Hybrid Fact-Checking that Integrates Knowledge Graphs, Large Language Models, and Search-Based Retrieval Agents Improves Interpretable Claim Verification

Shaghayegh Kolli,* **Richard Rosenbaum,*** **Timo Cavelius,**
Lasse Strothe, Andrii Lata, Jana Diesner
Technical University Munich
(shaghayegh.kolli, richard.rosenbaum, timo.cavelius, lasse.strothe, andrii.lata, jana.diesner)@tum.de

Abstract

Large language models (LLMs) excel in generating fluent utterances but can lack reliable grounding in verified information. At the same time, knowledge-graph-based fact-checkers deliver precise and interpretable evidence, yet suffer from limited coverage or latency. By integrating LLMs with knowledge graphs and real-time search agents, we introduce a hybrid fact-checking approach that leverages the individual strengths of each component. Our system comprises three autonomous steps: 1) a Knowledge Graph (KG) Retrieval for rapid one-hop lookups in DBpedia, 2) an LM-based classification guided by a task-specific labeling prompt, producing outputs with internal rule-based logic, and 3) a Web Search Agent invoked only when KG coverage is insufficient. Our pipeline achieves an F1 score of 0.93 on the FEVER benchmark on the Supported/Refuted split without task-specific fine-tuning. To address *Not enough information* cases, we conduct a targeted reannotation study showing that our approach frequently uncovers valid evidence for claims originally labeled as *Not Enough Information (NEI)*, as confirmed by both expert annotators and LLM reviewers. With this paper, we present a modular, open-source fact-checking pipeline with fallback strategies and generalization across datasets.

1 Introduction

LLMs have advanced knowledge-intensive NLP tasks, but can generate ungrounded or hallucinated content, which undermines their reliability for automated fact checking (Brown et al., 2020). Knowledge-graph (KG)-based systems can provide explicit and transparent evidence through structured triples, but remain restricted due to their limited coverage and slower response times in open-domain scenarios (Jiang et al., 2020; Kim et al., 2023c). Recent work, such as Generate-on-Graph (Xu et al., 2024), treats LLMs as agents that generate missing KG triples, highlighting the potential of hybrid agent–KG reasoning frameworks.

This paper asks how a modular hybrid system can make fact-checking more reliable, and shows how a real-time pipeline improves both coverage and interpretability. We propose a real-time, agent-based pipeline (Figure 1) that integrates three autonomous steps: 1) a KG Retrieval for rapid one-hop lookups in DBpedia (Lehmann et al., 2015); 2) Language models to classify claims with a task-specific classification prompt using labels such as *Supported*, *Refuted*, or *Not Enough Information (NEI)* (Wei et al., 2022); and 3) a Web Search Agent invoked only when *NEI* is returned, rewriting the claim for on-demand retrieval (Lewis et al., 2020; Tan et al., 2023a).[1] While our system does not perform multi-hop reasoning, it remains modular across evidence types (structured KG evidence, unstructured web evidence), using retrieval to compensate for KG's single-hop limitations. This KG-first, web-adaptive strategy leverages the explainability of structured data while preserving open-domain coverage.

We evaluated our approach on the FEVER benchmark (Thorne et al., 2018), its adversarial extension FEVER 2.0 (Thorne et al., 2019), and, that is, the FactKG dataset (Kim et al., 2023c), achieving up to 0.93 F1 on FEVER and competitive results across all three without task-specific tuning. A focused *Not Enough Information* reannotation study shows that our pipeline can uncover valid evidence for claims labeled as unverifiable, a finding corroborated by both expert human annotators and LLM reviewers.

*These authors contributed equally.

[1]The implementation is open source and on GitHub at github.com/AndriiLata/aiFactCheck.

2 Related Work

Recent work in automated fact verification has focused on integrating structured knowledge sources, retrieval components, and LLMs to improve factual consistency and evidence grounding (Cao et al., 2025; Opsahl, 2024; Kim et al., 2023a). A growing number of systems have been combining neural models with KGs (Zhou et al., 2019; Kim et al., 2023c; Yao et al., 2019) or using web-based retrieval to expand coverage (Chen et al., 2024).

