FactSearch: An Interactive Agentic Fact Search System for Verifying Large Language Model Outputs

Meng Fang
University of Liverpool
Meng.Fang@liverpool.ac.uk

Harry Mackenzie
University of Liverpool
Harry.Mackenzie@liverpool.ac.uk

Abstract

Large language models (LLMs) frequently generate factually incorrect or unverifiable statements, motivating tool-augmented verification systems that combine model reasoning with external evidence retrieval. For factuality evaluation to be scientifically reliable, verification pipelines must be controllable and reproducible: retrieval configuration and reasoning behaviour should be explicitly configurable and stable across runs. In practice, many existing systems depend on commercial search APIs whose ranking policies and retrieval behaviours are opaque and externally controlled, introducing uncontrolled variability into evaluation. This makes it difficult to disentangle reasoning errors from retrieval effects. We present FactSearch, a reproducibility-oriented agentic fact search system for claim-level factuality verification, built on a locally aggregated open-source search infrastructure. FactSearch follows an agentic verification workflow: it decomposes model outputs into atomic factual claims, generates targeted search queries, retrieves supporting evidence via a self-hosted meta-search engine, and performs modular verification within a fully configurable pipeline. By treating retrieval infrastructure as a first-class component, the system enables systematic analysis of retrieval–reasoning interactions. An interactive web interface supports transparent inspection and practical deployment. The project is available at https://factsearch.github.io.

1 Introduction

Large language models (LLMs) frequently generate factually incorrect or unverifiable statements, motivating the development of agentic, tool-augmented verification systems that combine model reasoning with external evidence retrieval (Augenstein et al., 2024). Recent work has proposed a range of automatic fact-checking frameworks for evaluating LLM outputs, including RARR (Gao et al., 2023), FactScore (Min et al., 2023), FacTool (Chern et al., 2023), CoVe (Dhuliawala et al., 2024), and OpenFactCheck (Wang et al., 2025). These systems demonstrate that retrieval-augmented reasoning can substantially improve factuality assessment.

However, reliable factuality evaluation requires that verification pipelines be controllable and reproducible. Retrieval depth, ranking behavior, evidence selection, and reasoning configuration should be explicitly configurable and stable across runs in order to enable systematic analysis. In practice, most existing verification systems depend on commercial search APIs whose ranking policies, retrieval coverage, and system behaviors are opaque and externally controlled (Wang et al., 2025; Chern et al., 2023). Search results may vary across time, geographic location, rate limits, or API versions, and per-query costs constrain large-scale experimentation. As a result, retrieval infrastructure becomes a hidden confounding factor in factuality evaluation, making it difficult to disentangle reasoning failures from search variability and limiting cumulative scientific progress.

To this end, we present FactSearch, an agentic fact search system for claim-level factuality verification built on locally aggregated open-source search infrastructure. FactSearch decomposes model responses into atomic factual claims (Min et al., 2023; Chern et al., 2023), retrieves supporting evidence via a self-hosted meta-search engine, and performs claim-level verification through a modular reasoning pipeline. By eliminating reliance on commercial search APIs, the system enables controllable retrieval configurations resulting in improved reproducibility between runs, in addition to zero marginal retrieval cost.

The system supports both lightweight open-source local LLMs for fully offline deployment and high-performance API-based models for flexible evaluation. An interactive web-based interface allows users to inspect verification decisions at the

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 367–373

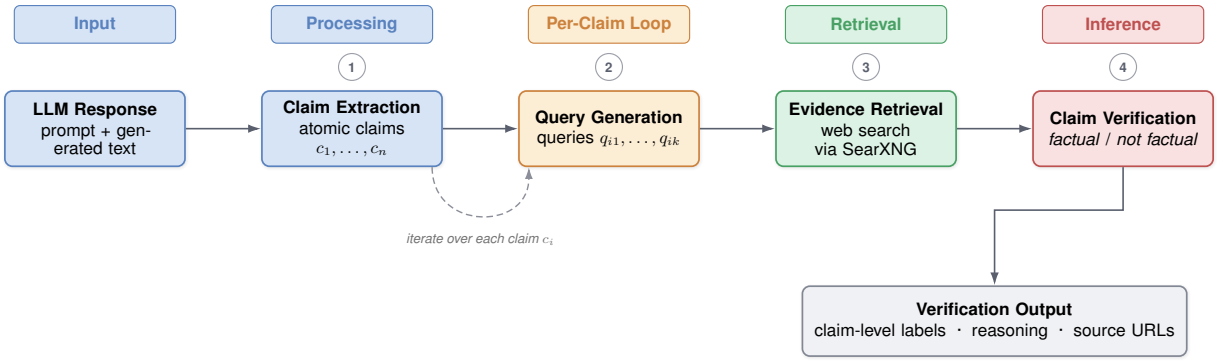

Figure 1: Overview of the agentic claim verification workflow. Given an LLM response, the system (1) extracts atomic claims, (2) generates targeted search queries per claim, (3) retrieves evidence via a local meta-search procedure, and (4) produces per-claim classifications, outputted together with natural-language reasoning and source URLs.

claim level, examine supporting evidence, and trace reasoning outputs in a transparent manner.

