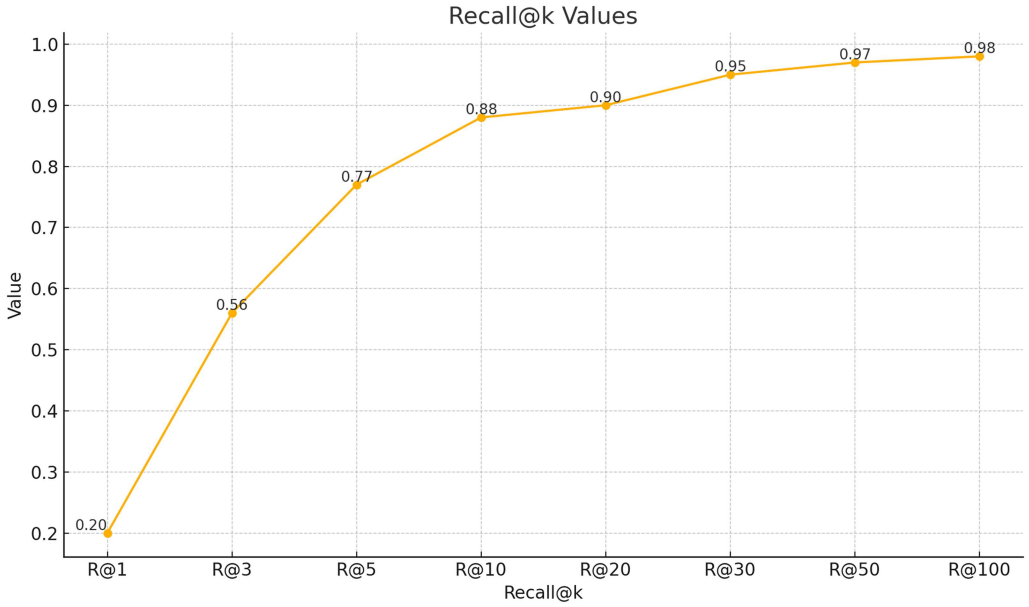

Figure 2
Recall@k performance of the $E5_{base}$ model at different values of k.

For our experiments, we evaluate different values of k (as shown in Figure 2) and ultimately select for all subsequent experiments $k = 30$ as it provides a balance between performance and efficiency. The fine-tuned $E5_{base}$ model achieves a Recall@30 (R@30) of 0.95. This is a significant improvement compared with the same model without fine-tuning, which only achieves an R@30 of 0.52. The fine-tuning process over 3.2M passages proved crucial for this performance gain. We remark that R@K represents an upper bound of our factuality evaluation performance when external knowledge is integrated into the verification process. Further analysis and details can be found in Appendix C.

End-to-End Factuality Evaluation. The results of the evaluated LLMs for the end-to-end factuality evaluation task are shown in Table 5. We conduct evaluations in Zero-Shot

Table 5
Balanced accuracy (%) on the gold benchmark of LLM-OASIS for end-to-end factuality evaluation. We report results of LLMs across different settings: Zero-Shot (ZS), Few-Shot (FS), and their respective Explain-the-Answer variants (ZS+EX, FS+EX). "Size" denotes the number of parameters in billions (B). Results are averaged over five runs with different seeds; standard deviations are reported as subscripts.

Model	Size	ZS	ZS+EX	FS	FS+EX
Phi-4-mini-instruct	4B	$54.4_{\pm 0.4}$	$52.6_{\pm 0.5}$	$53.3_{\pm 0.4}$	$52.3_{\pm 0.5}$
Falcon3-Mamba-7B-Instruct	7B	$55.3_{\pm 0.4}$	$52.3_{\pm 0.5}$	$53.0_{\pm 0.4}$	$54.1_{\pm 0.4}$
Mistral-7B-Instruct-v0.3	7B	$51.2_{\pm 0.5}$	$54.6_{\pm 0.4}$	$53.4_{\pm 0.4}$	$56.2_{\pm 0.5}$
Qwen2.5-7B-Instruct	7B	$55.8_{\pm 0.4}$	$52.7_{\pm 0.5}$	$57.2_{\pm 0.4}$	$\underline{57.3}_{\pm 0.4}$
Llama-3.1-8B-Instruct	8B	$53.6_{\pm 0.3}$	$54.9_{\pm 0.4}$	$55.0_{\pm 0.4}$	$55.5_{\pm 0.5}$
Phi-4	14B	$\underline{57.2}_{\pm 0.3}$	$\mathbf{57.9}_{\pm 0.4}$	$\underline{57.6}_{\pm 0.4}$	$57.0_{\pm 0.4}$
Llama-3.3-70B-Instruct	70B	$\mathbf{59.2}_{\pm 0.4}$	$\mathbf{57.5}_{\pm 0.4}$	$\mathbf{61.7}_{\pm 0.4}$	$\mathbf{60.0}_{\pm 0.4}$

Table 6
Balanced accuracy (B-Accuracy) on the gold benchmark of LLM-OASIS for end-to-end factuality evaluation. We compare different models in Zero-Shot (ZS) and Retrieval-Augmented Generation (RAG) settings. "Size" denotes the number of parameters in billions (B). Results for open-weight models are averaged over five runs with different random seeds, and standard deviations are reported as subscripts.

		B-Accuracy (%)	
Model	**Size**	**ZS**	**RAG**
Phi-4-mini-instruct	4B	$54.4_{\pm 0.4}$	$56.3_{\pm 0.4}$
Falcon3-Mamba-7B-Instruct	7B	$55.3_{\pm 0.4}$	$51.0_{\pm 0.6}$
Mistral-7B-Instruct-v0.3	7B	$51.2_{\pm 0.5}$	$50.5_{\pm 0.5}$
Qwen2.5-7B-Instruct	7B	$55.8_{\pm 0.4}$	$54.7_{\pm 0.4}$
Llama-3.1-8B-Instruct	8B	$53.6_{\pm 0.3}$	$54.1_{\pm 0.4}$
Phi-4	14B	$57.2_{\pm 0.3}$	$57.6_{\pm 0.4}$
Llama-3.3-70B-Instruct	70B	$59.2_{\pm 0.4}$	$58.8_{\pm 0.4}$
GPT-4o	N/A	**60.8**	**68.0**
Our Model (Fine-tuned)	**1B**	–	**69.2**$_{\pm 0.4}$

(ZS) and Few-Shot (FS) prompting, with each setting also assessed using the Explain-Then-Answer (EX) approach. As shown, the balanced accuracy scores of all evaluated LLMs remain low, often only marginally surpassing the random baseline. The top-performing model across all configurations is, unsurprisingly, the largest one—Llama-3.3-70B-Instruct—yet it reaches a maximum of just 61.7% accuracy in the FS setting. This outcome highlights that our benchmark is extremely challenging even for state-of-the-art LLMs. The core difficulty lies in the nature of the task: Models must have to assess the factuality of a text containing a subtle falsification seamlessly embedded within an otherwise factual context (cf. Section 3.2). Moreover, this shows that, despite likely having been exposed to the entire Wikipedia during the pretraining phase, the evaluated models still struggle to assign correct factuality labels.