KG-based methods often rely on symbolic triples of the form `(subject, predicate, object)` as evidence. Prior studies have explored how to align natural language claims with KG facts using embedding models (Yao et al., 2019), graph-based reasoning (Zhou et al., 2019), semantic matching between claims and triples (Kim et al., 2023c), and LLMs (Kim et al., 2023b). While KGs offer structured and interpretable evidence, they can be limited by coverage and connectivity, particularly for claims requiring multi-hop or commonsense reasoning (Peng et al., 2023).

In contrast, web-based fact-checking systems retrieve textual evidence from open-domain sources. OE-Fact (Tan et al., 2023b), for instance, used LLMs to process retrieved snippets and generate decisions. Retrieval-augmented generation (RAG) (Lewis et al., 2020) has also been applied to fact verification tasks by conditioning generation on retrieved content. However, reliance on web-based, unstructured evidence raises concerns around evidence quality and verifiability.

There is a growing interest in agent-based and modular architectures for fact verification. The FIRE system (Xie et al., 2024) employs an iterative retrieval and verification process, where the model dynamically decides whether to retrieve more evidence or make a decision. Such approaches reflect a broader trend toward separating evidence retrieval from claim evaluation, often across different evidence sources or reasoning stages (Zhang et al., 2023). Finally, several studies have pointed out limitations with benchmark labels, particularly in the NEI category (Hu et al., 2024). Prior work has shown that some NEI claims can be verified with external evidence (Schuster et al., 2019), highlighting the role of human judgment in evaluating evidence sufficiency and the need for annotation guidelines that reflect real-world complexity.

To expand on this prior work, we developed a modular pipeline that combines structured KG evidence with an agent fallback retrieval and includes an interpretable classification component.

3 Methodology

Given a natural language claim C, our goal is to predict a label $Y \in \{\text{SUPPORTED}, \text{REFUTED}, \text{NEI}\}$, along with a small set of textual or structured evidence E^* that justifies the decision. Our system follows a two-stage architecture: a KG-first classification stage, followed by a fallback retrieval and reasoning stage using open-domain web evidence. The system does not require task-specific training and operates in a zero-shot inference mode. An overview of the pipeline is shown in Figure 1.

Stage 1: Knowledge Graph First Pass

Entity linking: We use ReFinED (Ayoola et al., 2022) to detect and disambiguate named-entity mentions in the claim c, mapping each surface span to a Wikidata Q-ID (Vrandečić and Krötzsch, 2014); if none is produced, we fall back to spaCy's EntityLinker (Honnibal et al., 2020). Resolved

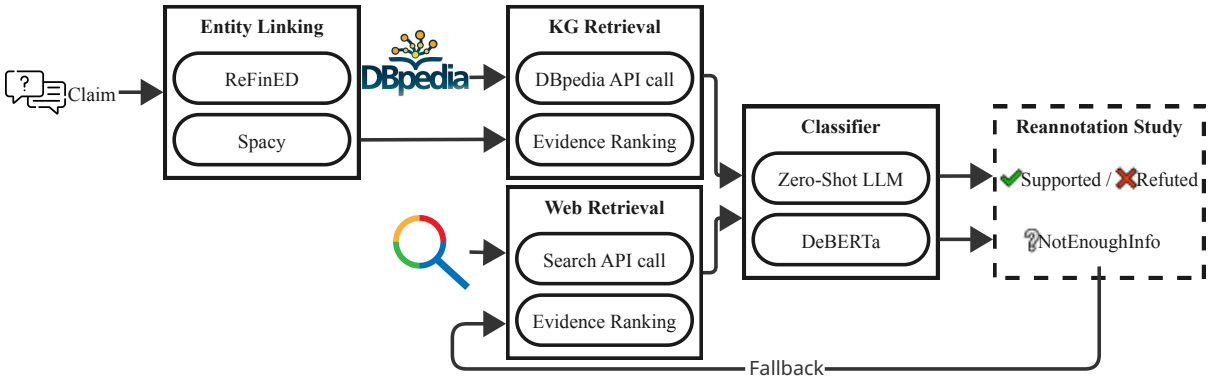

Figure 1: Hybrid fact-verification pipeline: a KG-first pass links entities to Wikidata Q-IDs, retrieves and ranks one-hop DBpedia triples for classification; NEI outputs trigger a Web-RAG fallback that rewrites the claim, retrieves web snippets, and re-evaluates with the same model. Ambiguous NEI cases are validated by human annotators.