Through this demonstration, we show how factuality inspection can be made controllable and accessible — facilitating the development of practical verification workflows and systematic research on LLM factual reliability.

In summary, we have made the following contributions:

1. An agentic fact search and verification pipeline with locally-deployed open-source search infrastructure at zero retrieval cost;

2. Support for both local and API-based LLMs — FactSearch enables fully local execution using lightweight open models while also supporting high-performance verification with flagship proprietary models;

3. Claim-level verification decisions, including traceable supporting evidence through a web-based GUI designed to be accessible to non-technical users.

2 System Overview

FactSearch is designed as an interactive, web-based agentic fact search system that enables claim-level factuality inspection of LLM-generated responses. Users can submit a human-made prompt and corresponding LLM-generated response, and the system returns a comprehensive claim-level and response-level factuality assessment.

At a high level, the verification process follows four key stages following user input, as illustrated in Figure 1. First, the system decomposes the LLM response into atomic factual claims (Min et al., 2023) suitable for independent verification. Second, for each claim, two targeted search queries are generated. Thirdly, search queries are passed to a local instance of the open-source meta-search engine SearXNG (SearXNG Developers, 2023), from which FactSearch parses evidence snippets to support or refute claims. Finally, each claim is assessed against the retrieved evidence, producing a factuality label and natural language reasoning which is then displayed to the user alongside evidence snippets and associated source links to enable easy access to retrieved information for further in-depth manual examination.

2.1 Interactive User Interface

FactSearch is primarily designed to be run via the web interface shown in Figure 2, although it can also be executed from the command line. The web-based interface is designed for interactive exploration of results without requiring programming knowledge or technical expertise. Users can submit questions and LLM-generated responses, configure a language model that includes local Qwen3 (Yang et al., 2025) or API-based GPT models, trigger the pipeline, and view detailed factuality assessments. Each claim can be expanded to reveal in-depth reasoning, Source URLs (and accompanying page summaries), as well as viewing search queries used in the evidence retrieval step. Additionally, completed verification runs can be exported in .txt format for documentation or further analysis. Figure 3 shows the results interface after the verification pipeline has been executed, including an expanded claim-level view with retrieved evidence, reasoning, and the final factuality label. The FactSearch inputs used in this example are:

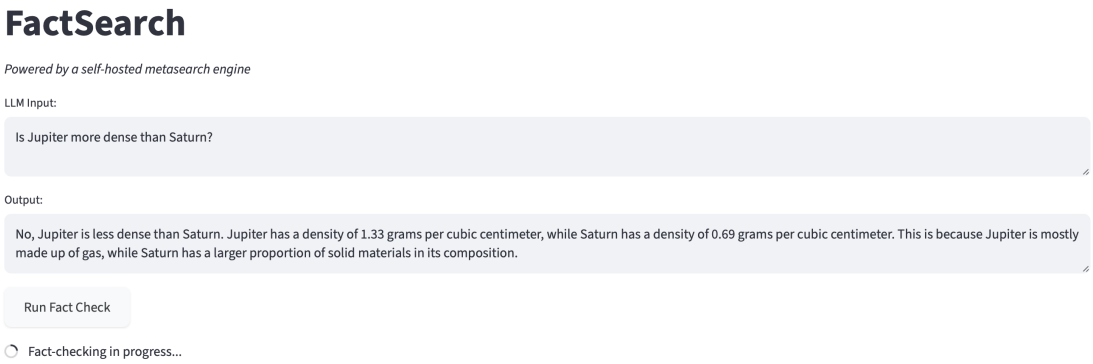

Figure 2: FactSearch input interface. Users provide an original prompt and the corresponding LLM-generated response for verification.

```
Prompt:
Is Jupiter more dense than Saturn?

Response:
No, Jupiter is less dense than Saturn. Jupiter
has a density of 1.33 grams per cubic
centimeter, while Saturn has a density of 0.69
grams per cubic centimeter. This is because
Jupiter is mostly made up of gas, while Saturn
has a larger proportion of solid materials in
its composition.
```

The interface abstracts system complexity while preserving transparency in the verification process by surfacing information sources utilised by the language model in the intermediate steps between the user input and the generation of a factuality label. This system enables practical and user-centred inspection of factual reliability in LLM outputs.