In Table 6 we report the performance of our approach (cf. Section 5) compared to all the other evaluated models in ZS and RAG settings. Results show that our pipeline-based approach using small language models (cf. Section 5) achieves the highest balanced accuracy (69.2%), outperforming nearly all evaluated LLMs—with only GPT-4o, in the RAG setting, approaching comparable performance, performing slightly below our model (by approximately 1.2 percentage points). To verify whether these differences are statistically significant, we compute McNemar's test (McNemar 1947) across all model pairs in the RAG setting (see Figure 3). The results confirm that our model performs significantly better than all the LLMs ($p < 0.05$), with the only exception being GPT-4o ($p = 0.434$).

Although this outcome can partly be attributed to the fine-tuning of our model components on LLM-OASIS, it remains notable given that our system has considerably fewer parameters than its counterparts. We argue that this advantage stems not only from the quality of the training data but also from the modular design of our approach, which decomposes the factuality evaluation task into simpler subtasks. This structure allows small models to handle each task effectively, leading to overall performance on par with, or exceeding, that of much larger LMs. However, the fact that the best-performing system achieves a score of $\sim$0.70 further highlights the inherent

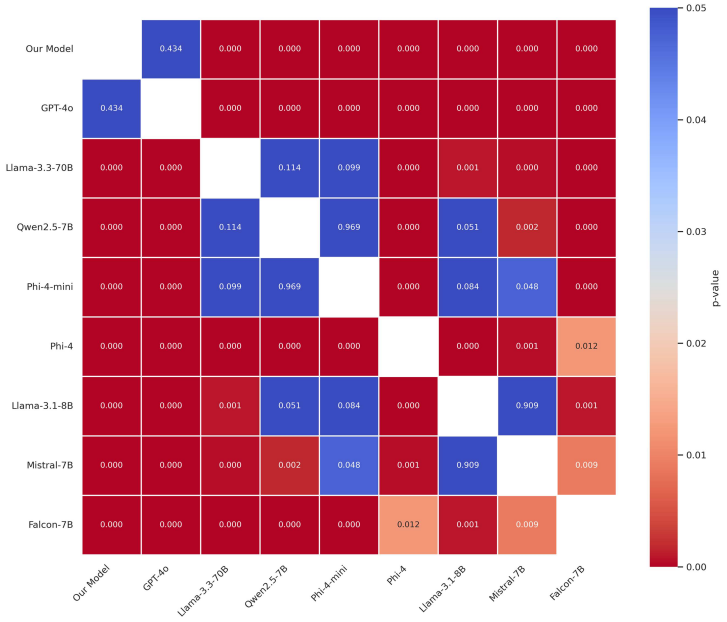

Figure 3
McNemar's test p-values for all model comparisons in the Retrieval-Augmented Generation (RAG) setting. Cells in blue indicate pairs with no statistically significant difference ($p > 0.05$); shades of red indicate significant differences, with darker tones for lower p-values.

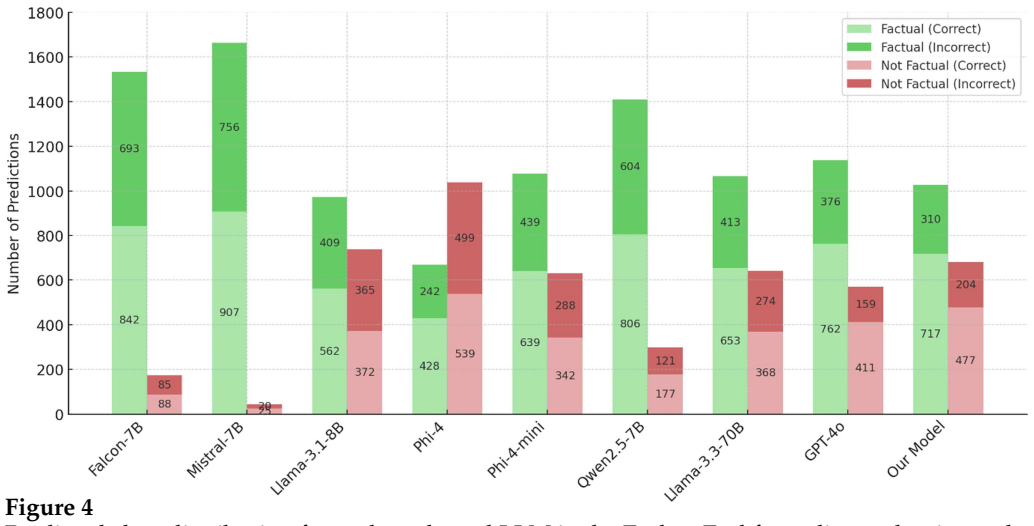

Figure 4
Predicted class distribution for each evaluated LLM in the End-to-End factuality evaluation task under the *RAG* setting, using the gold benchmark of LLM-OASIS. For each model, the left (green) bar represents the number of instances predicted as *Factual*, and the right (red) bar those predicted as *Not Factual*. Within each bar, the lower segment corresponds to correct predictions, while the upper (darker) segment indicates incorrect ones.

challenge proposed by our benchmark. This complexity is further supported by our prediction distribution analysis (Figure 4), which shows that nearly all models—except Phi-4—exhibit a strong bias toward labeling passages as "Factual," often failing to detect the subtle falsifications present in half of the examples. Notably, while Phi-4 does

