open proceedings

Benchmarking Large Language Models for Knowledge Graph Validation

Farzad Shami
farzad.shami@aalto.fi
Aalto University
Espoo, Finland

Stefano Marchesin
stefano.marchesin@unipd.it
University of Padua
Padua, Italy

Gianmaria Silvello
gianmaria.silvello@unipd.it
University of Padua
Padua, Italy

Abstract

Knowledge Graphs (KGs) store structured factual knowledge by linking entities through relationships, crucial for many applications. These applications depend on the KG's factual accuracy, so verifying facts is essential, yet challenging. Expert manual verification is ideal but impractical on a large scale. Automated methods show promise but are not ready for real-world KGs. Large Language Models (LLMs) offer potential with their semantic understanding and knowledge access, yet their suitability and effectiveness for KG fact validation remain largely unexplored.

In this paper, we introduce `FactCheck`, a benchmark designed to evaluate LLMs for KG fact validation across three key dimensions: (1) LLMs internal knowledge; (2) external evidence via Retrieval-Augmented Generation (RAG); and (3) aggregated knowledge employing a multi-model consensus strategy. We evaluated open-source and commercial LLMs on three diverse real-world KGs. `FactCheck` also includes a RAG dataset with 2+ million documents tailored for KG fact validation.

The experimental analyses demonstrate that while LLMs yield promising results, they are still not sufficiently stable and reliable to be used in real-world KG validation scenarios. Integrating external evidence through RAG methods yields fluctuating performance, providing inconsistent improvements over more streamlined approaches – at higher computational costs. Similarly, strategies based on multi-model consensus do not consistently outperform individual models, underscoring the lack of a *one-fits-all* solution. These findings further emphasize the need for a benchmark like `FactCheck` to systematically evaluate and drive progress on this difficult yet crucial task.

Keywords

Knowledge Graph, Large Language Model, Fact Validation

1 Introduction

Knowledge Graphs (KGs) are machine-interpretable, directed, labeled multigraphs in which nodes represent entities or concepts, and edges denote typed semantic relations. They provide a structured representation of real-world knowledge, enabling reasoning, integration, and querying across information sources [23, 26, 47]. KGs have been deployed in a wide range of applications [23, 24], including: (1) web search for semantic understanding of queries and content [16, 41]; (2) e-commerce, for recommendation [56] and conversational agents [49]; (3) social networks, for modeling user interests [18, 41]; and (4) other domains such as finance [3], transport [21], and energy [72]. However, the effectiveness of downstream applications depends on the accuracy of the KG's facts. Each individual piece of knowledge, which is typically represented as an `<S,P,O>` triple (i.e., Subject-Predicate→Object), must be factually correct. In addition, the reliability of the entire KG depends not only on the correctness of these atomic facts, but also on the way they are interconnected [8, 50].[1]

A crucial step after the creation of the KG is assessing the veracity of its facts [23, 51]. This involves determining how accurately the data reflects real-world entities, relationships, and phenomena. Fact validation presents a significant challenge and is expensive [36, 37]. The most reliable option involves manual or computer-assisted annotation by human experts [42, 66]. However, this process is extremely time-consuming [12, 43]. Since experts often need to audit facts relying on multiple external reference sources, in large-scale KGs (e.g., DBpedia [33] or YAGO [22]), verifying each individual triple can take several minutes, making manual inspection and correction of errors infeasible at scale.

As a result, automated fact-checking methods [11, 14, 58, 62, 63], often based on rules and enforceable constraints [10, 40], have emerged as more scalable alternatives to address the time and cost limitations of human-based solutions. While these methods are effective for well-defined and frequently occurring facts [29]), they fall short when it comes to generalizing across the wide variety of facts found in real-world KGs. Manual definition, on the other hand, is both difficult and expensive. Therefore, (semi-) automatic methods that extract rules and constraints can be employed. Nonetheless, these methods predominantly cater to rules that identify frequent positive instances and encounter difficulties with cases pertaining to infrequent facts or necessitate the application of negative rules [45].

These limitations have led to the adoption of fact-checking systems with machine/deep learning solutions [17]. In this realm, a viable approach could be to utilize Large Language Models (LLMs) for fact-checking, as they have demonstrated near-human-level performance on various tasks [46]. Within this framework, LLMs offer various advantages: they can extract contextual information from text, comprehend the semantics of statements, and possess an extensive internal knowledge base [48, 65]. However, current LLMs generate hallucinated and unfaithful responses [60]. Additionally, recent work has highlighted that LLMs are particularly problematic for fact validation tasks, exhibiting systematic biases and knowledge gaps that can affect their reliability [59]. To combat the limitations caused by knowledge cutoff and hallucination in LLMs, current systems built on top of LLMs often implement a Retrieval-Augmented Generation (RAG) approach in which the LLM is supplemented with data from external sources to improve their responses [28]. However, despite all the recent progress in LLM research and the capability of LLMs to tackle a wide range of tasks, there appear to be no existing benchmarks specifically measuring the performance of LLMs in KG fact validation [51].

Hence, we present `FactCheck`, a general-purpose benchmark designed to assess LLMs in the validation of KG facts across three

[1] We use the terms fact, statement, and triple interchangeably depending on the context.

EDBT '26, Tampere (Finland)

principal dimensions: (1) LLM internal knowledge; (2) external evidence through Retrieval-Augmented Generation (RAG); and, (3) synthesized knowledge from multiple models.

FactCheck relies on a validation pipeline that transforms structured triples into natural language statements for evaluation of their factual accuracy. The validation procedure begins with KG entities and relations, derives structured triples, checks them against reliable sources, and calculates accuracy scores.

FactCheck is driven by the following research questions (RQs):

RQ1: How effective are LLMs at KG fact-checking when relying only on their internal knowledge?

RQ2: Does external evidence improves the ability of LLMs to fact-check KGs?

RQ3: Does aggregating predictions from multiple LLMs lead to more reliable validation of KG facts?

RQ1 targets a recent debate concerning LLMs functioning as Knowledge Bases (KBs), aiming to evaluate how factual and complete the internal knowledge of an LLM is, for both previously seen and unseen knowledge [19, 20, 71]. We do not prompt the LLM to retrieve knowledge to evaluate its completeness and accuracy. Instead, we ask the LLM to judge the accuracy of externally provided facts, which requires it to depend solely on its internal knowledge. Our focus is directed towards this research approach, acknowledging that studies indicate querying an LLM for the verification of information accuracy produces more favorable outcomes compared to prompting it to generate or assess its own content [27, 31].

RQ2 targets the effectiveness of augmenting LLMs with external evidence to improve KG fact-checking, contributing to ongoing discussions around RAG and its role in factual verification [28, 53, 70]. While classical RAG approaches often outperform LLMs that rely solely on internal knowledge, recent findings indicate that RAG effectiveness can diminish in complex or multi-turn settings, where context management and evidence selection become more error-prone [32]. Moreover, integrating external evidence can introduce contextual bias, where the model overly trusts retrieved content [34]. With FactCheck, we aim to foster research on whether and under what conditions external evidence helps KG fact validation, to what extent, and under which conditions.

RQ3 targets a growing body of work investigating whether aggregating outputs from multiple LLMs can lead to more accurate or reliable factual verification [7, 54]. While individual LLMs may vary in factual accuracy, reasoning patterns, and susceptibility to hallucinations, recent studies suggest that combining multiple models – via voting, consensus, or arbitration mechanisms – can mitigate individual model biases and increase robustness [64, 67]. However, this approach introduces its own challenges, including disagreement resolution, scaling cost, and the risk of amplifying shared misconceptions among models trained on overlapping data. FactCheck can help explore whether ensemble-style reasoning from multiple LLMs can improve the reliability of KG fact-checking.

Contributions. We propose FactCheck, a benchmark for KG fact validation using LLMs, which comes with several advantages:

(1) FactCheck integrates various LLMs for KG fact validation. The benchmark evaluates these models using both their internal knowledge and external evidence through RAG. It also explores consensus-based verification via majority voting strategies. Experiments with mid-sized (7–9B parameters) and commercial LLMs highlight the challenges of the task.

(2) FactCheck is built upon three real-world KG datasets: *FactBench* [14], *YAGO* [43], and *DBpedia* [38], covering broad spectrum of knowledge, ranging from everyday facts to complex, domain-specific information, ensuring a diverse and representative evaluation of fact validation capabilities.

(3) FactCheck includes a large-scale RAG dataset featuring several questions paired with corresponding Google Search Engine Results Pages (SERPs). The dataset comprises 2M+ documents covering a broad range of factual information, making it one of the most comprehensive and publicly available RAG resources for KG fact validation. FactCheck includes a mock API that simulates real search APIs, allowing users to reproduce data retrieval, test retrieval methods, and extend RAG methods without direct access to search engines.

(4) A dedicated web application (https://factcheck.dei.unipd.it/), enabling users to visually explore and analyze each step of the verification process, also featuring error analysis modules that categorize reasoning errors, enabling systematic identification of LLM limitations in fact-checking scenarios.

(5) FactCheck enables comprehensive evaluation by combining performance metrics with resource usage analysis. Model predictions are evaluated against gold-standard labels to assess accuracy and reliability. The benchmark also tracks computational costs (inference time and token usage).

Evaluation with different methodologies and datasets highlights the difficulty and inherent complexity of the fact validation task in KG. The main insights of our work are three-fold: First, while LLMs show promising capabilities in KG fact validation, they are still far from being reliably deployed in real-world validation scenarios. Second, integrating external knowledge through RAG yields fluctuating performance, providing inconsistent improvements over more streamlined approaches at significantly higher computational costs. Finally, consensus-based strategies using multiple models are unable to consistently outperform individual models. Altogether, these results highlight the task's difficulty and complexity, underscoring the need for a dedicated benchmark to drive progress.

Outline. The rest of the paper is organized as follows. In Section 2, we review related work on automated KG fact-checking and benchmark development. In Section 3, we introduce the FactCheck benchmark. We detail the FactCheck construction in Section 4, covering both dataset selection and RAG corpus creation. Section 5 outlines the experimental setup, with results discussed in Section 6. Section 7 provides a qualitative error analysis of failure cases. Finally, in Section 8, we draw final remarks.

2 Related Work

2.1 Automated KG Fact Checking

Fact-checking methods can be categorized into approaches that directly utilize the KG to find a supporting path for the given statements [29, 57, 58, 61] and others relying on external reference sources to find supporting or conflicting evidence [14, 62]. Table 1 represent comparative analysis of these two paradigms.