IDs are mapped to DBpedia via `owl:sameAs` (Auer et al., 2007). Many synonyms and paraphrases are covered through surface-form dictionaries via ReFinED and Wikipedia redirects, but it does not handle arbitrary paraphrases. In case no Wikidata ID can be assigned, the mention is skipped in the KG stage but may still be handled by the fallback.

Triple retrieval: For each linked entity e, we issue a one-hop SPARQL (Prud'hommeaux and Seaborne, 2006) query to extract all RDF triples $t = \langle s, p, o \rangle \in$ DBpedia where $s = e$ or $o = e$. For example, one triple could look like this: "Barack_Obama -> birthPlace -> Hawaii". We exclude triples with metainformation predicates using a handcrafted blacklist.

Triple scoring: Each candidate triple t is paired with the original claim and scored for semantic relevance using the `ms-marco-MiniLM-L6-v2` cross-encoder (Wang et al., 2020). The input format for this is [CLS] C [SEP] t [SEP]. We retain the top $k = 5$ highest-scoring triples, denoted as $E^*_{\text{KG}} = \{t_1, \ldots, t_k\}$.

KG classification: The set $\{C\} \cup E^*_{\text{KG}}$ is passed to either a GPT-4o mini (OpenAI, 2024) instance or a DeBERTa-v3 MNLI (He et al., 2023) model instance. The model assigns a local label $y_{\text{KG}} \in \{\text{S}, \text{R}, \text{N}\}$ and provides a justification based on the supporting evidence triples. If $y_{\text{KG}} \in \{\text{S}, \text{R}\}$, the pipeline terminates and outputs $Y = y_{\text{KG}}$. Otherwise, we proceed to Stage 2.

Stage 2: Web-Based Fallback

Query rewriting: For cases labeled NOT ENOUGH INFO, we prompt GPT-4o mini to paraphrase the original claim into 3–5 high-recall search queries. These are submitted to the Google Programmable Search API (Developers, 2025).

Snippet retrieval: The top $n \leq 100$ web snippets are collected. Each snippet s_j is scored with the same MiniLM cross-encoder as in Stage 1. We retain the top $k = 5$ snippets, forming $E^*_{\text{Web}} = \{s_1, \ldots, s_k\}$.

Evidence classification: Each (C, s_j) pair is classified using a modular verifier—either a zero-shot LLM (GPT-4o mini) or a DeBERTa-v3 MNLI model—with all configuration details deferred to Section 4. The final verdict is $Y = y_{\text{Web}}$ and $y_{\text{Web}} \in \{\text{SUPPORTED}, \text{REFUTED}, \text{NEI}\}$. If NEI is returned as the output, the fallback mechanism is not triggered again. When the pipeline was configured with an LLM and DeBERTa, we observed that the fallback mechanism was invoked in about 23% of all test cases.

4 Implementation

Our system is built in a modular way so that it can be accessed through a simple REST interface (Fielding, 2000). The modularity makes it easy to test different components or replace models. We experiment with two evidence classifiers:

GPT-4o mini (LLM): For each evidence item e_i, we construct a JSON prompt containing the claim c and the list $\{e_i\}$ (triples or snippets). The model returns {"label": S|R|N, "reason": r}, where r is a single sentence that cites evidence. During development, we tested various LLM prompt variants to maximize classification accuracy and robustness before settling on the final versions reported in our results. The final prompts can be found in the appendix B.

DeBERTa-v3-MNLI: We cast fact verification as natural-language inference. Every pair $\langle c, e_i \rangle$ is transformed into [CLS] e_i [SEP] c [SEP]. The model (He et al., 2023) outputs logits (ℓ_E, ℓ_N, ℓ_C) for {ENTAILMENT, NEUTRAL, CONTRADICTION}. We apply `softmax` and pick the label with the highest probability $p_{\max}$. Afterwards, we map them back to the FEVER labels.

Datasets: For our main experiments, we use the FEVER dataset, which labels claims as Supported, Refuted, or Not Enough Information. To ensure fair comparison across experiments and with other papers and avoid ambiguity, we randomly sample 1,000 FEVER claims, explicitly removing all NEI-labeled instances.

5 Results and Discussion

Table 1 reports the standard NLP accuracy evaluation metrics of precision, recall, and F_1 across (i) claim-only baselines, (ii) single source stages (KG only or Web only), and (iii) the complete two-stage pipeline. Three annotated output examples are provided in Appendix A.