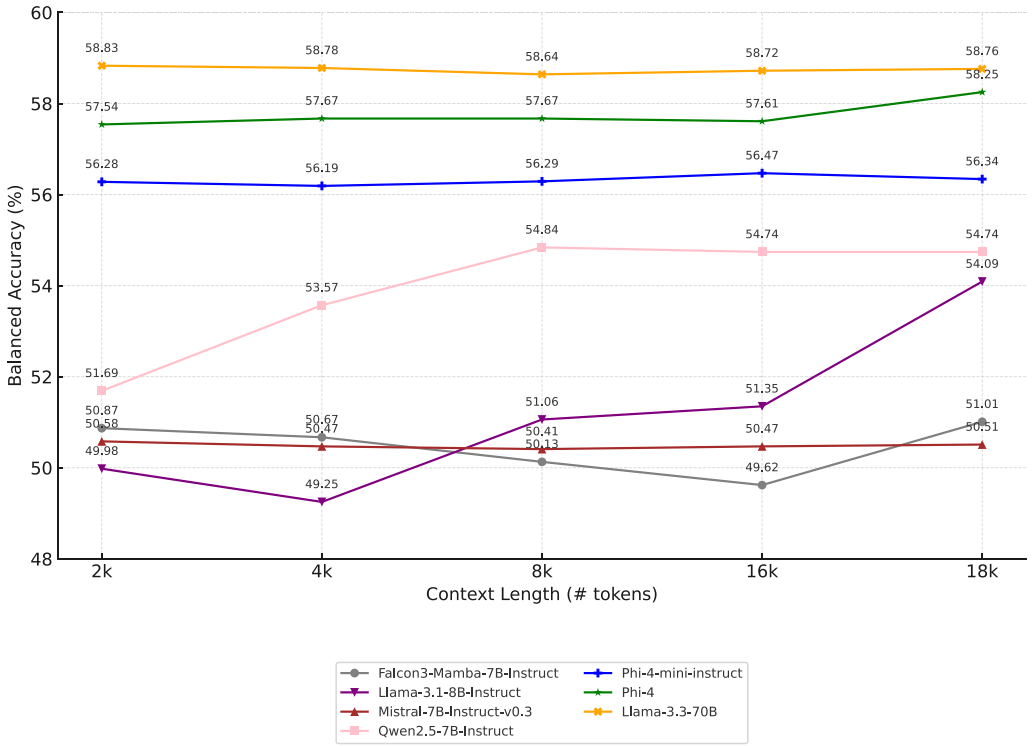

Figure 5
Balanced Accuracy (%) of different models in the Retrieval-Augmented Generation (RAG) setting with increasing context length. Each model is evaluated on inputs of 2k, 4k, 8k, 16k, and 18k tokens, which marks the length of the longest prompt instantiated in our evaluation.

not display this bias, it achieves at most 58.8% balanced accuracy, indicating limited effectiveness overall.

The results in Table 6 also suggest that incorporating retrieved external knowledge not only fails to consistently boost performance but may also result in degradation compared to the Zero-Shot (ZS) setting, with most LLMs—including Llama-70B—struggling to leverage the retrieved evidence effectively. To further investigate this, we assess whether increased context length confounds models in this setting, by evaluating LLMs while truncating their input at different lengths. Figure 5 reports balanced accuracy across varying input lengths, with 18k tokens marking the length of the longest prompt instantiated in our evaluation. Across most models, performance remains relatively stable as input size increases. Notably, Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct exhibit moderate upward trends, suggesting some benefit from extended context. These results suggest that context length alone does not hamper model performance. Importantly, at 18k tokens, we guarantee that in 95% of cases the passage required in order to verify the text is present in the retrieved evidence (cf. Figure 2). This setup allows evidence availability to be decoupled from reasoning: failures are most likely due to difficulties in exploiting the available information rather than to missing context. GPT-4o stands out in this regard, showing a +8 point gain in the RAG setting over ZS (cf. Table 6), suggesting a more effective reasoning capability over retrieved evidence compared to the evaluated open-weights models.

Table 7
Balanced accuracy (%) on the gold benchmark of LLM-OASIS for evidence-based claim verification. We compare different models across evaluation settings: Zero-Shot (ZS), Zero-Shot with Explanation (ZS+EX), Few-Shot (FS), and Few-Shot with Explanation (FS+EX). "Size" denotes the number of parameters in billions (B). Results for open-weight models are averaged over five runs with different random seeds, and standard deviations are reported as subscripts.

Model	Size	ZS	ZS+EX	FS	FS+EX
Phi-4-mini-instruct	4B	70.50$_{\pm 0.5}$	84.06$_{\pm 0.6}$	56.56$_{\pm 0.6}$	54.78$_{\pm 0.5}$
Falcon3-Mamba-7B-Instruct	7B	63.45$_{\pm 0.7}$	74.98$_{\pm 0.5}$	67.97$_{\pm 0.6}$	69.89$_{\pm 0.6}$
Mistral-7B-Instruct-v0.3	7B	72.87$_{\pm 0.5}$	80.67$_{\pm 0.5}$	73.40$_{\pm 0.6}$	77.81$_{\pm 0.6}$
Qwen2.5-7B-Instruct	7B	84.85$_{\pm 0.6}$	87.11$_{\pm 0.5}$	84.17$_{\pm 0.7}$	84.25$_{\pm 0.6}$
Llama-3.1-8B-Instruct	8B	78.56$_{\pm 0.6}$	86.55$_{\pm 0.7}$	76.94$_{\pm 0.5}$	76.53$_{\pm 0.6}$
Phi-4	14B	84.14$_{\pm 0.4}$	92.69$_{\pm 0.5}$	87.57$_{\pm 0.5}$	88.32$_{\pm 0.4}$
Llama-3.3-70B-Instruct	70B	**91.79**$_{\pm 0.4}$	**93.97**$_{\pm 0.3}$	**92.77**$_{\pm 0.3}$	**94.08**$_{\pm 0.4}$
GPT-4o	N/A	89.49	**93.93**	88.24	90.82

Table 8
Balanced accuracy (%) of our claim verification model fine-tuned on LLM-OASIS, evaluated on the gold benchmark. Results are averaged over five runs with different random seeds; standard deviation is reported as a subscript.

Model	Size	B-Accuracy (%)
Our Model (Fine-tuned)	0.4B	93.30$_{\pm 0.4}$

7.2 Task 2: Evidence-Based Claim Verification

In this section, we present the results for the second task we aim to evaluate with our benchmark (see Section 4.2), i.e., evidence-based claim verification. We first evaluate the performance of the LLMs across the studied prompt settings (ZS, ZS+EX, FS, and FS+EX). The outcomes of these experiments are presented in Table 7. Notably, all systems achieve higher performance compared with the previous setting (e.g., our system goes from 69.24 in the end-to-end task to 93.30 in this task). We attribute this to three main factors. First, this task is a simpler instance of the previous one, namely, the model is required to verify a single claim rather than a passage. Second, the system is provided with the exact evidence needed to verify the claim, whereas in the end-to-end formulation, each model relies on several passages returned by the retriever, hence possibly introducing noise in the process. Finally, the end-to-end verification implies reading and reasoning on a huge context (4k tokens on average) rather than the limited context (100 tokens on average) of this task.