(1) Internal KG-Based Fact Checking. Knowledge Stream (KStream) and Relational Knowledge Linker (KLinker) [58] are unsupervised, network-flow-based approaches designed to assess the truthfulness of factual statements expressed as <S,P,O> triples. KStream models a KG as a flow network, where the path carries flow from a subject to an object to support or refute a

Table 1: Comparative analysis of Internal KG-Based versus External Evidence-Based fact-checking mechanisms.

Feature	Internal KG-Based Fact Checking	External Evidence-Based Fact Checking
Principle	*Coherence*: Consistent with graph patterns.	*Correspondence*: Aligns with external sources.
Primary Evidence	Graph topology, paths, and flow networks	Unstructured text, webpages, and search snippets
Assumption	Derives negative signals from missing links based on local completeness.	Missing links are verified against external data under incompleteness.
Mechanism	Path mining, link prediction.	IR, NLP, RAG.
Handling Negatives	Synthesized via sampling strategies (e.g., [29]).	Retrieval failure or contradiction.
Trade-offs	**(+)** Fast, Consistent. **(-)** Misses graph errors.	**(+)** High validity. **(-)** Slow, source-dependent.
Examples	KStream [58], PredPath [57], COPPAL [61].	DeFacto [14], KGValidator [5], `FactCheck` (Ours).

given statement. KLinker, on the other hand, focuses on discovering relational paths that link entities to each other. COPPAL [61] proposes a corroborative meta-path to find statement-supporting paths. These approaches focus only on positive evidential paths and are heavily restricted due to the incomplete nature of KGs. Approaches like PredPath [57] attempt to utilize both negative and positive paths to cover a broader range of factual statements. PredPath assigns weights to discriminative predicate paths by considering only correct examples, ignoring counterexamples. This can lead to improperly weighted rules. In addition, Kim and Choi [29] presents an unsupervised rule-based approach that significantly outperforms the state-of-the-art unsupervised approaches in this area. They calculate a truth score for the given statement by finding positive and negative evidential paths in a KG, generating examples for the training phase, creating a model for learning from positive and negative rules, and scoring the triple based on established evidence.

While these methods are effective, they rely entirely on the underlying KG, which may contain errors or be incomplete; thus, they cannot be used to assess the accuracy of the KG itself.

***(2) External Evidence-Based Fact Checking*.** DeFacto [14] is a supervised learning method that validates KG triples using evidence retrieved on the Web. To compute an evidence score, this method integrates trustworthiness metrics with textual evidence. Syed et al. [62] proposed a fact validation method that uses textual evidence from a static reference corpus as external knowledge. They verbalized triples into natural language, queried a search engine to retrieve similar corpus sentences, and then extracted evidence and features from these sentences to estimate each KG triple's confidence with a trained model. Recently, Boylan et al. [5] introduced KGValidator, a framework for the automatic evaluation of KG completion models using LLMs. KGValidator assesses predicted triples by leveraging multiple sources of context, including the LLM's internal knowledge, user-provided textual documents, and web resources. In contrast to this methodological contribution, FactCheck focuses on providing the supporting evaluation infrastructure – i.e., datasets, metrics, and curated evidence corpora – needed to systematically assess and compare such validation approaches.

Aligning with prior work that incorporates external sources for fact verification [5, 14, 62], `FactCheck` allows LLMs to employ external evidence retrieved from Web SERPs. Additionally, `FactCheck` offers several LLM-based baselines, enabling a comparative evaluation of LLM with external evidence-driven solutions. Moreover, `FactCheck` assesses LLM performance across three real-world KG datasets (13,530 facts) tailored for the task, supported by 2M+ retrieved documents as external evidence.

2.2 Benchmarks and Datasets

CRAG [69] is a **benchmark** designed to evaluate the effectiveness of RAG systems, with a focus on factual accuracy. It includes 4, 409 Question-Answer pairs spanning five domains and eight question categories. To simulate realistic usage scenarios, CRAG offers mock APIs for web and KG searches. The benchmark specifically targets challenges such as answering less popular or rapidly evolving facts, assessing LLM performance across varying levels of entity popularity and temporal relevance. While CRAG and FactCheck both utilize RAG, they address fundamentally different problems with distinct evaluation goals. Indeed, FactCheck evaluates KG fact validation, prioritizing accuracy and consistency. CRAG cannot replace FactCheck because high-performing QA models often fail at the strict, granular logic required to validate isolated KG triples. Additionally, `FactCheck` provides detailed information on computational costs and resource efficiency, both aspects not extensively covered by CRAG. Hence, although related, these benchmarks address different aspects of factual verification.

Beyond CRAG, there are several pipelines and shared tasks for fact-checking purposes targeting textual claims. RumourEval [15] evaluated classification systems by analyzing social media posts by stance detection and rumor veracity verification, employing a dataset containing data from Twitter and Reddit. CLEF CheckThat! [1] offers sentence-level subjectivity detection in news articles. ClaimBuster [2] introduced an automated end-to-end fact-checking pipeline integrating claim detection, matching, and verification. As said, these benchmarks primarily target unstructured textual claims and cannot be used for KG fact verification.

Few **datasets** have been proposed for KG verification [14, 38, 43]. A key one is *FactBench* [14], built from DBpedia [33] and Freebase [4] KGs to evaluate validation systems on systematic errors. Other datasets include *YAGO* [43] and *DBpedia* [38], which consist of samples drawn from their respective KGs and manually annotated by experts for correctness. While these datasets have been employed in both manual and automated verification settings, they have seen minimal to no use with LLM-based approaches. Hence, we employ FactBench, YAGO, and DBpedia in `FactCheck`, as they capture complementary aspects of fact verification challenges, enabling a multifaceted evaluation of LLM-based strategies. Another related dataset is FactKG [30], designed for fact verification over KGs. However, FactKG uses KGs to verify textual claims, whereas our work takes the opposite direction: using external evidence to help LLMs validate KG facts.

3 FactCheck

This section details the strategies used in `FactCheck` to address the study's RQs. The benchmark includes multiple strategies using both open-source and commercial LLMs. In §3.1, we present two approaches that rely solely on LLMs' internal knowledge to verify KG facts (RQ1). In §3.2, we introduce a RAG approach that augments LLMs with external evidence (RQ2). Finally, §3.3

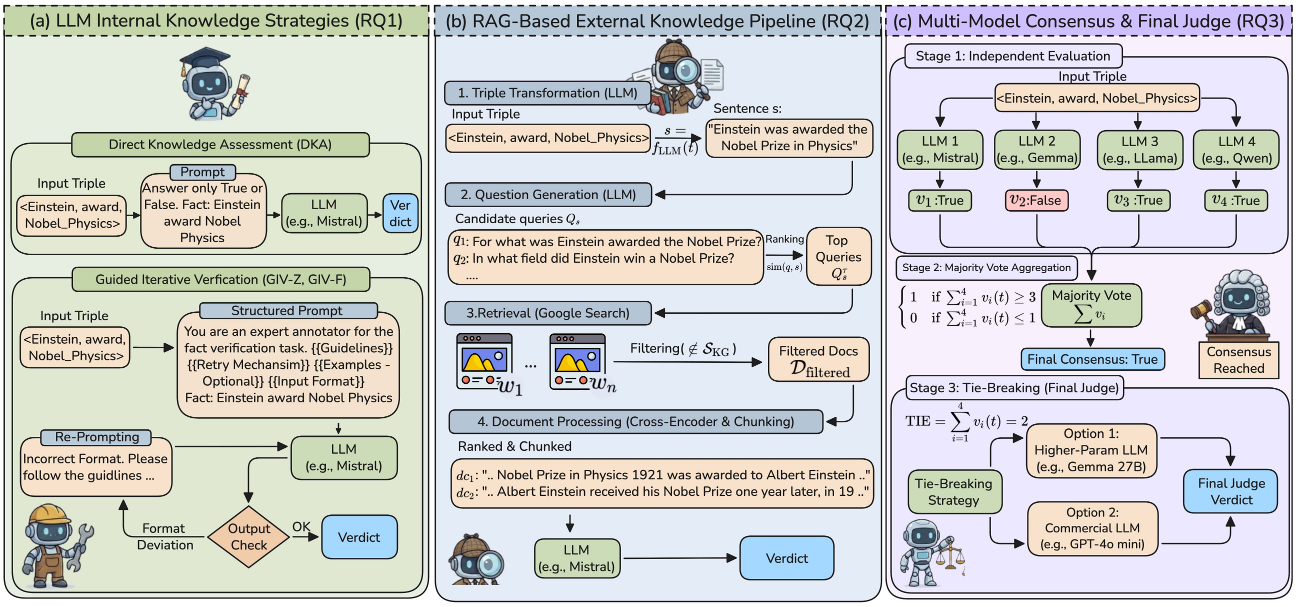

(a) LLM Internal Knowledge Strategies (RQ1)
Direct Knowledge Assessment (DKA)
Input Triple
<Einstein, award, Nobel_Physics>
Prompt
Answer only True or False. Fact: Einstein award Nobel Physics
LLM (e.g., Mistral)
Verdict
Guided Iterative Verfication (GIV-Z, GIV-F)
Input Triple
<Einstein, award, Nobel_Physics>
Structured Prompt
You are an expert annotator for the fact verification task. {{Guidelines}} {{Retry Mechansim}} {{Examples - Optional}} {{Input Format}} Fact: Einstein award Nobel Physics
Re-Prompting
Incorrect Format. Please follow the guidlines ...
LLM (e.g., Mistral)
Format Deviation
Output Check
OK
Verdict
(b) RAG-Based External Knowledge Pipeline (RQ2)
1. Triple Transformation (LLM)
Input Triple
<Einstein, award, Nobel_Physics>
$s = f_{\text{LLM}}(t)$
Sentence s:
"Einstein was awarded the Nobel Prize in Physics"
2. Question Generation (LLM)
Candidate queries Q_s
q_1: For what was Einstein awarded the Nobel Prize?
q_2: In what field did Einstein win a Nobel Prize?
....
Ranking
$\text{sim}(q,s)$
Top Queries Q_s^τ
3.Retrieval (Google Search)
w_1
...
w_n
Filtering($\notin \mathcal{S}_{\text{KG}}$)
Filtered Docs $\mathcal{D}_{\text{filtered}}$
4. Document Processing (Cross-Encoder & Chunking)
Ranked & Chunked
dc_1: ".. Nobel Prize in Physics 1921 was awarded to Albert Einstein .."
dc_2: ".. Albert Einstein received his Nobel Prize one year later, in 19 .."
LLM (e.g., Mistral)
Verdict
(c) Multi-Model Consensus & Final Judge (RQ3)
Stage 1: Independent Evaluation
Input Triple
<Einstein, award, Nobel_Physics>
LLM 1 (e.g., Mistral)
LLM 2 (e.g., Gemma)
LLM 3 (e.g., LLama)
LLM 4 (e.g., Qwen)
v_1 :True
v_2:False
v_3 :True
v_4 :True
Stage 2: Majority Vote Aggregation
$\begin{cases} 1 & \text{if } \sum_{i=1}^{4} v_i(t) \geq 3 \\ 0 & \text{if } \sum_{i=1}^{4} v_i(t) \leq 1 \end{cases}$
Majority Vote $\sum v_i$
Final Consensus: True
Consensus Reached
Stage 3: Tie-Breaking (Final Judge)
$\text{TIE} = \sum_{i=1}^{4} v_i(t) = 2$
Tie-Breaking Strategy
Option 1: Higher-Param LLM (e.g., Gemma 27B)
Option 2: Commercial LLM (e.g., GPT-4o mini)
Final Judge Verdict

Figure 1: Overall overview of the FactCheck benchmark.

describes a multi-model consensus strategy that aggregates predictions from multiple LLMs to improve verification accuracy (RQ3).