Baselines: Following the claim-only setting in prior work, zero-shot LLMs without retrieval can resolve a portion of FEVER claims but remain ungrounded. The best baseline here (Zero-Shot 4o-mini) results in an F_1 0.801, while Zero-Shot 4.1-nano leads to F_1 0.734. Although these models are competitive, the absence of explicit evidence limits the verifiability of their reasoning.

Separate Stages: Single-source variants show opposing error profiles. KG-only with an LLM results in high precision (0.944) but lower recall (0.734), reflecting reliable yet sparse coverage.

In contrast, web-only configurations are more balanced (e.g., LLM Web-only: Prec. 0.912, Rec. 0.908), suggesting broader coverage at the cost of increased noise.

Model Variant	Prec.	Rec.	F1
Baselines			
Random Choice	0.500	0.500	0.500
BERT-Base (no ret.)	0.649	0.594	0.620
Zero Shot 4.1 nano[1]	0.816	0.720	0.734
Zero Shot 4o mini[2]	**0.826**	**0.790**	**0.801**
Separate Stages			
KG alone, LLM	**0.944**	0.734	0.826
KG alone, DEBERTA	0.882	0.620	0.714
Web only, LLM	0.912	**0.908**	**0.909**
Web only, DEBERTA	0.913	0.878	0.895
Full Pipeline			
LLM, LLM	0.920	0.916	0.917
DEBERTA, LLM	0.883	0.853	0.859
LLM, DEBERTA	**0.930**	**0.926**	**0.927**
DEBERTA, DEBERTA	0.887	0.849	0.860
Stronger LLM 4.1 Mini[1]			
LLM, LLM	**0.932**	**0.931**	**0.931**
LLM, DEBERTA	0.919	0.899	0.908

Table 1: Performance comparison of model variants on FEVER. [1](OpenAI, 2025), [2](OpenAI, 2024)

Full pipeline: Combining KG-first inference with a web fallback led to the highest overall performance among the configurations evaluated. Using the baseline language model (GPT-4o-mini), the full pipeline incorporating a downstream DEBERTA classifier resulted in an F_1 score of approximately 0.927, compared to 0.917 with the language model alone. Substituting the language model with GPT-4.1-mini further increases the F_1 score to 0.931. Consistent with prior work (Li et al., 2024), our pipeline maintains stable performance across different classifier configurations and benefits from increased model capacity.

Design Choice: We adopt a KG-first approach to prioritize precision and interpretability, resorting to Web retrieval only when KG evidence is insufficient (NEI). This design choice improves transparency by grounding decisions in structured evidence and reducing unnecessary web queries.

Dataset	Prec.	Rec.	F1
FEVER 2.0	0.797	0.769	0.783
FactKG	0.791	0.757	0.774

Table 2: Performance on other fact-checking datasets.

Comparisons: Without task-specific fine-tuning, our pipeline transfers well to FEVER 2.0 (F_1=0.78) and FactKG (F_1=0.77). These results can be seen in table 2.

Results	Mode	Acc.
FEVER, Ours	S/R	0.931
(Lewis et al., 2020)	S/R	0.895
FEVER, Ours	S/R/N	0.702
(Tan et al., 2023a)	S/R/N	0.542
FEVER 2.0, Ours	S/R	0.732
(Yuan and Vlachos, 2024)	S/R	0.733

Table 3: Direct comparisons to other related work.

In the context of recent systems using open-domain retrieval and LLMs, prior work reports 89.5% S/R on FEVER with Wikipedia retrieval and a seq2seq verifier (Lewis et al., 2020); Yuan and Vlachos reported 73.34% S/R on FEVER 2.0 via zero-shot triple extraction and KG retrieval, which we match (73%); and Tan et al. reported 54.2% S/R/N on FEVER with web evidence, which we exceed even without considering NEI (results in table 3.

5.1 Analysis of NEI-Labeled Claims

A recurring issue in FEVER involves NEI labels for which our system nonetheless retrieves supporting or refuting evidence. To further examine this, we constructed a targeted evaluation: we randomly sampled 150 NEI claims where our model consistently surfaced evidence and asked two human annotators and one LLM to judge evidence sufficiency (Appendix C).