To assess the effectiveness of a specialized model trained directly on our resource, we evaluate our claim verifier (cf. Section 5.3), thereby excluding the claim extraction and retrieval components from our pipeline. The results, shown in Table 8, indicate that our lightweight fine-tuned system (0.4B) obtains a very high balanced accuracy (93.30%). This provides further evidence that fine-tuning on high-quality task-specific data can enable a small model to rival or even outperform much larger LLMs in factuality evaluation tasks.

8. Conclusion and Future Work

In this article, we introduce LLM-OASIS, a large-scale resource for end-to-end factuality evaluation obtained by extracting and falsifying information from Wikipedia. Specifically, as outlined in Figure 1, given a text from Wikipedia, we extract a set of factual and unfactual claims, with the latter obtained by falsifying one of the facts expressed in the original text. Starting from these sets, we generate a factual text, which is a paraphrase of the original one, and its unfactual counterpart, featuring the falsified claim. This results in 81k ⟨factual, unfactual⟩ pairs that are suitable for training factuality evaluation systems, making LLM-OASIS the largest resource for this task. In contrast to previous work in this domain, such as FEVER, which is focused on the simpler task of claim verification, our resource is the first enabling the training of end-to-end factuality evaluation systems, i.e., approaches that are able to assess the factuality of generic text in natural language.

We additionally devise a human annotation process to create a gold standard for benchmarking factuality evaluators and to validate the quality of the proposed data creation pipeline. LLM-OASIS enables two challenging tasks: *end-to-end factuality evaluation*, which tests the ability of models to verify factual accuracy in raw texts in natural language, and *evidence-based claim verification*, which focuses on assessing individual claims against provided evidence.

Our experiments reveal that open-weight LLMs, such as Phi-4 and Llama 3, fall short in the end-to-end task, only marginally surpassing the random baseline. In the same setting, even GPT-4o faces significant challenges, in both zero-shot and RAG settings, i.e., when provided with supporting evidence from Wikipedia, only achieving 60% and 68% accuracy, respectively. This underscores the difficulty of the proposed benchmark and its potential to drive progress in factuality evaluation. Furthermore, thanks to LLM-OASIS, we design a novel baseline for end-to-end factuality evaluation, which consists of a pipeline of smaller, specialized models trained on three subtasks, namely, claim extraction, evidence retrieval, and claim verification. Our approach demonstrates competitive or even superior performance to GPT-4o, showcasing the potential of smaller LMs fine-tuned on specific data for factuality evaluation.

Looking forward, we plan to expand LLM-OASIS to incorporate data from diverse domains and multiple languages, enhancing its utility and applicability. With the aim of fostering research in factuality evaluation, we release our resource at `https://github.com/Babelscape/LLM-Oasis`.

9. Challenges and Discussion

In this section, we reflect on a number of relevant aspects emerging from the design and construction of our benchmark, including open challenges, modeling decisions, and future directions for improving factuality evaluation.

Quality of the Silver Data. Even if we manually validate a subset of the data, our resource is LLM-generated. The utilized model, namely, GPT-4, achieved very high performance in the various generation tasks (cf. Section 4), but the introduced errors, even if they are few, may affect the quality of the training dataset. For this reason, we suggest leveraging the automatically generated portion of our resource for developing systems, rather than benchmarking, for which we direct to our gold standard benchmark.

Multi-step Prompting. A potential limitation of our dataset generation process is the adoption of a unified prompt in a single API call to GPT-4, rather than employing a multi-step prompting strategy. While multi-step prompting could, in principle, improve the performance of individual data generation stages (e.g., claim extraction, falsification, and paraphrasing), we opted for a single-prompt approach primarily due to budget constraints. Using GPT-4, which is a paid model, a multi-step strategy would have significantly increased the number of API calls, as each step would have required re-sending the entire Wikipedia passage. This would have resulted in approximately four times the cost, due to higher input token usage across multiple calls. During development, we qualitatively compared both approaches and observed no substantial improvement in the quality of the generated outputs. Therefore, we adopted the single-prompt strategy as a more efficient and cost-effective solution, without compromising the integrity of the generated data.

Reliance on Wikipedia. Additionally, LLM-OASIS is limited to Wikipedia as the source of factual information. This restricts the diversity of the dataset and may not offer coverage of other kinds of texts, such as scientific articles or news. We also note that our end-to-end evaluation task may require periodic updates as Wikipedia evolves: While unfactual texts generally remain valid over time, factual ones could become outdated. To maintain long-term relevance, future iterations of the dataset will incorporate updated Wikipedia dumps, ensuring that the benchmark remains challenging as LLMs get exposed to updated knowledge.

Rarity of the Falsified Facts. We acknowledge that rare or less frequent facts—typically referred to as the long tail—represent a known challenge for factuality evaluation systems, including ours. While our modular pipeline does not rely on domain-specific priors and is, in principle, extendable to less frequent content, performance may degrade if relevant knowledge is underrepresented in the training data of each component.

That said, we argue that LLM-Oasis already provides a challenging setting, even without explicitly targeting long-tail phenomena. First, we note that state-of-the-art open-weight LLMs, even in Few-Shot settings, do not surpass 61.7% accuracy on our benchmark—highlighting the inherent difficulty of the task. This suggests that factuality evaluation remains an open problem even when models are tested on commonly known entities.

Second, although we constructed our dataset from the top 100k most-viewed Wikipedia pages to ensure quality and consistency, popularity at the page level does not imply that all facts within that page are well-known or frequently mentioned elsewhere. Indeed, popular entries often contain historical nuances, or lesser-known anecdotes that are less likely to be memorized or represented in LLMs' training data.

Finally, assessing factual rarity is not trivial. The frequency of a fact is hard to quantify reliably, as it can be expressed in many ways across Wikipedia. For this reason, we believe that increasing the dataset's coverage of long-tail content remains a valuable future direction, but our current benchmark already captures a wide factual spectrum and exposes significant limitations in existing systems.

Scope of our Resource. Our work specifically targets the evaluation of factuality in LLM-generated outputs, aiming to detect unintended hallucinations resulting from limitations in reasoning or retrieval—rather than adversarial manipulations intended to deceive, which are common in human-made misinformation (e.g., in Politifact [Alhindi,

Petridis, and Muresan 2018] or Snopes [Vo and Lee 2020]). These two tasks are fundamentally different: While both may involve similar technologies, their goals, linguistic characteristics, and difficulty profiles diverge significantly. In particular, LLM hallucinations tend to be subtle, plausible, and embedded in factual context, often without stylistic or rhetorical markers of deception.