3.1 LLM Internal Knowledge

To address RQ1, FactCheck employs two different strategies:

Direct Knowledge Assessment (DKA) is a simple strategy consisting of a basic, direct prompt for the LLM without any further guidance. DKA aims to evaluate the ability of LLMs to verify facts using only internal knowledge. We use DKA as the baseline for comparing different LLMs and more advanced strategies. An example is reported in the top left part of Figure 1(a).

Guided Iterative Verification (GIV) (see the bottom left part of Figure 1(a)) is an iterative prompting approach leveraging a structured prompt template that outlines the expected output format, and, optionally, enforces dataset-specific constraints. If a model's output is non-conformant, the system triggers a re-prompting, explicitly flagging the non-compliance. Responses that repeatedly fail to meet the criteria are marked as invalid. We consider both zero- and few-shot settings. In the few-shot setting, we include a small set of correctly evaluated triples as examples to guide the model's understanding of the task. These examples are shared across datasets and KG-independent at the semantic level, while their encoding is adapted to the target KG to align with predicate and schema conventions.

3.2 External Knowledge

To address RQ2, we enhance LLMs with RAG. Given a KG triple t, we retrieve a set of documents $\mathcal{D}$ containing potentially supporting or refuting evidence. We implement this through a multistage pipeline comprising four main phases: (1) triple transformation, (2) question generation and ranking, (3) document retrieval and filtering, and (4) document processing and chunking. Figure 1(b) illustrates the core components of the RAG-based verification engine in FactCheck.

In the **Triple Transformation** phase (1), structured KG triples are converted into human-readable sentences. This transformation is performed using an LLM to address the substantial variability in how different KGs represent $\langle S, P, O \rangle$ data. KGs follow heterogeneous conventions for encoding triples, and these source-specific formats often hinder effective information retrieval. Common issues include (1) KG-specific namespaces (e.g., *dbpedia.org/resource/:term:*); (2) special notation such as underscores or camelCase (e.g., *isMarriedTo*, *Alexander_III_of_Russia*); and (3) predicates that lack sufficient grammatical or semantic context. Such representations can restrict search results to the original source pages from which the triples were extracted, thereby introducing retrieval bias and limiting coverage during evaluation. By contrast, natural language reformulations facilitate the discovery of a broader range of relevant web sources. We define this process as a transformation function $s = f_{\text{LLM}}(t)$ that maps a triple t to a natural language sentence s.

In the **Question Generation and Ranking** phase (2), for any given sentence s, we prompt an LLM to generate a set of candidate queries $Q_s = \{q_1, q_2, \ldots, q_{k_q}\}$. The goal of generating multiple questions is to broaden the semantic coverage of the original triple, improving the chances of retrieving relevant evidence – even when the input is ambiguous, noisy, or underspecified. Generating multiple questions also helps mitigate the paraphrasing bias that the LLM may introduce when turning triples into natural language. By formulating several distinct questions, we broaden the range of possible interpretations of a given triple, thereby weakening the link to any single facet that might otherwise be imposed by one particular LLM-generated paraphrase. To identify the most informative queries, we apply a cross-encoder model (*jina-reranker-v1-turbo-en*), which corresponds to the normalized dot product between the cross-encoder's final representation and a learned relevance vector (i.e., a sigmoid-scaled dot-product score). This score reflects the semantic proximity between a candidate query $q \in Q_s$ and the original sentence s. The resulting set is $Q_s^{\text{ranked}} = \{q_{(1)}, q_{(2)}, \ldots, q_{(k_q)}\}$, where $\text{sim}(q_{(i)}, s) \geq \text{sim}(q_{(i+1)}, s)$ for all $i \in \{1, 2, \ldots, k_q - 1\}$. We retain the top-$\tau$ queries, denoted as Q_s^τ, using a predefined threshold $\tau \in [0, 1]$ to ensure only the most relevant queries are used.

In the **Document Retrieval and Filtering** phase (3), we issue each query in Q_s^τ to Google Search using specific parameters to ensure consistency. We set lr = "lang_en" and hl = "en" to enforce English content and interface settings, and gl = "us" to standardize the geolocation to the United States, thereby mitigating local personalization bias. Using num = "100", we collect the top $n_{\max} = 100$ retrieved webpages, denoted as $\mathcal{R}(q) = \{w_1, w_2, \ldots, w_{n_{\max}}\}$. For each webpage $w_i \in \mathcal{R}(q)$, we extract its textual content, denoted as $\text{text}(w_i)$. The set of documents retrieved for a given query q is then defined as $\mathcal{D}(q) = \{d_i = \text{text}(w_i) \mid w_i \in \mathcal{R}(q)\}$. To obtain the full document pool associated with the original triple t, we take the union over all queries in Q_s^τ: $\mathcal{D} = \bigcup_{q \in Q_s^\tau} \mathcal{D}(q)$. To ensure evidence independence and avoid circular verification, we define $\mathcal{S}_{\text{KG}}$ as the set of original KG sources – for instance, Wikipedia entries when verifying facts from DBpedia and FactBench datasets. We use this set to filter out any retrieved documents that directly originate from these sources. The resulting filtered document set is defined as $\mathcal{D}_{\text{filtered}} = \{d \in \mathcal{D} \mid \text{source}(d) \notin \mathcal{S}_{\text{KG}}\}$.

Finally, in the **Document Processing and Chunking** phase (4), we use a cross-encoder to identify the k_d most relevant documents with respect to the sentence s. For each document $d \in \mathcal{D}_{\text{filtered}}$, a similarity score $\text{sim}_d(d, s)$ is computed using the same approach as above. The top k_d documents, ranked by similarity, form the final set $\mathcal{D}_{\text{final}} = \{d_1, d_2, \ldots, d_{k_d}\}$. Each document in $\mathcal{D}_{\text{final}}$ is segmented into smaller, overlapping passages using a sliding window chunking strategy. These chunks are subsequently used as contextual input in the LLM prompt during the fact validation stage.

3.3 Multi-Model Consensus

Since LLMs can output different answers for the same fact-checking task, we also explore a model consensus strategy (Figure 1(c)). Building on §3.1 and §3.2, let $\mathcal{M} = \{M_1, M_2, M_3, M_4\}$ be the set of LLMs. For each triple t, each model $M_i \in \mathcal{M}$ produces a binary verdict $v_i(t) \in \{0, 1\}$, where 0 means "false" and 1 means "true".

We employ a simple majority vote strategy to determine the final verdict. The consensus decision $V_{\text{final}}(t)$ for a given triple t is:

$$V_{\text{final}}(t) = \begin{cases} 1 & \text{if } \sum_{i=1}^{4} v_i(t) \geq 3 \\ tie & \text{if } \sum_{i=1}^{4} v_i(t) = 2 \\ 0 & \text{otherwise} \end{cases}$$

The strategy aims to mitigate errors by reducing the impact of outlier predictions. In the event of a tie, we apply a conflict resolution strategy. Let M_{judge} be the *final judge* module responsible for breaking ties. We explore two approaches for defining M_{judge}:

(1) A higher-parameter variant of one of the models in our set $\mathcal{M}$, selected based on its consistency score CA_M. This score represents the proportion of instances where the model's output agrees with the majority prediction across datasets – serving as a proxy for its alignment with correct outcomes. We test both the most consistent (highest CA_M) and least consistent (lowest CA_M) models, upgrading them to higher-parameter versions (e.g., Gemma2:9B ⟶ 27B).

(2) A commercial model with a different architecture and training pipeline – such as *GPT-4o mini* – to offer an independent perspective in resolving ambiguous cases.

4 Benchmark Construction

In this section, we present the entire pipeline for constructing the `FactCheck` benchmark. First, in §4.1, we detail the process of collecting triples from existing KG datasets, along with the creation of a new dataset specifically tailored for the RAG methodology. Next, in §4.2 and §4.3, we describe the LLMs, the evaluation metrics, and the automated assessment procedures used in `FactCheck`.

4.1 Datasets

The `FactCheck` dataset consists of two main components: (i) triples derived from three real-world KGs, and (ii) content retrieved from Google SERPs. This section describes each of these components and introduces the mock API, which mimics a realistic scenario and provides standardized access to the dataset for reproducible experimentation.

KG Datasets. We include triples from three real-world and widely used KG datasets – FactBench, YAGO, and DBpedia. Note that we employ these datasets with a snapshot-based semantics: a triple is deemed true if it is supported by the underlying KG snapshot used to build it, and false otherwise. Table 2 summarizes the key statistics for each of these datasets.

Table 2: Summary of FactBench, YAGO, and DBpedia datasets.

	FactBench	YAGO	DBpedia
Num. of Facts	2,800	1,386	9,344
Num. of Predicates	10	16	1,092
Avg. Facts per Entity	2.42	1.69	3.18
Gold Accuracy (μ)	0.54	0.99	0.85

FactBench is a multilingual benchmark developed by Gerber et al. [14] to evaluate fact validation algorithms. It includes ten relation types and supports English, German, and French. In `FactCheck`, we focus exclusively on the English subset. Positive (correct) facts are sourced from DBpedia and Freebase, while negative (incorrect) facts are generated systematically by altering the correct ones – ensuring adherence to domain and range constraints. We use a configuration with a proportion of positive facts of $\mu = 0.54$, achieved by mixing correct facts with incorrect ones generated through various negative sampling strategies [37].