Over 70% of cases were deemed *sufficient* by at least one human, indicating that the pipeline retrieves meaningful evidence for many claims labeled NEI. Inter-annotator agreement was moderate: Fleiss' κ among humans was 0.385 (compare Figure 2 in Appendix C), with unanimous agreement in 70.7% of instances; LLM-human agreement varied (compare Figure 2, reflecting the sub-

jectivity of sufficiency judgments. These findings suggest that assessing sufficiency depends on annotator strictness and perceived completeness of the evidence. Including more annotators, reconciliation among human annotators, and a broader range of NEI cases could strengthen the reliability of these conclusions. Despite variability, the >70% sufficiency rate (cf. Fig. 3 in Appendix C) suggests that our pipeline reliably finds relevant evidence. Thus, excluding NEI from baseline comparisons is methodologically justified under our setup.

6 Conclusion and Future Work

We present a real-time fact-checking pipeline that combines the strengths of KGs and web retrieval to address the limitations of existing LLM-based and KG-based systems. Our KG-first, web-adaptive approach delivers both high precision and broad coverage, achieving strong empirical results across FEVER and other standard benchmarks without task-specific fine-tuning. It offers competitive accuracy with stronger reliability and interpretability than purely web-based or neural setups. In addition, our NEI re-annotation study shows that in over 70% of cases, the system retrieves meaningful evidence for claims originally labeled *Not Enough Information*. However, subjectivity in human judgments remains a challenge.

Overall, our work demonstrates the value of integrating structured and unstructured evidence for robust, interpretable open-domain fact verification. For future work, we plan to enhance support for multi-hop evidence, improve the detection of truly unverifiable claims, explore alternative classifiers, and extend our approach to additional knowledge sources and datasets.

Limitations

While our KG-first, web-adaptive pipeline achieves strong performance and generalizes well across benchmarks, several limitations remain.

Retrieving multi-hop evidence from KGs is still a major challenge. Our system mainly uses single-hop paths for speed and coverage, but more complex claims may require combining information from multiple nodes or documents, which is not fully captured by our current approach.

The pipeline is also sensitive to error propagation from early components into the pipeline; a long-standing issue in pipelines from NLP tasks to downstream applicationsDiesner et al.. Small mistakes in entity linking, predicate selection, or evidence ranking can propagate through the system and lead to incorrect final labels. This suggests that improving component accuracy, especially early on in the upstream parts, could further enhance overall system reliability.

Additionally, our method assumes that either supporting or refuting evidence can always be found in the KG or on the web. As a result, the system currently has no mechanism for properly handling NEI claims and cannot explicitly indicate when evidence is missing. This limits its applicability to datasets where NEI is a significant or required label.

Finally, by emphasizing broad coverage and adaptability for open-domain fact-checking, the system trades off a few SOTA points on specific, specialized benchmarks. This reflects design choices made to favor practical, real-time usage over narrow optimization.

Ethical Considerations

Developing automated fact-checking systems involves several ethical challenges, particularly around fairness, transparency, and reliability. Our pipeline relies on data from public KGs and accessible (in the sense of visible) web sources, which may contain biases, errors, misinformation, and a lack of diverse perspectives, and relies on the provision of these data by others, which may imply intellectual property constraints that limit their use depending on jurisdiction and use case. These limitations can influence both evidence retrieval and final predictions. Users are responsible for copyright compliance, and we recommend favoring open-access sources. A key part of our evaluation involved human annotation. We recruited two graduate students with strong English proficiency and familiarity with research ethics. Annotators participated in structured training sessions to ensure consistent application of our guidelines. Their judgments in the NEI reannotation study highlighted the subjectivity involved in assessing evidence sufficiency and underscored the importance of incorporating human input when evaluating model outputs. Our system currently does not explicitly model uncertainty or signal when evidence is insufficient, which can lead to overconfident predictions in cases beyond the scope of available sources. Additionally, biases in benchmark datasets, including claim selection and annotation practices, can impact generalizability.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg. Springer-Verlag.

Tomiwa Ayoola, Shikhar Tyagi, James Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. Refined: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Han Cao, Lingwei Wei, Wei Zhou, and Songlin Hu. 2025. Enhancing multi-hop fact verification with structured knowledge-augmented large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39:23514–23522.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. Complex claim verification with evidence retrieved in the wild. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3569–3587, Mexico City, Mexico. Association for Computational Linguistics.

Google Developers. 2025. Custom search json api – programmable search engine. Online. Documentation last updated May 7, 2025.

Jana Diesner, Craig Evans, and Jinseok Kim. 2015. Impact of entity disambiguation errors on social network properties. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, pages 81–90.