Multilinguality. Finally, our analysis and experiments are limited to English-only data, which constrains the applicability of our findings to other languages. In future iterations, we plan to extend LLM-OASIS to include a broader range of domains and languages, in order to better support multilingual and cross-domain factuality evaluation.

Appendix A. Examples of Generated Texts

We present several examples derived from our dataset to show the model's capability in generating both factual and unfactual texts. These examples show how our pipeline produces paraphrased versions of original texts and introduces subtle yet critical factual inaccuracies.

Appendix A.1 Example 1

Original Text:

Albert Einstein was a German-born theoretical physicist who developed the theory of relativity, one of the two pillars of modern physics. His work is also known for its influence on the philosophy of science. Einstein is best known for his mass–energy equivalence formula $E = mc^2$, which has been dubbed "the world's most famous equation".

Extracted Claims:

1. Albert Einstein was a German-born theoretical physicist.
2. He developed the theory of relativity.
3. The theory of relativity is one of the two pillars of modern physics.
4. Einstein's work influenced the philosophy of science.
5. He is best known for his mass–energy equivalence formula $E = mc^2$.
6. The formula $E = mc^2$ is dubbed "the world's most famous equation".

Factual Text:

Albert Einstein, originally from Germany, was a theoretical physicist who formulated the theory of relativity, a cornerstone of modern physics. His contributions significantly impacted the philosophy of science. The mass–energy equivalence equation $E = mc^2$, which he is most famous for, is often called "the world's most famous equation".

Falsified Claim:

He developed the theory of quantum mechanics.

Unfactual Text:

Albert Einstein, originally from Germany, was a theoretical physicist who formulated the theory of quantum mechanics, a cornerstone of modern physics. His contributions significantly impacted the philosophy of science. The mass–energy equivalence equation $E = mc^2$, which he is most famous for, is often called "the world's most famous equation".

Appendix A.2 Example 2

Original Text:

The Amazon Rainforest, also known as Amazonia, is a moist broadleaf forest in the Amazon biome that covers most of the Amazon basin of South America. This region includes territory belonging to nine nations. The majority of the forest is contained within Brazil, with 60% of the rainforest.

Extracted Claims:

1. The Amazon Rainforest is also known as Amazonia.
2. It is a moist broadleaf forest in the Amazon biome.
3. The Amazon Rainforest covers most of the Amazon basin of South America.
4. The region includes territory belonging to nine nations.
5. The majority of the forest is contained within Brazil.
6. Brazil contains 60% of the rainforest.

Factual Text:

Amazonia, widely known as the Amazon Rainforest, is a damp broadleaf forest located within the Amazon biome, covering a significant portion of the Amazon basin in South America. This vast region spans across nine countries, with Brazil housing 60% of the rainforest.

Falsified Claim:

The majority of the forest is contained within Peru.

Unfactual Text:

Amazonia, widely known as the Amazon Rainforest, is a damp broadleaf forest located within the Amazon biome, covering a significant portion of the Amazon basin in South America. This vast region spans across nine countries, with Peru housing 60% of the rainforest

These examples show the effectiveness of the model in creating pairs of factual and unfactual texts. The alterations are subtle, ensuring that the generated unfactual texts are challenging for both human annotators and automated systems to detect, thus providing a robust testbed for evaluating the factual accuracy of LLM-generated content.

Appendix B. Prompts for End-to-End Factuality Evaluation

To accomplish the task of end-to-end factuality evaluation, we use different strategies depending on the language model being used. For models like Llama, which supports a system prompt, we set specific instructions as the system message. For models like Mistral, which do not support a system prompt, we include the instructions at the beginning of the text. In our experiments, we set the temperature to 0.0 to guarantee consistency of the results across different runs.

The prompts used for factuality evaluation in Zero-Shot and RAG are displayed in Tables B.1 and B.2. In the latter setting, we prompt all the LLMs with the same pieces of evidence retrieved and used by our NLI module (cf. Section 7.1). Concerning the Explain-then-Answer paradigm, we expand the set of instructions with the following recommendation:

Motivate your response with an explanation and then reply with "Factual" or "Not Factual"
Output format:
EXPLANATION: explanation
LABEL: label, i.e., "Factual" or "Not Factual"

Table B.1
Zero-Shot Prompt for factuality evaluation of a text.

Determine whether the given text is factual or not.
1. Read the input text.
2. Evaluate the factual accuracy of the input text based on your training data and knowledge.
3. If the input text is factually-accurate, i.e. supported by known information, respond with "Factual"
4. Respond with "Not Factual" if the input text contains even a single inaccuracy.
5. Just reply with "Factual" or "Not Factual", do not generate any additional text to the answer.

Table B.2
Prompt for factuality evaluation in RAG setting.

Determine whether the given text is factual or not using the provided evidence. If the information is not present in the evidence, rely on prior knowledge.
1. Read the input text.
2. Read the evidence if provided.
3. Assess whether the input text is factual based on the evidence if present.
4. If the evidence is not provided or is insufficient, use your prior knowledge to determine the factuality.
5. Respond with "Not Factual" if the input text contains even a single inaccuracy.
6. If the evidence is not related to the text to verify, rely on your prior knowledge to provide the answer.
7. Just reply with "Factual" or "Not Factual", do not generate any additional text to the answer.

Appendix C. Further Details on Evidence Retriever Module

In this section, we present further details about our evidence retrieval model. To assess the contribution of different components, we performed an ablation study on the retrieval module. All models were trained using the same hyperparameters described in Section 5.2. Results were computed on the corpus $\mathcal{D}$, which contains 2.5 million passages, and evaluated on the validation split of the dataset. After training, our best model achieved a recall at $k = 30$ (R@30) of 0.95.

We utilized the $E5_{base}$ model (Wang et al. 2022), built upon the bert-base-uncased (Devlin et al. 2019) architecture, with weights initialized from Sentence-Transformers (Reimers and Gurevych 2019). As part of our ablation study, we also trained the *bert-base-uncased* model with the same hyperparameters, achieving a recall of 0.85. This significant performance drop compared to the fully fine-tuned *E*5 demonstrates the effectiveness of the additional pretraining done in *E*5.