YAGO is an evaluation dataset sampled from the YAGO KG, originally introduced by Ojha and Talukdar [43] and widely adopted for KG accuracy estimation [12, 36, 37]. It comprises 1,386 facts spanning 16 distinct predicates, with an average of 1.69 facts per entity. All facts are annotated by crowdworkers, resulting in a gold standard accuracy of $\mu = 0.99$. This high accuracy presents a unique challenge for fact-checking, as LLMs may be biased toward classifying all facts as correct, thereby inflating performance metrics.

DBpedia is an evaluation dataset sampled from the DBpedia KG, originally introduced by Marchesin et al. [38]. It was constructed using a combination of sampling and active learning techniques, with both expert and layman annotators involved to ensure high annotation quality. The triples were acquired from the 2015-10 English version of DBpedia, with subject entities required to be part of triples that include `rdfs:label` and `rdfs:comment` predicates. To focus exclusively on factual assertions, T-Box triples – those representing ontological entities and schema-level relationships – were excluded, retaining only A-Box assertions, which represent concrete factual claims. Each triple was annotated by at least three annotators, resulting in a dataset of 9, 934 triples with a gold standard accuracy of $\mu = 0.85$, covering 1, 092 distinct predicates.

RAG Dataset. We constructed a RAG dataset comprising questions derived from KG facts and corresponding search results. This dataset was created as support to effectively evaluate LLM performance in fact validation tasks involving external knowledge. The dataset consists of two main components: the generated questions and their associated search results obtained from Google SERPs.

For **Questions**, we used an LLM to generate $k_q = 10$ distinct questions for each transformed triple s, aiming to explore different facets of the underlying fact. For dataset construction, we included all questions that were successfully extracted from the model's output. Each question is published along with its corresponding similarity score, computed with respect to the transformed triple. FactCheck comprises a total of $Q = 130,820$ questions generated for 13, 530 facts. Each fact is associated with a variable number of questions (q_t) ranging from $\min(q_t) = 2$ to $\max(q_t) = 10$, with a mean of $\mu_{q_t} = 9.67$ and a median of $\tilde{q}_t = 10.00$.

Each question is assigned a similarity score $\delta \in [0, 1]$ that quantifies its semantic closeness to the transformed triple. Across all questions, the similarity scores exhibit a mean of $\mu_\delta = 0.63$ and a median of $\tilde{\delta} = 0.66$. The standard deviation is $\sigma_\delta = 0.25$, indicating moderate variability. The first quartile is $Q_1 = 0.44$ and the third is $Q_3 = 0.84$, resulting in an Inter Quartile Range (IQR) of $\text{IQR} = Q_3 - Q_1 = 0.40$, which confirms substantial variation in similarity scores across the dataset.

To further analyze this distribution, we categorize the questions into three similarity tiers: high similarity ($\delta \geq 0.70$), constituting 45% of the dataset; medium similarity ($0.40 \leq \delta < 0.70$), accounting for 34%; and low similarity ($\delta < 0.40$), making up the remaining 21%. This distribution shows that 79% of the dataset consists of questions with at least moderate similarity to the transformed triple ($\delta \geq 0.40$), and nearly half show high similarity. This range of similarity levels covers both semantically close and more loosely related interpretations of each fact.

Regarding **Google Search Results**, for each fact, we submitted the transformed original triple along with the top three generated questions – ranked by their similarity scores – to Google Search. After parsing the HTML responses, we retrieved each URL using the *GRequests* Python library. The content of the resulting webpages was extracted using the *newspaper4k* [2] Python package.

The corpus consists of D = 2,090,305 documents across 13,530 triples. Each triple t is linked to d_t documents, with $\min(d_t) = 0$, $\max(d_t) = 337$, mean $\mu_{d_t} = 154.51$, and median $\tilde{d}_t = 160$. The slightly higher median indicates a mild negative skew, with most triples having document counts around or just above the mean.

We define $\mathcal{E}_{\text{text}} \subset D$ as the subset of documents with empty text content. This subset contains $|\mathcal{E}_{\text{text}}| = 263,515$ documents, representing the 13% of the entire collection. Consequently, the text coverage rate – i.e., the proportion of documents presenting text content – is $1 - |\mathcal{E}_{\text{text}}|/|D| = 0.87$ (87%). This high coverage rate supports the reliability of the constructed document collection.

Table 3: Summary of average time and token usage for each step in the RAG dataset generation pipeline.

Task	Avg. Time	Avg. tokens
Question Generation	9.60 sec	672.58
Get documents (Google pages)	3.60 sec	–
Fetch documents for each triple	350 sec	–

In Table 3, we report the time consumption and token expenditure incurred during the generation of the RAG dataset. Overall, question generation requires an average of 9.60 seconds per fact, whereas the complete Google results retrieval process takes approximately 364.4 seconds.

To ensure fairness and reproducibility in evaluation, we generated all questions and collected the corresponding Google SERP results in advance. This provides a consistent evidence base for LLMs, avoiding discrepancies caused by changes in live search outputs. The complete dataset is publicly available on our HuggingFace project page and accessible via the mock API.[3]

Mock API. In FactCheck, we integrate a web search-like API for content retrieval to simulate realistic scenarios for RAG. This API facilitates reproducible benchmarking by offering standardized access to pre-collected search data, thereby removing temporal variability in search results.

For each fact in the considered datasets, we issued queries using both the transformed triple and the top three generated questions. We stored the first 100 results for each query from Google SERP, and subsequently retrieved and preserved the actual content of each linked webpage. As previously discussed, we filtered out sources directly related to the original fact to avoid circular verification.

We implemented standardized endpoints that emulate conventional web search APIs while returning consistent results from our dataset. Through this mock API, researchers can perform identical retrieval operations across multiple experimental runs, ensuring fair comparisons between different LLM configurations, prompting strategies, and verification approaches. The mock API can be accessed at https://factcheck-api.dei.unipd.it/. Full documentation is available on GitHub.[4]

4.2 Models

We integrate four open-source LLMs in the 7-9B parameter range as the backbone of our KG fact validation pipeline: Gemma2, Qwen2.5, Mistral, and Llama3.1. We prioritize open-source models for several reasons. First, they can be deployed in diverse environments, including settings with strict data privacy requirements or limited API access, as they can be hosted locally without relying on external services. Second, they offer greater tunability, allowing fine-tuning on domain-specific data or adaptation to specialized fact validation tasks. Third, they are significantly more cost-effective for large-scale applications, avoiding per-token API costs that can become prohibitive when processing extensive KGs. To provide a performance reference and assess the gap between open-source and commercial solutions, we also include GPT-4o mini, a commercial model from OpenAI.

Gemma2:9B, developed by Google, is an open-source 9B parameter model optimized for efficiency [13], excelling in natural language understanding and generation.

[2] https://newspaper4k.readthedocs.io

[3] https://huggingface.co/datasets/FactCheck-AI/FactCheck

[4] https://github.com/FactCheck-AI/FactCheck-MockAPI

Qwen2.5:7B, from Alibaba Cloud, is an open-source 7B parameter model notable for improved instruction-following, reasoning, and structured data handling [52, 68].

LLaMA3.1:8B, by Meta, is an open-source 8B parameter model that features an extensive 128k token context window and enhanced multilingual support, making it suitable for long-context and diverse language tasks [9].

Mistral:7B, developed by Mistral AI, is a 7B parameter model known for its performance and compactness balance, demonstrated across various benchmarks [25].

GPT-4o mini, developed by OpenAI as a smaller variant of GPT-4o, offers strong reasoning capabilities with reduced latency and cost [44], serving as a commercial baseline for advanced knowledge retrieval and fact verification.

4.3 Performance Metrics and Evaluation

To assess the effectiveness of the considered fact validation strategies, we focus on two key measures: Class-wise F1 Score and Consensus Alignment. These measures are chosen to account for class imbalance, capture per-class performance, and evaluate agreement for multi-model consensus approaches. We also evaluate efficiency by computing the average response time required by each considered strategy to provide a verification response.

Class-wise F1 Scores ($F1(c)$) are calculated independently for "True" (T) and "False" (F) labels to assess performance on each single category, rather than aggregating them. This granular view highlights potential disparities in model performance between the two classes. The $F1$ score for a given class $c \in \{T, F\}$ is defined as:

$$F1(c) = \frac{2 \cdot \text{Precision}(c) \cdot \text{Recall}(c)}{\text{Precision}(c) + \text{Recall}(c)},$$

where Precision(c) and Recall(c) denote the precision and recall calculated specifically for class c.

Consensus Alignment (CA_M) quantifies the agreement between a given model's predictions and the majority vote across all evaluated facts. Specifically, for a model M, it is defined as:

$$\text{CA}_M = \frac{1}{|G|} \sum_{t \in G} \mathbb{I}(\text{response}(M,t) = \text{majorityVote}(t))$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, which evaluates to 1 if the condition is met and 0 otherwise. Here, response(M, t) represents the prediction of model M for triple t, and majorityVote(t) is the label assigned by the majority of models in the ensemble. The CA_M score ranges from 0 to 1. High CA_M identifies the "Most Representative" model serving as the best single proxy for the group's consensus, and low CA_M identifies the "Outlier" model. This indicates a model that systematically deviates from the majority opinion.

To evaluate **efficiency**, we measure the fact average response time in seconds, denoted as $\bar{\theta}$. To ensure a robust assessment that is not distorted by extreme values, we apply an outlier removal process based on the IQR method. Given a model-dataset pair, let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ be the set of model's response times over the n dataset facts. We start by computing the first $Q_1 = P_{25}(\Theta)$ and third $Q_3 = P_{75}(\Theta)$ quartiles, and then derive $\text{IQR} = Q_3 - Q_1$. Finally, we define the lower and upper bounds for acceptable values as $L_{\text{lower}} = Q_1 - 1.5 \times \text{IQR}$ and $L_{\text{upper}} = Q_3 + 1.5 \times \text{IQR}$. We exclude all response times outside these bounds, resulting in the filtered set $\Theta' = \{\theta \in \Theta \mid L_{\text{lower}} \leq \theta \leq L_{\text{upper}}\}$. The average response time per fact is then the mean response time over the filtered set, computed as: $\bar{\theta} = \frac{1}{|\Theta'|} \sum_{\theta \in \Theta'} \theta$.

Table 4: Configuration parameters used in the RAG pipeline.

RAG Component	Parameter
Human Understandable Text	Gemma2:9b
Question Generation	Gemma2:9b
Question Relevance	Jina-reranker-v1-turbo-en
Relevance Threshold	0.5
Selected Questions	3
Selected Documents (k_d)	10
Document Selection	ms-marco-MiniLM-L-6-v2
Embedding Model	bge-small-en-v1.5
Chunking Strategy	Sliding Window (size = 3)

5 Experimental Setup

This section details the technical specifications, computational infrastructure, and methodological framework used to implement `FactCheck`. We describe the hardware environments, model configurations, and procedural protocols.