Joe Durbin. 2024. Eric trump believes victorious father donald will accomplish more than if he'd won in 2020: 'Be careful what you wish for'. *New York Post*. Accessed: 2025-07-31.

Roy Thomas Fielding. 2000. *Architectural styles and the design of network-based software architectures*. University of California, Irvine.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–7. Association for Computational Linguistics.

Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. 2024. Towards understanding factual knowledge of large language models. In *The Twelfth International Conference on Learning Representations*.

Yichen Jiang, 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.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023a. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9410–9421, Singapore. Association for Computational Linguistics.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023b. Kg-gpt: A general framework for reasoning on knowledge graphs using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023c. Factkg: Fact verification via reasoning on knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 16190–16206. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, and et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Miaoran Li, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhu Zhang. 2024. Self-checker: Plug-and-play modules for fact-checking with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 163–181, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o mini: Advancing cost-efficient intelligence. System card & launch announcement.

OpenAI. 2025. Introducing gpt-4.1 in the api. Includes GPT-4.1, Mini, and Nano models.

Tobias Aanderaa Opsahl. 2024. Fact or fiction? improving fact verification with knowledge graphs through simplified subgraph retrievals. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 307–316, Miami, Florida, USA. Association for Computational Linguistics.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102.

Eric Prud'hommeaux and Andy Seaborne. 2006. Sparql query language for rdf. In *W3C Recommendation*, volume 15, page 2008.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3419–3425.

Xin Tan, Bowei Zou, and Ai Ti Aw. 2023a. Evidence-based interpretable open-domain fact-checking with large language models. *arXiv preprint arXiv:2312.05834*.

Yujia Tan, Wenpeng Zhang, Xiang Ren, and Qiji Chen. 2023b. Oe-fact: Open-domain explanation-enhanced fact-checking with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

James Thorne, 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)*. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The fever2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Quoc Le, Denny Zhou, Ed Chi, Troyer Leang, and Matthew White. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824–24837.

Zhen Xie, Qiji Chen, Xiang Ren, and Xuezhe Ma. 2024. Fire: Fact-checking with iterative retrieval and verification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. Generate-on-graph: Treat llm as both agent and kg for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Moy Yuan and Andreas Vlachos. 2024. Zero-shot fact-checking with semantic triples and knowledge graphs. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 105–115.

Hengran Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2023. From relevance to utility: Evidence retrieval with feedback for fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6373–6384, Singapore. Association for Computational Linguistics.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. Gear: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901. Association for Computational Linguistics.

A Labeled Output Examples

Example 1:

Claim: “Eric Trump’s father is banned from ever becoming president.”
True Label: **Refuted**
Entities: Eric_Trump, President, Father
System: NEI → Web Search → **Refuted**
Explanation: Snippet 2 indicates Donald Trump is a President-Elect, so he is eligible to become president.
Evidence: "Eric Trump, the second son of President-Elect Donald Trump, told The Post this week his father has a long to-do list ready for his White" (Durbin, 2024)

Example 2:

Claim: “Black Mirror is a British science fiction television series about modern society.”
True Label: **Supported**
Entities: Black_Mirror, Television_in_the_United_Kingdom, Science_Fiction
System: **Supported**
Explanation: Path 1 confirms Black Mirror is a British anthology television series exploring science fiction themes about modern society.
Evidence: Path 1: Black_Mirror → Abstract

Example 3:

Claim: “Arya Stark was created by George R. R. Martin.”
True Label: **Supported**
Entities: Arya_Stark, George_R._R._Martin
System: **Supported**
Explanation: Path 1 directly records creator George R. R. Martin for Arya Stark.
Evidence: Path 1: Arya_Stark → creator → George_R._R._Martin

B Classifier prompts

LLM prompt for KG stage

System Prompt (static)

```
You are a world-class fact-verification
assistant.
Given a claim and a numbered list of evidence
paths, choose exactly one label:
  • Supported – at least one path exactly affirms
    the claim’s assertion.
  • Refuted – at least one path explicitly
    contradicts it (e.g. predicate like “is
    not”).
  • Not Enough Info – otherwise.
Rules:
1. If any path affirms the claim’s
predicate+object, label Supported.
2. Only label Refuted if a path uses negation or
clear contradiction.
3. Otherwise label Not Enough Info.
4. Use only the provided paths; do NOT invent
facts.
5. Keep reasoning private — do NOT show
chain-of-thought.
6. Output only a single JSON object:
{
   "label": <Supported|Refuted|Not Enough Info>,
   "reason": <one concise sentence citing path
number(s)>
}
```