Additionally, we experimented with other architectures from the *E*5 family. The $E5_{small}$ model obtained a recall of 0.75, whereas the $E5_{large}$ model slightly outperformed $E5_{base}$, achieving a recall of 0.96. Despite the marginal 1% performance gain, we opted to use the $E5_{base}$ model in our final system due to the substantial increase in computational resources and training time required by the $E5_{large}$ model, which did not justify the small performance improvement.

The results of all models tested during the ablation study are summarized in Table C.1, confirming the robustness and efficiency of the $E5_{base}$ model for claim retrieval, balancing performance with computational cost.

Table C.1
Performance of different models on claim retrieval task.

Model	Recall@30
$E5_{base}$ (without fine-tuning)	0.52
$E5_{base}$	0.95
bert-base-uncased	0.85
$E5_{small}$	0.75
$E5_{large}$	0.96

Appendix D. Details About the Utilized LLMs

In this section, we detail the models we used in this work. For the generation of our dataset, we used GPT-4 API, with an approximate cost of $2,000. As for the open-source models we utilized for the LLM baselines, we used the instruction tuned versions of Mistral[11] and LLama 3[12] publicly available on HuggingFace. For the benchmark evaluation, we utilized the OpenAI API. Specifically, for GPT-4o, we used the model *GPT-4o-2024-05-13*. For the claim-extractor, we used the pre-trained T5-base[13] as our base model.

11 https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3.
12 https://huggingface.co/meta-Llama/Meta-Llama-3-8B-Instruct.
13 https://huggingface.co/google-t5/t5-base.

Appendix E. Annotation Guidelines

In this section, we illustrate the annotation guidelines utilized. Annotators are asked to perform four different tasks related to factuality evaluation. For each task, annotators receive specific guidelines which we report in what follows. As a standard guideline for all tasks, annotators are required to discard instances entirely or partially written in a language other than English. Furthermore, in case of pronominal ambiguity occurring in a given claim, if the human annotator cannot determine, with a high degree of confidence, the noun to which a given pronoun refers, that claim is discarded. Annotators are required to participate in joint sessions to resolve challenges and collaboratively develop agreed-upon solutions.

Appendix E.1 Task 1: Claim Extraction

Task Description. In this step, you will verify if claims extracted from a given text are accurately represented within the original text. You will receive a 5-sentence passage extracted from Wikipedia, along with corresponding claims pre-extracted by a language model. Note: a claim denotes an atomic fact, that is, an elementary information unit found in a text, that does not require further subdivision, and that can be checked for its truthfulness.

Annotation Format. You will be provided with a TSV (Tab-Separated Values) file containing three columns:

- Column 1: Identifier (either "text" or "claim id").
- Column 2: Text or claim.
- Column 3: Empty. You have to fill in this column.

Annotation Procedure.

1. Read the original text and claims thoroughly.
2. For each claim, determine if it is accurately represented in the original text.
3. Place a "v" in the third column if the claim is present in the original text, otherwise mark it with an "x".

Annotation Example. We report an example of annotated instance in Table E.1.

Additional Guidelines. Annotators are required to discard an entire instance, composed of the original text and the corresponding claims, if the original text is not grammatically correct, e.g., if it is syntactically ill-formed, or if it is semantically unclear, that is, if it is formulated in a way that the annotator cannot determine the meaning conveyed either by the entire text or one of its segments. Furthermore, annotators are required to discard sentences that cannot be considered as claims for the purposes of our work, e.g., sentences composed of a single word.

Table E.1
Example of annotated instance in task 1 (claim extraction).

Identifier	Text	Annotation
original_text	This type of meringue is safe to use without cooking. It will not deflate for a long while and can be either used for decoration on pie, or spread on a sheet or baked Alaska base and baked. Swiss meringue is whisked over a bain-marie to warm the egg whites, and then whisked steadily until it cools. This forms a dense, glossy marshmallow-like meringue. It is usually then baked.	
claim 1	Swiss meringue is safe to use without cooking.	v
claim 2	Swiss meringue will not deflate for a long while.	v
claim 3	Swiss meringue can be used for pie decoration or on a baked Alaska base.	v
claim 4	Swiss meringue is whisked over a bain-marie to warm the egg whites.	v
claim 5	Swiss meringue is then whisked steadily until it cools.	v
claim 6	Swiss meringue forms a dense, glossy, marshmallow-like texture.	v
claim 7	Swiss meringue is usually baked after preparation.	v
claim 8	Swiss meringue can be mixed with vanilla or chocolate to add flavor.	x

Appendix E.2 Task 2: Claim Falsification

Task Description. In this step, you will identify whether a given claim has been altered to introduce unfactual information.

Annotation Format. You will receive a pair of claims, where the second claim is an unfactual version of the first

- Column 1: The original claim.
- Column 2: The unfactual claim.
- Column 3: Empty. You have to fill in this column.

Annotation Procedure.

1. Compare the two claims provided.
2. Determine if the unfactual claim introduces new, untrue information compared to the original claim.
3. Mark column 3 with “v” if unfactual information is introduced, otherwise mark it with “x”.

Annotation Example. We report an example of annotated instance in Table E.2.

Additional Guidelines. If the original claim contains a word that is replaced with its hyponym in the candidate nonfactual claim, while the overall meaning of both claims

Table E.2
Example of annotated instance in task 3 (claim falsification).

Identifier	Claim	Annotation
claim 1	The remix in Thank You track was Lassie Come Home.	v
claim 2	The remix in Thank You track was not Lassie Come Home.	x
claim 3	The Plateau served as a model for colonial capitals.	v
claim 4	The Plateau served as a model for other districts.	x
claim 5	Christoph Waltz replaced Billy Bob Thornton.	v
claim 6	Christoph Waltz replaced Brad Pitt.	x

remains unchanged also based on the annotator's world knowledge, then both claims are considered to be semantically equivalent.

Appendix E.3 Task 3: Factual Text Generation

Task Description. In this step, you will assess whether the semantics of claims is preserved in a paraphrased version of the text.

Annotation Format. You will receive a TSV file with four columns:

- Column 1: Identifier (either "paraphrase" or "claim id").
- Column 2: Text or claim.
- Column 3: Empty. You have to fill in this column.
- Column 4: Empty. You have to fill in this column.

Annotation Procedure.

1. Compare each claim with its representation in the paraphrased text.
2. Determine if its semantics is preserved.
    - If it is preserved (regardless of whether it is reported identically in the paraphrase), place a "v" in the third column.
    - Use "x" otherwise.
3. Determine if it is paraphrased.
    - If a claim is paraphrased, mark the fourth column with "v".
    - If not paraphrased (e.g. identical), mark column 4 with "x".