To retrieve Google SERP results, we employed a Unix-based server equipped with 2 CPU cores and 4 GB of RAM. For triple transformation and question generation, we used a MacBook Pro powered by an Apple M2 Max chip with 32 GB of RAM. All other experiments involving LLMs, including prompting and evaluation, were conducted on a Mac Studio (Model: Mac14,14) equipped with an Apple M2 Ultra chip featuring 24 cores (16 performance and 8 efficiency cores) and 192 GB of unified memory.

Open-source LLMs were executed locally using *Ollama*,[5] an open-source framework that streamlines the deployment and usage of LLMs on local machines. For monitoring model behavior, including token usage and inference time, we integrated OpenTelemetry via tooling from the OpenLIT project.[6] This setup provides robust monitoring for LLMs, vector databases, and GPUs usage.

Configuration parameters for the RAG pipeline are reported in Table 4. These settings were determined through a series of experiments comparing alternative configurations. The results of these ablation studies are available in the GitHub repository.[7]

For multi-model consensus, we have two distinct experimental scenarios: one using higher-parameter open-source models, and the other using a commercial LLM, as described in §3.3. In the open-source scenario, after computing model consistency across datasets, we selected the models with the highest and lowest consistency scores. We then replaced the base versions with their larger counterparts: LLaMA3.1 (8B ⟶ 70B), Gemma2 (9B ⟶ 27B), Qwen2.5 (7B ⟶ 14B), and Mistral (7B ⟶ `nemo:12B`). In the commercial baseline scenario, we used OpenAI GPT-4o mini, providing a strong reference point for comparison with open-source alternatives.

6 Experimental Analysis

In this section, we present a comprehensive evaluation of LLM performance on the `FactCheck` benchmark, evaluating their proficiency in KG fact validation. Tables 5 and 7 report the $F1$ scores

[5] https://ollama.com/

[6] https://openlit.io/

[7] https://github.com/FactCheck-AI/FactCheck/blob/main/extra-experiments/ablation_study_results/README.md

Table 5: Performance evaluation of fact verification systems. The assessment covers various methodologies (DKA, GIV-Z, GIV-F, RAG). In each column, the best-performing method is highlighted in bold, and the second-best method is underlined.

Dataset	Method	Gemma2		Qwen2.5		Llama3.1		Mistral		GPT-4o mini	
		$F1(T)$	$F1(F)$	$F1(T)$	$F1(F)$	$F1(T)$	$F1(F)$	$F1(T)$	$F1(F)$	$F1(T)$	$F1(F)$
FactBench	DKA	0.75	0.74	0.55	0.71	0.73	0.74	0.68	0.73	0.52	0.72
	GIV-Z	0.73	0.73	0.51	0.70	0.52	0.70	0.77	0.72	0.48	0.71
	GIV-F	0.79	0.76	0.74	0.73	0.75	0.72	0.81	0.73	0.49	0.71
	RAG	**0.91**	**0.89**	**0.89**	**0.85**	**0.83**	**0.80**	**0.87**	**0.82**	**0.91**	**0.90**
Mean		0.80	0.78	0.67	0.75	0.71	0.74	0.78	0.75	0.60	0.76
YAGO	DKA	0.82	0.02	0.42	0.02	0.71	0.02	0.59	0.01	0.48	0.02
	GIV-Z	0.88	**0.03**	0.53	0.02	0.52	0.02	0.75	**0.02**	0.51	0.02
	GIV-F	**0.92**	0.02	0.72	**0.03**	0.83	0.02	0.90	0.01	0.53	0.02
	RAG	**0.92**	**0.03**	**0.92**	**0.03**	**0.91**	0.02	**0.96**	**0.02**	**0.89**	0.02
Mean		0.89	0.03	0.65	0.03	0.74	0.02	0.80	0.02	0.60	0.02
DBpedia	DKA	**0.85**	0.36	0.63	0.33	**0.81**	0.29	0.79	0.34	0.56	0.31
	GIV-Z	0.81	0.37	0.63	0.33	0.53	0.31	0.87	0.23	0.48	0.31
	GIV-F	**0.85**	0.35	0.78	0.36	0.69	0.32	**0.89**	0.20	0.36	0.30
	RAG	0.79	**0.38**	**0.82**	**0.39**	0.74	**0.33**	0.82	**0.38**	**0.75**	**0.37**
Mean		0.83	0.37	0.72	0.35	0.69	0.31	0.84	0.29	0.54	0.32

for true and false labels separately for each model on the FactBench, YAGO, and DBpedia datasets. This analysis is organized around the three key research questions introduced earlier.

RQ1. Table 5 provides an overview of the evaluation results concerning the internal knowledge capabilities of LLMs. The analysis employs three verification paradigms: Direct Knowledge Assessment (DKA), as well as Guided Iterative Verification in both zero-shot (GIV-Z) and few-shot (GIV-F) contexts.

We observe a sensible performance variability across models and datasets. In the FactBench dataset, Gemma2 achieves the robust capabilities across both classes, reaching 0.79 for $F1(T)$ and 0.76 for $F1(F)$ in the GIV-F setting. In contrast, GPT-4o mini shows a distinct performance asymmetry. While its detection of incorrect facts is comparable to other models $F1(F) \approx 0.71$, its ability to verify true facts is consistently lower $F1(T) \approx [0.48, 0.52]$. This finding challenges the prevailing view that commercial or larger models outperform smaller or open-source counterparts.

Among the datasets, FactBench appears to be the most favorable for internal knowledge evaluation, as most models maintain a reasonable balance between $F1(T)$ and $F1(F)$. On the other hand, YAGO proves to be the most challenging due to its large number of correct facts. While models achieve high $F1(T)$ scores (up to 0.92), the $F1(F)$ scores are negligible (0.01 to 0.03). This drastic discrepancy indicates a strong model bias toward positive classifications, which hinders the detection of rare incorrect facts in highly imbalanced contexts. In comparison, DBpedia yields intermediate results; most models achieve respectable $F1(T)$ scores $[0.53, 0.89]$, yet they struggle to reliably identify incorrect information, with $F1(F)$ values generally remaining below 0.40.

Notably, the few-shot setup (GIV-F) consistently outperforms both DKA and GIV-Z settings. For instance, on FactBench, Mistral improves from 0.68 (DKA) to 0.81 (GIV-Z), while its performance on false claims remains stable around 0.73.. These gains are particularly pronounced for mid-tier models, which benefit more from structured prompting and exemplar-based guidance. By contrast, already well-performing models such as Gemma2 show relatively smaller performance gains.

Finding 1: Open-source models, such as Gemma2 or Mistral, outperform commercial alternatives like GPT-4o mini when relying exclusively on internal knowledge. Moreover, few-shot prompting consistently enhances performance, although the degree of improvement is influenced by dataset characteristics such as class balance and label distribution.

RQ2. We evaluate the performance of the RAG methodology across all models and datasets, and then compare it against the internal knowledge-based approaches in Table 5.

Overall, RAG achieves the highest performance across nearly all experimental settings. In particular, for the FactBench dataset, RAG delivers substantial improvements: for example, Qwen2.5 achieves a $F1(T)$ of 0.89, compared to 0.55 in the DKA setting. This trend holds across evaluated models, including GPT-4o mini, which shows a marked increase in performance – rising more than 25% in both $F1$ scores – when external evidence is incorporated.

However, the impact of RAG varies significantly across datasets. FactBench and YAGO show the greatest absolute gains, likely due to their broader diversity of factual content. In contrast, Dbpedia exhibits minimal improvements or even slight performance degradation in some cases. This may be attributed to schema diversity, which can complicate the retrieval process and diminish the relevance of the extracted evidence.

Finding 2: Incorporating external evidence via RAG represents a promising path to high-accuracy fact validation. However, its effectiveness is dependent on dataset characteristics.

RQ3. We investigate the effectiveness of multi-model consensus strategies, applying majority voting across our four open-source models. In cases of ties, we introduce a tie-breaking mechanism using either higher-parameter variants or a commercial model (GPT-4o mini). Table 7 summarizes the results.

Multi-model consensus provides more reliable performance across internal knowledge settings (DKA, GIV-Z, and GIV-F), although it does not consistently outperform all individual models. In many cases, it stabilizes performance across varying conditions rather than providing top results. Interestingly, the choice of tie-breaking model has minimal influence on final performance. Whether we use the most consistent model (agg-cons-up), the

Table 6: Model alignment analysis across fact validation methodologies and datasets. Consensus Alignment (CA_M) measure the percentage agreement between LLM predictions and majority vote decisions, with highest and lowest performing models highlighted for each method-dataset combination. Tie percentages indicate the frequency of split decisions requiring arbitration.

Dataset	Method	Ties	Gemma2	Qwen2.5	Llama3.1	Mistral
FactBench	DKA	16%	0.919	0.861	0.906	0.938
	GIV-Z	21%	0.914	0.893	0.913	0.814
	GIV-F	14%	0.937	0.861	0.901	0.909
	RAG	6%	0.968	0.970	0.897	0.960
YAGO	DKA	19%	0.798	0.797	0.916	0.920
	GIV-Z	26%	0.790	0.872	0.859	0.886
	GIV-F	16%	0.934	0.771	0.901	0.944
	RAG	6%	0.968	0.969	0.916	0.974
DBpedia	DKA	17%	0.937	0.772	0.891	0.920
	GIV-Z	24%	0.948	0.875	0.765	0.758
	GIV-F	17%	0.960	0.879	0.779	0.876
	RAG	9%	0.953	0.961	0.848	0.945

Table 7: Performance evaluation of fact verification systems. The assessment covers multi-model consensus. In each column, the best-performing method is highlighted in bold, and the second-best method is underlined.