3 Architecture and Implementation

FactSearch adopts a three-tier architecture consisting of (i) a web-based front-end interface, (ii) a modular agentic verification pipeline implemented as a Python backend, and (iii) an integration layer managing communication between internal pipeline components and external services. This separation ensures that user interaction, reasoning logic, and evidence retrieval remain independently maintainable and extensible.

3.1 Modular Verification Pipeline

Each pipeline stage is implemented as an independent module with clearly defined inputs and outputs, allowing individual components to be replaced or extended without affecting the rest of the system.

Claim Extraction is performed by prompting a language model with the full LLM response and instructing it to decompose the text into a structured list of atomic, independently verifiable factual statements.

Query Generation takes each atomic claim as input and prompts the language model to produce two targeted search queries per claim. Generating two queries aims to reduce confirmation bias in the retrieved evidence pool, a known limitation of single-query retrieval approaches.

Evidence Retrieval dispatches generated queries asynchronously to a locally hosted SearXNG instance. The FactSearch default search configuration aggregates results across Brave, Qwant, DuckDuckGo, Mojeek, Startpage simultaneously; additionally, developers can modify this selection to remove or add additional search endpoints via the SearXNG configuration files in the codebase. Asynchronous dispatch across all claims in a batch substantially reduces end-to-end latency compared to sequential retrieval. Retrieved results are normalised into a standardised internal format of evidence snippets. [1]

Claim Verification receives each claim paired with its retrieved evidence snippets and constructs a structured prompt asking the language model to reason over the evidence and produce a binary factuality label — factual or not factual — alongside a natural language explanation of its decision. Labels are then aggregated across all claims to produce a response-level factuality summary, computed as the proportion of claims verified as being factual.

[1]The deployed SearXNG instance aggregates publicly accessible search endpoints, including Brave Search (https://search.brave.com), Qwant (https://www.qwant.com), DuckDuckGo (https://duckduckgo.com), Mojeek (https://www.mojeek.com), and Startpage (https://www.startpage.com).

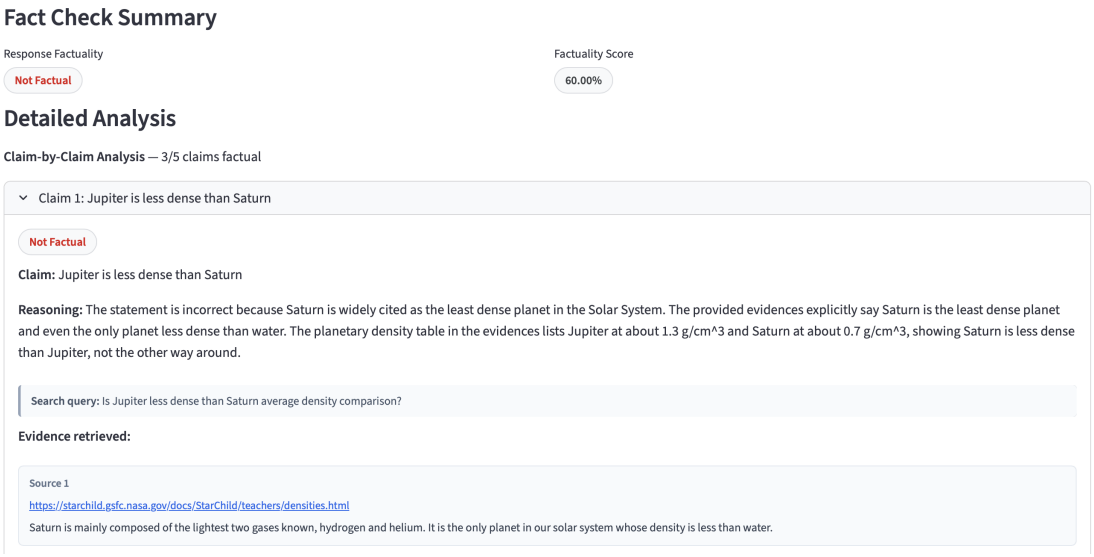

Figure 3: FactSearch results interface. The system reports an overall factuality score of 60% across five extracted claims and allows users to inspect each claim with its factuality label, reasoning, search query, and retrieved evidence. Expanded claim view window shows Claim 1 ("Jupiter is less dense than Saturn") has been flagged "Not Factual", alongside reasoning and sources below.