User Prompt (input)

```
Claim: <CLAIM>
Evidence paths:
<EVIDENCE_PATHS>
Instruction:
- Label Supported if any path’s predicate and
object exactly match the claim.
- Label Refuted only if a path explicitly
contradicts (uses “not”, “no”, etc.).
- Otherwise label Not Enough Info.
Examples:
1) Supported
Claim: “Alice’s birthplace is Canada.”
1. Alice → birthPlace → Canada
Output:
{"label":"Supported", "reason":"Path 1 exactly
matches birthPlace→Canada."}
2) Refuted
Claim: “Bob is an exponent of Doom metal.”
1. Bob → is not an exponent of → Doom_metal
Output:
{"label":"Refuted", "reason":"Path 1 explicitly
states ‘is not an exponent of Doom metal’."}
3) Not Enough Info
Claim: “Carol’s nationality is Spanish.”
1. Carol → birthPlace → Barcelona
Output:
{"label":"Not Enough Info", "reason":"Path 1 does
not confirm nationality."}
```

LLM prompt for Web-Search stage

System Prompt (static)

```
You are a world-class fact-verification
assistant.
Your job: given a claim and a small numbered list
of evidence snippets, decide only one of two
labels:
  • Supported – at least one snippet clearly
    confirms the claim.
  • Refuted – at least one snippet explicitly
    contradicts the claim.
You must not output any other label.
Use only the provided snippets; do not invent
facts or fetch external data.
Keep your reasoning private — do not expose
chain-of-thought.
Output exactly one JSON object:
{
   "label": <Supported|Refuted>,
   "reason": <one short sentence citing snippet
number(s)>
}
```

User Prompt (input)

```
Claim: <CLAIM>
Evidence snippets:
<EVIDENCE_SNIPPETS>
Instruction:
- If any snippet affirms the claim's exact
assertion, label Supported.
- If any snippet contradicts it (negation,
opposite fact), label Refuted.
- You must choose one of the two — no other
options.
Examples:
Supported Example:
Claim: "Alice's birthplace is Canada."
1. Alice → birthPlace → Canada
Output:
{"label":"Supported", "reason":"Snippet 1 shows
birthPlace → Canada."}
Refuted Example:
Claim: "Bob is an exponent of Doom metal."
1. Bob → is not an exponent of → Doom metal
Output:
{"label":"Refuted", "reason":"Snippet 1 states
'is not an exponent of Doom metal'."}
```

LLM prompt for zero-shot baselines

System Prompt (static)

```
You are a world-class fact checker. You will
receive a claim, and your job is to verify its
factual accuracy based only on your knowledge.
You must choose one of two labels:
  • Supported – the claim is clearly true.
  • Refuted – the claim is clearly false.
If unsure, make your best guess. Avoid using
vague language.
Output exactly one JSON object like this:
{
   "label": "Supported" or "Refuted",
   "reason": "short explanation of why you chose
this label"
}
```

User Prompt (input)

```
Claim: <CLAIM>
Decide whether this is Supported or Refuted.
```

Prompt for Web-Search Paraphrasing

System Prompt (static)

```
You are an expert fact-checking assistant who
writes superb web-search queries.
Given a claim, reformulate it into 3–5 concise,
high-recall search queries. Each query should:
  • be under 12 words
  • keep critical named entities, dates, and
    numbers
  • add quotation marks for exact phrases when
    helpful
  • avoid hashtags or advanced operators other
    than quotes
Return exactly one JSON object like this:
{"queries": [ ... ]}
```

User Prompt (input)

```
Claim: <CLAIM>
```

Column	Description
`nr`	Row number for easy reference
`claim`	The factual statement to be verified
`true_label`	Original FEVER dataset label (always "NOT ENOUGH INFO" for these samples)
`predicted_label`	Our system's prediction ("Supported", "Refuted", or "Not Enough Info")
`found_evidence`	Evidence found by our system (see format explanations below)
`llm_explanation`	LLM's reasoning for cases where prediction $\neq$ "Not Enough Info" (should be hidden during annotation)
`human_annotated`	**[YOUR TASK]** Mark as "sufficient" or "not sufficient"
`notes`	**[OPTIONAL]** Space for your reasoning or additional comments

Table 4: Column structure of our exported CSV file.