Dataset	Method	agg-cons up (Refer to Tab.6)		agg-cons down (Refer to Tab.6)		agg-GPT-4o mini	
		$F1(T)$	$F1(F)$	$F1(T)$	$F1(F)$	$F1(T)$	$F1(F)$
FactBench	DKA	0.68	0.75	0.69	0.75	0.69	0.75
	GIV-Z	0.74	0.76	0.64	0.74	0.63	0.74
	GIV-F	0.82	0.78	0.81	0.79	0.80	0.79
	RAG	**0.91**	**0.89**	**0.91**	**0.89**	**0.91**	**0.89**
Mean		0.79	0.80	0.76	0.79	0.76	0.79
YAGO	DKA	0.59	**0.02**	0.63	**0.02**	0.61	**0.02**
	GIV-Z	0.63	**0.02**	0.73	**0.02**	0.65	**0.02**
	GIV-F	0.84	**0.02**	0.84	**0.02**	0.84	**0.02**
	RAG	**0.93**	**0.02**	**0.94**	**0.02**	**0.93**	**0.02**
Mean		0.75	0.02	0.78	0.02	0.76	0.02
DBpedia	DKA	0.84	0.37	0.80	0.37	0.78	0.37
	GIV-Z	0.77	0.38	0.73	0.36	0.71	0.36
	GIV-F	**0.85**	**0.40**	**0.86**	**0.39**	**0.81**	0.38
	RAG	0.80	0.39	0.81	**0.39**	0.80	**0.39**
Mean		0.81	0.39	0.80	0.38	0.77	0.38

least consistent model (agg-cons-down), or GPT-4o mini, the resulting scores remain nearly identical across all datasets and methods. This suggests that the majority vote mechanism effectively captures the most reliable signal, and the specific choice of arbitrator is less impactful than having a consistent tie-resolution strategy in place.

Our consistency analysis, shown in Table 6, further reveals that agreement among models increases with methodological complexity. For instance, RAG results in lower tie rates – ranging from 6% to 9% – compared to 21% to 26% in GIV-Z. This reinforces the notion that external evidence not only improves individual model performance but also enhances cross-model alignment. However, this increased agreement may also reflect a stronger influence of shared contextual evidence, potentially reducing reliance on internal knowledge and thereby introducing uniformity at the cost of model individuality or specificity.

Table 8: Execution time ($\bar{\theta}$, in seconds) for fact validation across different methodologies (DKA, GIV-Z, GIV-F, and RAG). The fastest configuration is highlighted in green, while the slowest configuration is marked in red.

Dataset	Method	Gemma2	Qwen2.5	Llama3.1	Mistral
FactBench	DKA	0.21	0.18	0.30	0.17
	GIV-Z	0.62	0.40	0.50	0.45
	GIV-F	0.78	0.51	0.67	0.65
	RAG	2.27	2.39	2.73	1.69
YAGO	DKA	0.22	0.19	0.31	0.19
	GIV-Z	0.62	0.41	0.45	0.47
	GIV-F	0.78	0.54	0.69	0.67
	RAG	2.10	2.39	2.68	1.63
DBpedia	DKA	0.35	0.25	0.37	0.24
	GIV-Z	0.70	0.43	0.58	0.53
	GIV-F	0.89	0.56	0.69	0.78
	RAG	2.55	2.55	2.87	1.77

Finding 3: Multi-model consensus offers a simple yet robust mechanism to stabilize fact validation performance. While it does not always outperform individual models, it mitigates the impact of weaker ones. The specific choice of arbitrator has a limited impact. Moreover, external evidence promotes greater model alignment, though care must be taken to avoid overfitting to contextual bias.

Computational Efficiency. Beyond accuracy metrics, we evaluate the computational efficiency of different approaches. Table 8 reports execution times ($\bar{\theta}$, in seconds) for fact validation using the four open-source LLMs across the three reference datasets. Within each dataset, DKA yields the lowest execution times, ranging from 0.21 to 0.30 seconds on FactBench, from 0.19 to 0.31 seconds on YAGO, and from 0.24 to 0.37 seconds on DBpedia. GIV-Z shows an increase over DKA, with approximately double the execution time on FactBench and YAGO, such as an increase from 0.18 to 0.40 seconds on Qwen2.5 for FactBench. GIV-F requires more time than GIV-Z, with values reaching up to 0.78 seconds. RAG results in the highest execution times across all datasets and models, with values including 2.73 seconds on Llama3.1 for FactBench and over 2.5 seconds for several models on DBpedia.

The comparison within each dataset indicates that, as expected, RAG incurs the greatest computational cost, often exceeding DKA by a factor of six or more. The increase in execution time follows the progression from DKA to GIV-Z to GIV-F to RAG in all configurations. This pattern suggests a direct relationship between the methodological complexity of the verification strategy and its computational cost.

On a different note, multi-model consensus can be parallelized, meaning that inference latency is bounded by the slowest model rather than the sum of all models. In practice, if models exhibit varying response times (e.g., 0.3–0.5 seconds), consensus inference requires waiting for the slowest response, resulting in slightly higher latency compared to selecting only the fastest model. Tie-breaking further adds inference overhead, as it requires an additional model query. Moreover, the coordination and resource allocation across multiple models introduce minor but non-negligible computational overhead. Despite this, consensus brings benefits: the trustworthiness of the predictions increases due to the aggregation of diverse model perspectives.

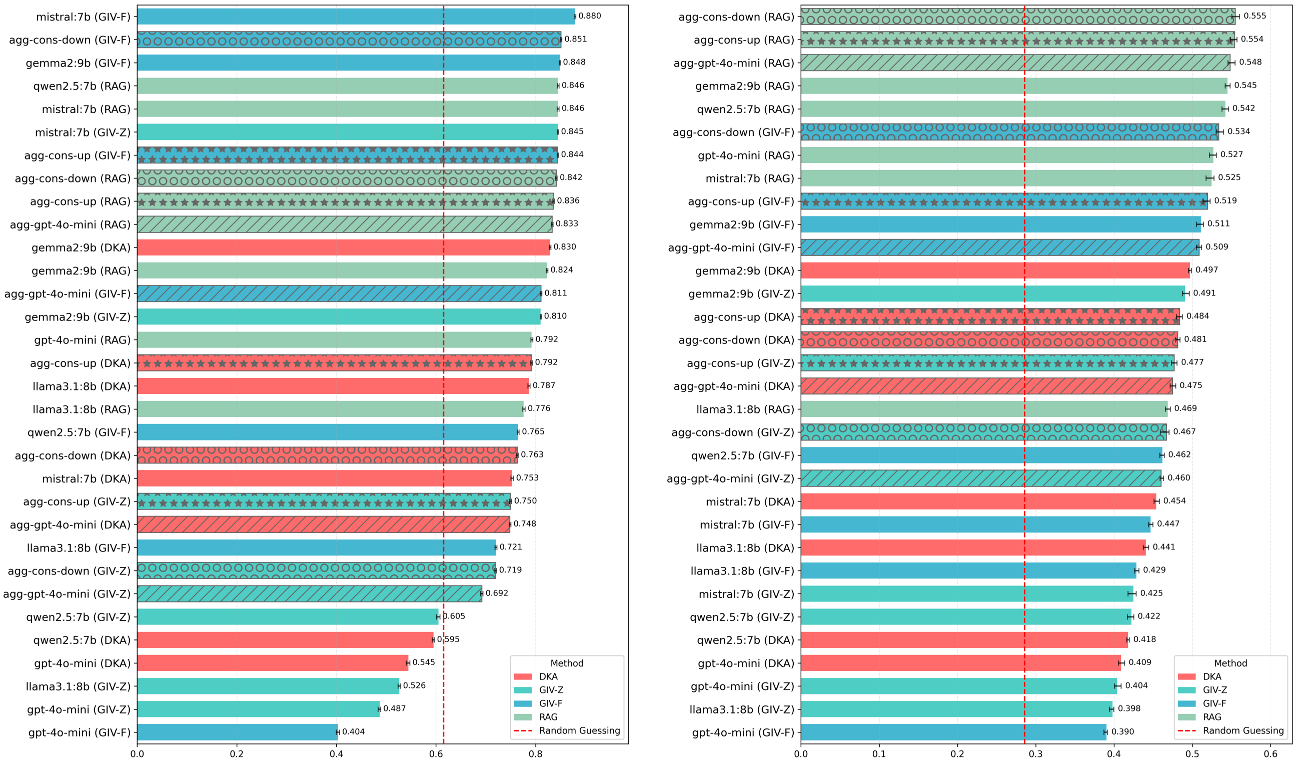


Figure 2: $F1$ scores for FactCheck benchmark. The left plot displays $F1(T)$ scores, and the right plot displays $F1(F)$ scores. Multi-model consensus results are shown with hatching, and the red dotted line indicates the guess rate.

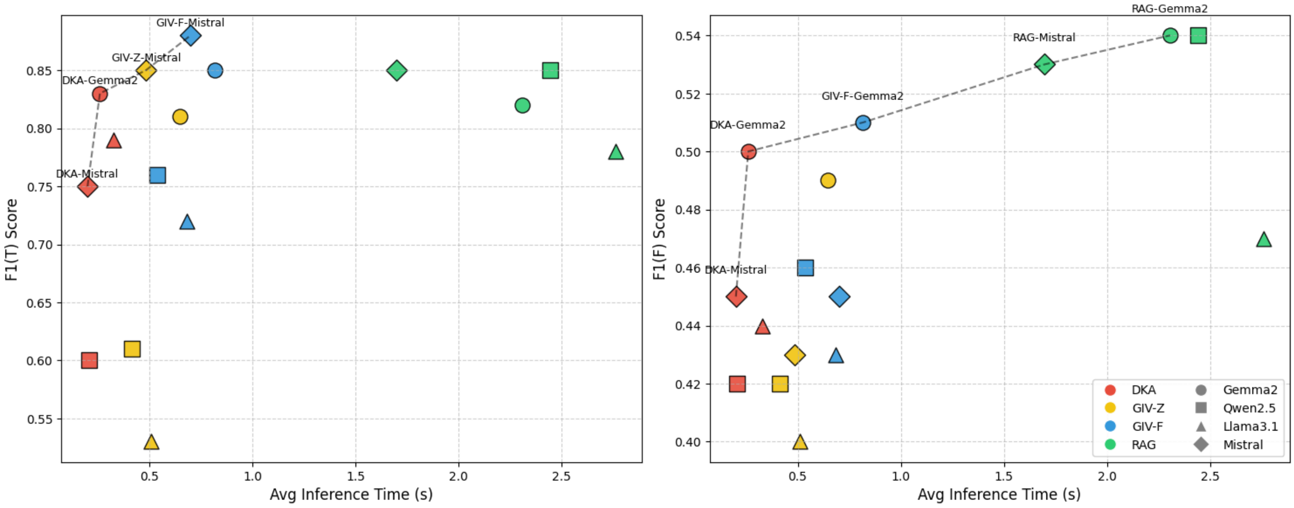


Figure 3: Trade-off analysis between computational cost ($\bar{\theta}$) and verification performance ($F1(F)$ and $F1(T)$). The dashed line represents the Pareto frontier, highlighting configurations that achieve optimal efficiency (highest accuracy for a given time budget).