3.2 Integration Layer

The integration layer acts as a communication bus between the verification pipeline and its external dependencies. It handles three primary responsibilities: query dispatch and response parsing for the local SearXNG instance, evidence normalisation and filtering before passing results to the reasoning stage, and prompt handling and response parsing for language model API calls. This layer ensures that both the retrieval backend and the reasoning model can be swapped independently — for example substituting a local Qwen instance for a GPT-backed API without requiring changes to pipeline logic.

3.3 Live Demonstration Workflow

The live demo showcases how users can inspect the factual reliability of LLM outputs in real time via the FactSearch web interface.

During the demo, we illustrate how a multi-claim LLM response containing both correct and incorrect statements can be decomposed and verified at a fine-grained level. In particular, we focus on showcasing the detection of an unsupported over-claim hallucination (Li et al., 2024; Huang et al., 2025) in the medical domain. In this case, FactSearch flagged the following claim made by Claude Sonnet 4.6:

```
Claim 8: Vaccines are engineered to produce
better immunity than natural infection.
```

FactSearch correctly reasoned that whilst this is often the case, studies show mixed results and this claim is not universally true. The system reasoned that outcomes are heavily dependent on the pathogen, vaccine and outcomes measures, which was verified by source material retrieved from sources including World Health Organisation webpages, academic papers, and government health service sites.

4 Experiments

4.1 Empirical Evaluation

We evaluate our method on the KBQA evaluation set used in FacTool (Chern et al., 2023), containing real-world prompts sourced from various platforms and datasets. This evaluation set contains 50 question-response pairs across diverse domains, including history, science, and current events. Each response contains an average of 3.2 factual claims that require external evidence for verification. We report precision, recall, F1-score, and accuracy at both the claim and response levels.

4.1.1 Setup

We compare FactSearch against FacTool (Chern et al., 2023) using identical model configurations and evaluation protocol. The key architectural difference is the search backend: FacTool uses Google Serper API (commercial), while FactSearch uses self-hosted SearXNG configured

System	Claim-Level				Response-Level			
	Accuracy	Recall	Precision	F1-score	Accuracy	Recall	Precision	F1-score
Self-Check (0-shot)	77.25	84.75	85.23	84.99	54.00	95.65	50.00	65.67
Self-Check (3-shot)	79.83	85.88	87.36	86.61	64.00	52.17	63.16	57.14
FacTool (GPT-4)	84.12	85.31	93.21	89.09	78.00	60.87	**87.50**	71.79
FactSearch (GPT-4)	91.07	95.45	93.33	94.38	80.00	76.92	76.92	76.92
FactSearch (GPT-5.2)	**92.68**	**94.89**	**97.22**	**96.04**	**82.00**	**82.69**	75.00	**78.68**
Δ (Ours GPT-4 vs FT GPT-4)	+6.95	+10.14	+0.12	+5.29	+2.00	+16.05	−10.58	+5.13
Δ (Ours GPT-5.2 vs FT GPT-4)	+8.56	+9.58	+4.01	+6.95	+4.00	+21.82	−12.50	+6.89

Table 1: Performance comparison on KBQA benchmark. All metrics reported as percentages. Self-Check and FacTool (FT) results reproduced from Chern et al. (2023). Bold indicates best performance per column.

with Brave, Qwant, DuckDuckGo, Mojeek and Startpage. We evaluate FactSearch with both GPT-4 (OpenAI et al., 2024) and GPT-5.2 to assess performance across foundation model generations. For evaluation, we report results using GPT-based FactSearch to ensure stable and performance-comparable reasoning across experimental settings. While FactSearch supports both local and API-accessible models (including open-weight models deployed locally or accessed via unified API providers), local model performance can vary substantially depending on hardware capability. To ensure reproducibility and comparability of results, we therefore focus on a single standardized API-based model in this experiment.

We additionally compare against two Self-Check baselines that prompt the LLM to verify its own outputs: Self-Check (zero-shot CoT) and Self-Check (3-shot CoT). The difference between the two is that the zero-shot procedure provides the LLM with no extra information, whereas 3-shot provides the LLM with three examples of similar prompt scenarios from which it can derive information to support its decision.

Metrics are computed at two granularities: claim-level, where each extracted claim is independently classified as factual or non-factual, and response-level, where a response is classified as factual only if all constituent claims are verified as true.