C Fact-Checking System Evaluation: Annotation Guidelines for NEI claims

Annotation Instructions

For each row, you need to evaluate whether the evidence provided is sufficient to support the predicted label.

Step-by-Step Process

1. Read the claim carefully
   - Understand exactly what factual statement is being made.
2. Note the predicted label
   - Check if the system predicts `Supported`, `Refuted`, or `Not Enough Info`.
3. Analyze the found evidence
   - **For DBpedia evidence:** Assess if the knowledge paths logically support or refute the claim.

- **For Web evidence:** Evaluate the quality and relevance of the snippets, considering source reliability.

4. Consider additional context (optional)
   - You are welcome to search for additional sources online if needed.
   - Remember that our system considered many more sources than shown.
5. Make your judgment
   - In the `human_annotated` column, enter:
     - `sufficient` if the evidence adequately supports the predicted label.
     - `not sufficient` if the evidence is inadequate, unreliable, or contradictory.
6. Add notes (optional)
   - Use the `notes` column to explain your reasoning.
   - Particularly helpful for borderline cases or when you disagree with the prediction.

Evaluation Criteria

For `sufficient` evidence:

- Evidence directly relates to the claim.
- Sources appear credible and reliable.
- Information is specific and detailed enough to support the conclusion.
- Multiple independent sources corroborate the finding (when available).

For `not sufficient` evidence:

- Evidence is tangentially related or off-topic.
- Sources appear unreliable or biased.
- Information is too vague or general.
- Evidence contradicts itself or the predicted label.

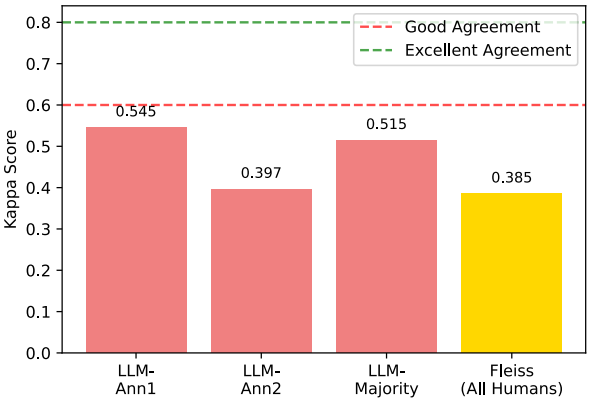

Figure 2: Agreement Scores Comparison. LLM–Human Cohen's κ and Human Fleiss' κ.

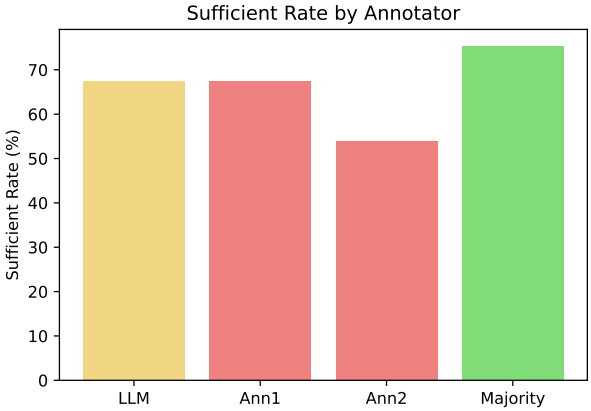

Figure 3: Sufficiency rate differs slightly between annotators.

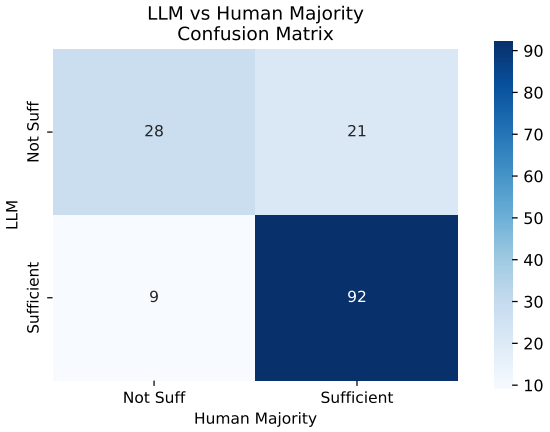

Figure 4: Confusion matrix comparing the LLM's sufficiency judgments with the human majority vote.