To characterize the balance between predictive accuracy and computational expense, we examined the Pareto efficiency of our methods across the different models (Figure 3). This analysis reveals a clear separation in the utility of each strategy: RAG-based techniques generally cluster in the upper-right quadrant, especially with respect to the $F1(F)$ metric, indicating that their increased latency (≈1.6s–2.9s) is exchanged for enhanced detection of false claims. Conversely, DKA setups dominate the high-speed regime, delivering sub-second inference times (<0.3 s) that are appropriate for latency-sensitive use cases, albeit with lower sensitivity. The Pareto frontier indicates that mid-range approaches such as GIV-F (particularly when paired with Gemma2 and Mistral) strike an attractive trade-off, attaining competitive accuracy – at times even exceeding RAG on the $F1(T)$ metric – while incurring substantially less computational cost than full retrieval-based systems.

Finding 4: Computational efficiency varies widely across methods. On the one hand, RAG requires up to 10× more processing time compared to internal knowledge approaches. On the other hand, consensus strategies can be parallelized to ensure only modest latency increases with respect to internal knowledge methods.

Cross-Dataset Generalization and Stability. To assess the generalization capabilities and stability of LLM-based fact validation, we analyze the performance across different methods and aggregation strategies, which are visualized in the bar charts (Figure 2). The plots display the $F1$ scores for the True class (left chart) and False class (right chart) ranked by performance. The red dashed line represents the Random Guessing baseline, which sits at approximately 0.62 for $F1(T)$ and 0.29 for $F1(F)$, and this reflects the underlying class distribution challenges in the dataset.

RAG demonstrates the most consistent robustness. In the $F1(F)$ chart, which typically represents the harder task of identifying incorrect facts, RAG-based methods and their aggregations dominate the top rankings. On the other hand, GIV-F (blue bars) exhibits high variance. Although Mistral (GIV-F) achieves the absolute highest peak in the $F1(T)$ chart (0.88), other models using the same strategy, such as gpt-4o-mini, perform drastically lower at 0.40. This result falls significantly below the random guessing baseline and suggests that while GIV-F can prompt high recall for true facts in specific models, it lacks the stability of RAG. The DKA (red bars) methodology generally occupies the middle-to-lower tier, particularly in the $F1(F)$ analysis, which indicates that reliance on internal parametric knowledge alone is often insufficient for distinguishing false claims. Finally, the aggregation methods denoted as "agg-cons-∗" consistently appear in the upper echelons of both charts. This confirms that ensemble reasoning, specifically majority voting strategies, effectively mitigates the volatility of individual models and smoothes out the noise observed in strategies like GIV-Z and GIV-F.

Finding 5: RAG offers the strongest cross-dataset generalization, consistently outperforming internal knowledge methods in detecting false claims. Some GIV-F models reach top performance on True facts but are highly volatile. Notably, several internal knowledge methods perform below Random Guessing, showing that poor methodology can degrade reasoning to below a coin-flip baseline. Thus, consensus-based aggregation remains essential for stability and reducing model-specific bias.

7 Qualitative Error Analysis

For our error analysis, we categorize mistakes from open-source models using a semi-automated pipeline combining LLM-generated reasoning with contextual document embeddings. We collect logs of incorrect predictions and prompt the same LLM to explain each error. Then, we encode these explanations using the `cde-small-v1` model [39] and cluster them using UMAP for dimensionality reduction followed by HDBSCAN [6] to find clusters of varying densities. Finally, we assign descriptive labels to each cluster. The resulting error categories are: Unlabeled (E1): The supplied context is missing the asserted details or mentions of the relevant entities. Relationship Errors (E2): The model provides incorrect information about relationships between individuals, such as marital status or religious affiliation. Role Attribution Errors (E3): The model wrongly links people to particular roles, locations, or teams. Geographic/Nationality Errors (E4): Information about places or national affiliations is inconsistent with the context. Genre/Classification Errors (E5): The model miscategorizes movies, genres, or creative works connected to individuals or studios. Identifier/Biographical Errors (E6): Identifiers or biographical fact, such as award names, are inaccurate.

Table 9: Dataset-wise error clustering based on LLM-generated reasoning.

Dataset	Model	E1	E2	E3	E4	E5	E6	Total*
FactBench	Gemma2	4	36	45	176	13	1	275
	Qwen2.5	33	27	60	194	34	1	349
	Llama3.1	38	44	73	295	38	3	491
	Mistral	53	27	53	242	40	2	417
Unique. Ratio (%)		0.62	0.72	0.44	0.52	0.63	0.57	0.53
YAGO	Gemma2	6	134	0	14	51	2	207
	Qwen2.5	7	109	0	13	63	2	194
	Llama3.1	8	98	0	19	104	2	231
	Mistral	7	54	0	10	34	3	108
Unique. Ratio (%)		0.35	0.52	–	0.46	0.51	0.33	0.50
DBpedia	Gemma2	353	22	98	1729	459	299	2960
	Qwen2.5	339	19	91	1525	357	237	2568
	Llama3.1	382	28	109	2172	509	318	3518
	Mistral	325	20	94	1487	438	241	2605
Unique. Ratio (%)		0.41	0.43	0.44	0.42	0.42	0.40	0.41

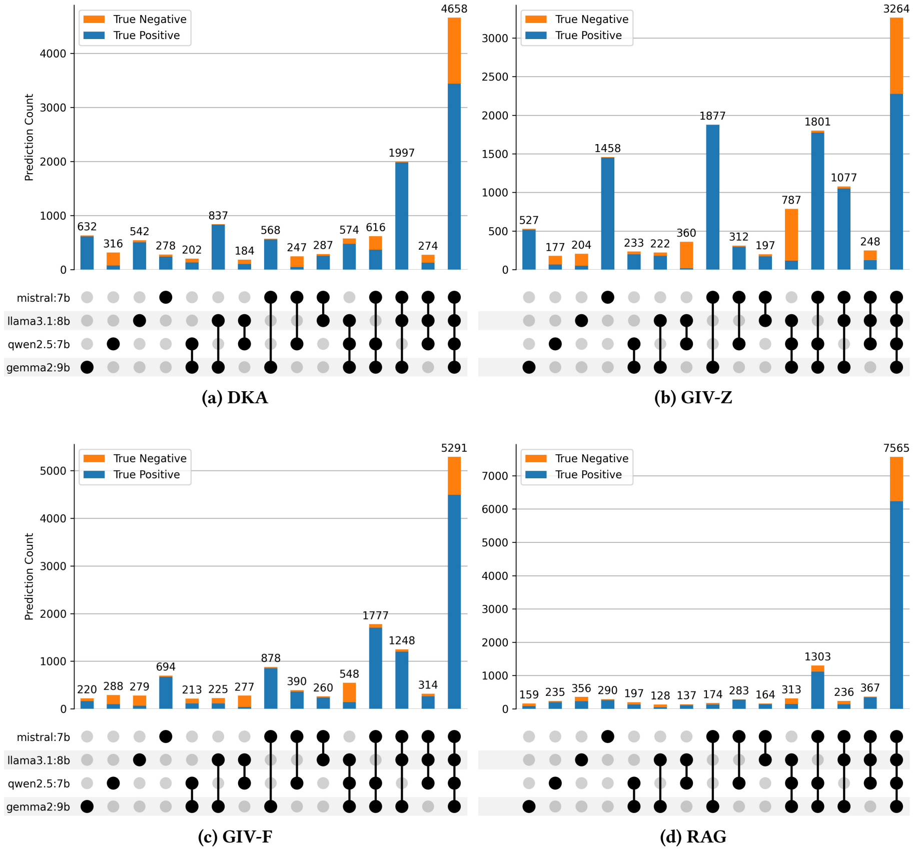
Table 9 shows the count of each error type on the evaluated datasets. As shown in Table 9, E4 errors form the predominant challenge in `FactCheck`. In addition, we extended this analysis on the DBpedia dataset using the stratification and topic modeling from Marchesin et al. [38] to understand the impact of fact popularity and domain. The results reveal that error rates decrease in partitions representing common knowledge and domains like "Education" and "News" yield lower error rates, while "Architecture" and "Transportation" remain more challenging. The entire verification process and the error analysis presented here can be interactively interpreted and visualized using our web-based platform available at https://factcheck.dei.unipd.it/ [55].

To study how the models complement each other, we examined overlaps in their predictions using UpSet plots [35]. As illustrated in Figure 4, the largest intersection generally corresponds to facts correctly predicted by all four models, indicating that open-source LLMs share much of their internal knowledge as well as their error profiles. This agreement is most pronounced in the RAG setting, where common external evidence steers the models toward the same conclusions, thereby reducing variance.

GIV-Z, however, departs from this pattern: the "all-model" intersection shrinks markedly relative to DKA (from roughly 4,600 to about 3,200) and is replaced by stronger pairwise overlaps (e.g., between Qwen2.5 and Gemma2). This pattern suggests that zero-shot prompting leads to more heterogeneous reasoning trajectories and greater disagreement among models. In contrast, GIV-F restores stronger consensus, raising the all-model intersection to over 5,200, indicating that few-shot demonstrations effectively harmonize model behavior. Overall, the limited true complementarity among models may explain why consensus methods stabilize predictions but rarely outperform the best single model.

8 Final Remarks

In this work, we introduced `FactCheck`, a benchmark for systematically evaluating LLMs in KG fact validation. Our evaluations



(a) DKA

(b) GIV-Z

(c) GIV-F

(d) RAG

Figure 4: Intersection of correct predictions across models. Bars show the number of correct samples by the specific combination of models indicated by the connected dots below.

on three real-world datasets included in FactCheck– FactBench, YAGO, and DBpedia – yielded several key findings. First, open-source LLMs, such as Gemma2, achieve promising verification performance, with $F1$ scores up to 0.79 and 0.76 using internal knowledge alone and exceeding 0.89 when augmented with RAG. Second, RAG improves performance across most settings, though at a significant computational cost – being roughly 10× slower than other methods. Third, multi-model consensus mitigates errors and provides more reliable responses than single-model predictions, in particular when relying on internal knowledge.

At the same time, we also identified several limitations: (1) dataset-specific challenges, such as class imbalance in YAGO and schema diversity in DBpedia; (2) infrastructure constraints, including a 0.08% retrieval failure rate due to network issues and regional restrictions; and (3) content filtering in hosted deployments, such as blocked factual content on sensitive topics for Azure’s GPT-4o-mini.

Hence, FactCheck advances the study of LLMs factual reasoning by leveraging the structured semantics of KGs, unlike prior benchmarks focused on unstructured claims or general-domain QA. It provides a controlled environment for reproducible, fine-grained analyses of model behavior, including internal knowledge use, retrieval effectiveness, and multi-model interactions. As a robust testbed, FactCheck supports the development of new prompting strategies, model architectures, and retrieval techniques for fact validation. By releasing it publicly, we aim to promote transparency, collaboration, and faster progress toward trustworthy, scalable KG validation systems.