Table E.3
Example of annotated instance in task 3 (factual text generation).

Identifier	Text	Semantics Preserved	Paraphrased
claim 1	'Call Me by Your Name' leads Dorian Award nominations	v	v
claim 2	Gregg Kilday authored the article on 10 January 2018	v	v
claim 3	The Hollywood Reporter published the article	v	v
claim 4	Article was retrieved on 11 January 2018	x	x
claim 5	The Jameson Empire Awards occurred in 2014	v	v
paraphrase	'Call Me by Your Name' took the lead in Dorian Award nominations. The article, penned by Gregg Kilday, was published by The Hollywood Reporter on January 10, 2018, and accessed the following day. Meanwhile, The Jameson Empire Awards were held back in 2014.		

In other words:

- <"v", "v"> in the last two columns means that the semantics is preserved and the text is paraphrased (at least one word changed).
- <"x", "v"> in the last two columns means that the semantics is NOT preserved but the text is paraphrased.
- <"v", "x"> in the last two columns means that the semantics is preserved but the text is NOT paraphrased.
- <"v", "x"> in the last two columns means that the semantics is preserved but the text is NOT paraphrased.
- <"x", "x"> in the last two columns means that neither the semantics is preserved nor the text is paraphrased (e.g., the claim is omitted).

Annotation Example. We report an example of annotated instance in Table E.3.

Additional Guidelines. If a nearly identical date appears in the factual text and in one claim, annotators should proceed as follows. If the date in the factual text includes the month and year, while the claim specifies the day, month, and year, even if the month and year in the claim coincide with those in the factual text, the semantics conveyed by the claim is considered to be different from that of the factual text.

Appendix E.4 Task 4: Unfactual Text Generation

Task Description. In this step, you will assess whether all claims, including the unfactual one, are accurately reflected in a generated unfactual text.

Annotation Format. You will receive all claims paired with the generated unfactual text.

- Column 1: Identifier (either "claim id", or "unfactual_text").
- Column 2: Text or claim.
- Column 3: Empty. You have to fill in this column.

Annotation Procedure.

- Review the generated unfactual text along with all claims provided.
- Determine if all claims are correctly reported in the text (i.e. the factual claims should remain factual and the unfactual claims should be unfactual). Ensure that the text in the "unfactual_text" field is not modified by the language model to be compliant with the unfactual claim. Paraphrasing in claims is allowed, you should focus on semantics.
- Mark column 3 with "v" if the unfactual text corresponds to the claims accurately, otherwise mark it with "x".

Annotation Example. We report an example of annotated instance in Table E.4.

Table E.4
Example of annotated instance in task 3 (unfactual text generation).

Identifier	Text	Annotation
original_text	In response to crisis, Ottoman statesmen adopted a compliant policy. Abdulmejid's inability to handle the situation heightened discontent regarding the Edict of Tanzimat. **To enhance European influence, opponents schemed to dethrone Abdulmejid for Abdulaziz.** The planned Kuleli Foundation revolt was thwarted before it could begin on 14 September 1859. Meanwhile, the financial crisis deepened as burdensome foreign debts strained the treasury.	
claim 1	Ottoman statesmen panicked and adopted a policy fulfilling every wish.	v
claim 2	Abdulmejid failed to prevent the situation, increasing dissatisfaction with the Edict of Tanzimat.	v
claim 3	Opponents planned to replace Abdulmejid with Abdulaziz to enhance European dominance.	v
claim 4	The Kuleli Foundation revolt was suppressed before starting on 14 September 1859.	v
claim 5	The financial situation worsened, and foreign debts burdened the treasury.	v
claim 6	To enhance European influence, opponents schemed to dethrone Abdulmejid for Abdulaziz.	x

Appendix F. Annotator Disagreements and Guideline Updates

This appendix provides concrete examples of the most frequent annotator disagreements encountered during the construction of LLM-OASIS, together with the

clarifications that were added to the annotation guidelines. Throughout, we refer to the three annotation labels used in the project:

- **V** (valid paraphrase / factually correct claim)
- **X** (invalid paraphrase / factually incorrect or incomplete claim)
- **D** (discarded instance: malformed, uninterpretable, or out of scope)

Appendix F.1 Paraphrase Coverage

Original claim:

"*Paul Revere designed the flat lid **with a ridge** for holding coals.*"

Paraphrase produced by the model:

"*In addition to coining the flat-lid design for holding coals, Paul Revere introduced legs to pots.*"

Two annotators initially labelled this as:

- **Annotator 1**: **X**—the paraphrase omits the phrase **with a ridge**.
- **Annotator 2**: **V**—major meaning preserved; one lexical change suffices.

Resolution and Guideline Update. A paraphrase must preserve *all obligatory semantic complements*. Missing the ridge detail changes the design specification; therefore the correct label is **X**.

Near-Duplicate Predicates with Modal Uncertainty

Pair:
"Holloway possibly **disappears** into blackness."
"Holloway possibly **finds escape** into blackness."

Annotators debated whether different verbs (*disappear* vs. *find escape*) preserved the same event.

Resolution and Guideline Update. Given the modal "possibly," both predicates entail an uncertain departure into darkness; the pair is therefore **V**. The updated guidelines clarify that when two predicates convey the same core event and modality, minor differences in verb choice are acceptable.

Mixed-Language or Fragmentary Instances

Guidelines instruct annotators to discard instances containing code-switching or non-Latin scripts not handled by the pipeline.

Policy for Noisy Lists

Long, unstructured lists of album titles, dates, or bibliographic metadata are difficult to map to atomic propositions. We require annotators to mark these as **D**.

Acknowledgments

We gratefully acknowledge the support of the PNRR MUR project PE0000013-FAIR. We also gratefully acknowledge the CREATIVE project (CRoss-modal understanding and gEnerATIon of Visual and tExtual content), which is funded by the MUR Progetti di Ricerca di Rilevante Interesse Nazionale programme (PRIN 2020). Alessandro Scirè, Andrei Stefan Bejgu, Simone Tedeschi, and Karim Ghonim have conducted part of this work during their enrollment in the Italian National Doctorate in Artificial Intelligence at Sapienza University of Rome. We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, hosted by CINECA (Italy).