4.1.2 Main Results

Table 1 presents our primary findings. FactSearch achieves competitive or superior performance across all metrics with both foundation models, with particularly strong improvements in recall at both granularities.

FactSearch outperforms tool-augmented and self-checking baselines. At the claim level, FactSearch reaches 94.38 F1 compared to FacTool's 89.09 (+5.29 points) and Self-Check (3-shot)'s 86.61 (+7.77 points). Using a more recent reasoning-optimised model further increases claim-level F1 to 96.04 (+6.95 points over FacTool). At the response level, FactSearch achieves 78.68 F1 compared to FacTool's 71.79 (+6.89 points), substantially outperforming both Self-Check variants.

Recall improvements indicate broader evidence coverage. FactSearch shows +10.14 claim-level recall and +16.05 response-level recall gains over FacTool. With the stronger model tier, response-level recall reaches 82.69% (+21.82 points over FacTool), suggesting that multi-engine aggregation combined with improved reasoning successfully retrieves relevant evidence for claims not well-covered by commercial search APIs. This is particularly critical for specialised topics and recent events where source diversity matters.

Precision trade-off reflects conservative verification. Response-level precision decreases relative to FacTool, reflecting stricter verification when sources conflict. Manual analysis reveals this stems from contradictory evidence across heterogeneous sources — for example when Wikipedia, news outlets, and academic databases disagree, FactSearch conservatively marks claims as non-factual. This is the safer failure mode for fact-checking applications, as false negatives (accepting fabricated claims) are more harmful than false positives (rejecting true claims). In such cases, it is generally preferable to flag a potentially refuted assertion, enabling further human investigation, especially in high-stakes decision making contexts. This behaviour can be adapted through prompt calibration depending on deployment requirements.

Model choice primarily affects calibration rather than capability. Comparing model tiers, the reasoning-optimised model improves claim-level precision (97.22% vs 93.33%) while maintain-

ing comparable recall. This indicates that performance gains arise mainly from improved evidence interpretation rather than retrieval differences, suggesting the FactSearch pipeline generalises across LLM backends.

Self-checking baselines show high false positive rates. Self-Check baselines exhibit substantially lower precision than tool-augmented approaches (50.00–63.16% response-level precision vs. FactSearch's 75.00–76.92%). LLMs tend to accept their own outputs as factual even when external evidence contradicts them, reinforcing the broader necessity of external verification tools.

4.1.3 Discussion

These results validate our core hypothesis: replacing commercial search APIs with locally aggregated open-source engines maintains or improves competitive verification performance while eliminating external dependencies. The precision tradeoff at the response level is an acceptable cost for substantial gains in recall, reproducibility, and deployment flexibility. Future work could explore (1) query refinement strategies to reduce conflicting evidence, (2) source reliability scoring to weight evidence appropriately, supporting an internal conflict resolution mechanism in the case of contradictory source material, (3) integration of specialised engines, datasets, or reasoning tools for domain-specific fact-checking.

Assuming an average of 5 claims per run, the pipeline incurs costs of approximately $0.018 per response under current OpenAI API pricing. If users choose the local variant powered by Qwen, then there is no cost.

5 Conclusion

We presented FactSearch, an interactive agentic fact search system for verifying LLM outputs. FactSearch decomposes model-generated responses into atomic factual claims, retrieves supporting evidence through locally aggregated open-source search infrastructure, and produces claim-level verification results with natural-language reasoning and source links. By exposing the retrieval and reasoning process through a web-based interface, the system supports transparent inspection of factuality decisions while reducing reliance on opaque commercial search APIs. Overall, FactSearch provides a practical and configurable platform for claim-level factuality verification and for studying the interaction between evidence retrieval and LLM-based reasoning.

Limitations

While FactSearch improves controllability and reproducibility of factuality verification pipelines, it does not eliminate all sources of uncertainty. Locally aggregated open-source search infrastructure may differ in coverage and ranking quality from commercial search engines, potentially affecting evidence completeness in domains with limited online visibility. Retrieval results remain dependent on the underlying web corpus and indexing configuration which change over time, thus it is expected that results will also vary, even if the system pipeline and reasoning model remain stable.

References

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. 2024. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, 6(8):852–863.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios. *Preprint*, arXiv:2307.13528.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and

open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.

Sewon Min, 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, Singapore. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

SearXNG Developers. 2023. Searxng. `https://github.com/searxng/searxng`. Open-source metasearch engine.

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi N. Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2025. OpenFactCheck: Building, benchmarking customized fact-checking systems and evaluating the factuality of claims and LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11399–11421, Abu Dhabi, UAE. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.