References

Abdin, Marah, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*. https://arxiv.org/abs/2412.08905

Alhindi, Tariq, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90. https://doi.org/10.18653/v1/W18-5513

Alves, Duarte M., José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*. https://doi.org/10.48550/arXiv.2402.17733

Augenstein, Isabelle, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP–IJCNLP)*, pages 4677–4682. https://doi.org/10.18653/v1/D19-1475

Brodersen, Kay Henning, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. 2010. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124. https://doi.org/10.1109/ICPR.2010.764

Chen, Shiqi, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. FELM: Benchmarking factuality evaluation of large language models. *Advances in Neural Processing Systems (NeurIPS 2023)*. https://doi.org/10.48550/arXiv.2310.00741

Chen, Yanran and Steffen Eger. 2023. MENLI: Robust evaluation metrics from natural language inference. *Transactions of the Association for Computational Linguistics*, 11:804–825. https://doi.org/10.1162/tacl_a_00576

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. https://doi.org/10.18653/v1/N19-1423

Fabbri, Alexander R., Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409. https://doi.org/10.1162/tacl_a_00373

Gillick, Daniel, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537. https://doi.org/10.18653/v1/K19-1049

Goyal, Tanya, Junyi Li, and Greg Durrett. 2022. News summarization and evaluation in the era of GPT-3. *arXiv preprint arXiv:2209.12356*. https://doi.org/10.48550/arXiv.2209.12356

Grattafiori, Aaron, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*. https://arxiv.org/abs/2407.21783

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*. https://openreview.net/forum?id=XPZIaotutsD

Hsieh, Cheng Ping, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*. https://arxiv.org/abs/2404.06654

Hu, Hai, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S. Moss. 2020. OCNLI: Original Chinese natural language inference. *EMNLP* (Findings), 2020: 3512–3526. https://doi.org/10.18653/v1/2020.findings-emnlp.314

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. https://arxiv.org/abs/2310.06825

Jiang, Yichen, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460. https://doi.org/10.18653/v1/2020.findings-emnlp.309

Kamalloo, Ehsan, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606. https://doi.org/10.18653/v1/2023.acl-long.307

Kamradt, Gregory. 2023. Needle in a Haystack – Pressure testing LLMs. https://github.com/gkamradt/LLMTest_NeedleInAHaystack

Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

Laban, Philippe, Alexander R. Fabbri, Caiming Xiong, and Chien-Sheng Wu. 2024. Summary of a haystack: A challenge to long-context LLMs and RAG systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9885–9903. https://doi.org/10.18653/v1/2024.emnlp-main.552

Laban, Philippe, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177. https://doi.org/10.1162/tacl_a_00453

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474. https://arxiv.org/abs/2005.11401

Liu, Yixin, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170. https://doi.org/10.18653/v1/2023.acl-long.228

Loshchilov, Ilya and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

McNemar, Quinn. 1947. Note on the sampling error of the difference between

correlated proportions or percentages. *Psychometrika*, 12(2):153–157. https://doi.org/10.1007/bf02295996 PubMed: 20254758

Microsoft, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. 2025. Phi-4-Mini technical report: Compact yet powerful multimodal language models via mixture-of-loRAs. https://arxiv.org/abs/2503.01743

Min, Sewon, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100. https://doi.org/10.18653/v1/2023.emnlp-main.741

Muhlgay, Dor, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. Generating benchmarks for factuality evaluation of language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 49–66. https://doi.org/10.18653/v1/2024.eacl-long.4

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*. https://doi.org/10.48550/arXiv.2303.08774

Pagnoni, Artidoro, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. https://doi.org/10.18653/v1/2021.naacl-main.383

Parrish, Alicia, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901. https://doi.org/10.18653/v1/2021.findings-emnlp.421

Pu, Xiao, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*. https://doi.org/10.48550/arXiv.2309.09558

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*. https://arxiv.org/abs/2412.15115

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21: 140:1–140:67. (2020).

Rasool, Zafaryab, Stefanus Kurniawan, Sherwin Balugo, Scott Barnett, Rajesh Vasa, Courtney Chesser, Benjamin M. Hampstead, Sylvie Belleville, Kon Mouzakis, and Alex Bahar-Fuchs. 2024. Evaluating LLMs on document-based QA: Exact answer selection and numerical extraction using CogTale dataset. *Natural Language Processing Journal*, 8:100083. https://doi.org/10.1016/j.nlp.2024.100083

Reimers, Nils and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. https://doi.org/10.18653/v1/D19-1410

Scirè, Alessandro, Karim Ghonim, and Roberto Navigli. 2024. FENICE: Factuality evaluation of summarization based on natural language inference and claim extraction. *ACL* (Findings). 2024: 14148–14161. https://doi.org/10.18653/v1/2024.findings-acl.841

Shazeer, Noam and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *ICML*, 2018: 4603–4611.

Tam, Derek, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2022. Evaluating the factual consistency of large language models

through summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255. https://doi.org/10.18653/v1/2023.findings-acl.322

Tang, Liyan, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644. https://doi.org/10.18653/v1/2023.acl-long.650

Tang, Liyan, Igor Shalyminov, Amy Wing mei Wong, Jon Burnsky, Jake W. Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. *NAACL-HLT* 2024: 4455–4480. https://doi.org/10.18653/v1/2024.naacl-long.251

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: A large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819. https://doi.org/10.18653/v1/N18-1074

Tonmoy, S. M. Towhidul Islam, S. M. Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*. https://doi.org/10.48550/arXiv.2401.01313

Vo, Nguyen and Kyumin Lee. 2020. Where are the facts? Searching for fact-checked information to alleviate the spread of fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731. https://doi.org/10.18653/v1/2020.emnlp-main.621

Wadden, David, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction? Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609

Wang, Hengyi, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. 2024. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3221–3241. https://doi.org/10.18653/v1/2025.naacl-long.166

Wang, Liang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. https://doi.org/10.48550/arXiv.2212.03533

Wang, Longyue, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661. https://doi.org/10.18653/v1/2023.emnlp-main.1036

Wang, Yiming, Zhuosheng Zhang, and Rui Wang. 2023. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665. https://doi.org/10.18653/v1/2023.acl-long.482

Yih, Wen-tau, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256.

Zha, Yuheng, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348. https://doi.org/10.18653/v1/2023.acl-long.634

Zhang, Biao, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case

study. In *International Conference on Machine Learning*, pages 41092–41110. PMLR.

Zhang, Shiyue and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632. https://doi.org/10.18653/v1/2021.emnlp-main.531

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. *ICLR 2020*. https://arxiv.org/abs/1904.09675

Zhang, Tianyi, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori Hashimoto. 2023. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57. https://doi.org/10.1162/tacl_a_00632

Zuo, Jingwei, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. 2024. Falcon Mamba: The first competitive attention-free 7B language model. *arXiv preprint arXiv:2410.05355*. https://arxiv.org/abs/2410.05355