Looking ahead, our findings suggest several promising research directions. First, fine-tuning or pretraining LLMs for KG fact validation could help mitigate limitations from imbalanced datasets. Second, hybrid retrieval strategies that combine structured KG traversal with unstructured web data may enhance retrieval quality, particularly for datasets like DBpedia. Finally, the benchmark can be extended to support the evaluation of fact-verification systems that also leverage logical rules in the KG, for example by exploiting the ontologies on which the KG is based (e.g., using transitivity, domain/range constraints, and other properties to assess the correctness and reliability of triples).

Acknowledgments

This work is partially supported by the HEREDITARY Project, as part of the European Union’s Horizon Europe research and innovation program under grant agreement No. GA 101137074. The authors thank Andrea Segala for contributing to the experiments on zero-shot and few-shot prompting during his master’s thesis.

Artifacts

The source code and datasets have been made publicly available at https://github.com/FactCheck-AI/ and https://huggingface.co/FactCheck-AI.

References

[1] Firoj Alam, Julia Maria Struß, Tanmoy Chakraborty, Stefan Dietze, Salim Hafid, Katerina Korre, Arianna Muti, Preslav Nakov, Federico Ruggeri, Sebastian Schellhammer, et al. 2025. The CLEF-2025 CheckThat! Lab: Subjectivity, Fact-Checking, Claim Normalization, and Retrieval. In *European Conference on Information Retrieval*. Springer, 467–478.

[2] Fatma Arslan, Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2020. A benchmark dataset of check-worthy factual claims. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 821–829.

[3] Luigi Bellomarini, Daniele Fakhoury, Georg Gottlob, and Emanuel Sallinger. 2019. Knowledge Graphs and Enterprise AI: The Promise of an Enabling Technology. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 26–37. doi:10.1109/ICDE.2019.00011

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. *Proc. Sigmod*, 1247–1250. doi:10.1145/1376616.1376746

[5] Jack Boylan, Shashank Mangla, Dominic Thorn, Demian Gholipour Ghalandari, Parsa Ghaffari, and Chris Hokamp. 2024. KGValidator: A Framework for Automatic Validation of Knowledge Graph Construction. arXiv:2404.15923 [cs.AI] https://arxiv.org/abs/2404.15923

[6] R. J. G. B. Campello, D. Moulavi, and J. Sander. 2013. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, Berlin, Heidelberg, 160–172.

[7] Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Dingqi Yang, Hailong Sun, and Philip S. Yu. 2025. Harnessing Multiple Large Language Models: A Survey on LLM Ensemble. *CoRR* abs/2502.18036 (February 2025). https://doi.org/10.48550/arXiv.2502.18036

[8] O. Deshpande, D. S. Lamba, M. Tourn, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan. 2013. Building, maintaining, and using knowledge bases: a report from the trenches. In *Proc. of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*. ACM, 1209–1220. doi:10.1145/2463676.2465297

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The Llama 3 Herd of Models. *CoRR* abs/2407.21783 (2024). https://doi.org/10.48550/arXiv.2407.21783

[10] Basil Ell, Andreas Harth, and Elena Simperl. 2014. SPARQL Query Verbalization for Explaining Semantic Search Engine Queries. In *The Semantic Web: Trends and Challenges*, Valentina Presutti, Claudia d'Amato, Fabien Gandon, Mathieu d'Aquin, Steffen Staab, and Anna Tordai (Eds.). Springer International Publishing, Cham, 426–441.

[11] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 413–422. doi:10.1145/2488388.2488425

[12] J. Gao, X. Li, Y. E. Xu, B. Sisman, X. L. Dong, and J. Yang. 2019. Efficient Knowledge Graph Accuracy Evaluation. *Proc. VLDB Endow.* 12, 11 (2019), 1679–1691. doi:10.14778/3342263.3342642

[13] Gemma-Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118 [cs.CL] https://arxiv.org/abs/2408.00118

[14] Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. 2015. DeFacto—Temporal and multilingual Deep Fact Validation. *Journal of Web Semantics* 35 (2015), 85–101. doi:10.1016/j.websem.2015.08.001 Machine Learning and Data Mining for the Semantic Web (MLDMSW).

[15] Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 Task 7: RumourEval, Determining Rumour Veracity and Support for Rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, Jonathan May, Ekaterina Shutova, Aurelie Herbelot, Xiaodan Zhu, Marianna Apidianaki, and Saif M. Mohammad (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, USA, 845–854. doi:10.18653/v1/S19-2147

[16] R. Guha, Rob McCool, and Eric Miller. 2003. Semantic search. In *Proceedings of the 12th International Conference on World Wide Web* (Budapest, Hungary) *(WWW '03)*. Association for Computing Machinery, New York, NY, USA, 700–709. doi:10.1145/775152.775250

[17] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the association for computational linguistics* 10 (2022), 178–206.

[18] Qi He, Bee-Chung Chen, and Deepak Agarwal. 2016. *Building the LinkedIn Knowledge Graph*. LinkedIn Engineering. https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph Accessed: 2025-04-16.

[19] Qiyuan He, Yizhong Wang, and Wenya Wang. 2024. Can Language Models Act as Knowledge Bases at Scale? *CoRR* (2024).

[20] Qiyuan He, Yizhong Wang, Jianfei Yu, and Wenya Wang. 2025. Language Models over Large-Scale Knowledge Base: on Capacity, Flexibility and Reasoning for New Facts. In *Proceedings of the 31st International Conference on Computational Linguistics*, Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (Eds.). Association for Computational Linguistics, Abu Dhabi, UAE, 1736–1753. https://aclanthology.org/2025.coling-main.118/

[21] Cory Henson, Stefan Schmid, Anh Tuan Tran, and Antonios Karatzoglou. 2019. Using a Knowledge Graph of Scenes to Enable Search of Autonomous Driving Data.. In *ISWC (Satellites)*. 313–314.

[22] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th International Conference Companion on World Wide Web* (Hyderabad, India) *(WWW '11)*. Association for Computing Machinery, New York, NY, USA, 229–232. doi:10.1145/1963192.1963296

[23] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge Graphs. *ACM Comput. Surv.* 54, 4, Article 71 (July 2021), 37 pages. doi:10.1145/3447772

[24] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip Yu. 2021. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE transactions on neural networks and learning systems* PP (04 2021). doi:10.1109/TNNLS.2021.3070843

[25] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[26] Xuhui Jiang, Chengjin Xu, Yinghan Shen, Xun Sun, Lumingyuan Tang, Saizhuo Wang, Zhongwu Chen, Yuanzhuo Wang, and Jian Guo. 2023. On the Evolution

of Knowledge Graphs: A Survey and Perspective. arXiv:2310.04835 [cs.AI] https://arxiv.org/abs/2310.04835

[27] Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When Can LLMs Actually Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs. *Transactions of the Association for Computational Linguistics* 12 (2024), 1417–1440. doi:10.1162/tacl_a_00713

[28] Mohammed Abdul Khaliq, Paul Yu-Chun Chang, Mingyang Ma, Bernhard Pflugfelder, and Filip Miletić. 2024. Ragar, your falsehood radar: Rag-augmented reasoning for political fact-checking using multimodal large language models. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*. 280–296.

[29] Jiseong Kim and Key-sun Choi. 2020. Unsupervised Fact Checking by Counter-Weighted Positive and Negative Evidential Paths in A Knowledge Graph. In *Proceedings of the 28th International Conference on Computational Linguistics*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 1677–1686. doi:10.18653/v1/2020.coling-main.147

[30] Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023. FactKG: Fact Verification via Reasoning on Knowledge Graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 16190–16206. doi:10.18653/v1/2023.acl-long.895

[31] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. Training Language Models to Self-Correct via Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=CjwERcAU7w

[32] Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. LLMs Get Lost In Multi-Turn Conversation. arXiv:2505.06120 [cs.CL] https://arxiv.org/abs/2505.06120

[33] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal* 6 (01 2014). doi:10.3233/SW-140134

[34] Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. 2024. Long Context RAG Performance of Large Language Models. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*. https://openreview.net/forum?id=Le9anH3kv1

[35] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. 2014. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 1983–1992. doi:10.1109/TVCG.2014.2346248

[36] Stefano Marchesin and Gianmaria Silvello. 2024. Efficient and Reliable Estimation of Knowledge Graph Accuracy. *Proc. VLDB Endow.* 17, 9 (2024), 2392–2404. doi:10.14778/3665844.3665865

[37] Stefano Marchesin and Gianmaria Silvello. 2025. Credible Intervals for Knowledge Graph Accuracy Estimation. *Proc. ACM Manag. Data (SIGMOD)* 3, 3, Article 142 (2025), 26 pages. doi:10.1145/3725279

[38] Stefano Marchesin, Gianmaria Silvello, and Omar Alonso. 2024. Utility-Oriented Knowledge Graph Accuracy Estimation with Limited Annotations: A Case Study on DBpedia. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 12, 1 (Oct. 2024), 105–114. doi:10.1609/hcomp.v12i1.31605

[39] John Xavier Morris and Alexander M Rush. 2025. Contextual Document Embeddings. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=Wqsk3FbD6D

[40] Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. 2013. Sorry, i don't speak SPARQL: translating SPARQL queries into natural language. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 977–988. doi:10.1145/2488388.2488473

[41] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale Knowledge Graphs: Lessons and Challenges: Five diverse technology companies show how it's done. *Queue* 17, 2 (April 2019), 48–75. doi:10.1145/3329781.3332266

[42] Allard Oelen, Markus Stocker, and Sören Auer. 2020. Creating a Scholarly Knowledge Graph from Survey Article Tables. In *Digital Libraries at Times of Massive Societal Transition*, Emi Ishita, Natalie Lee San Pang, and Lihong Zhou (Eds.). Springer International Publishing, Cham, 373–389.

[43] Prakhar Ojha and Partha Talukdar. 2017. KGEval: Accuracy Estimation of Automatically Constructed Knowledge Graphs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, Copenhagen, Denmark, 1741–1750. doi:10.18653/v1/D17-1183

[44] OpenAI. 2024. GPT-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2025-07-05.

[45] Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. 2018. Robust Discovery of Positive and Negative Rules in Knowledge Bases. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. 1168–1179. doi:10.1109/ICDE.2018.00108

[46] Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. 2023. Large Language Models and Knowledge Graphs: Opportunities and Challenges. *Transactions on Graph Data and Knowledge* 1, 1 (2023), 2:1–2:38. doi:10.4230/TGDK.1.1.2

[47] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial intelligence review* 56, 11 (2023), 13071–13102.

[48] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases?. 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)*. 2463–2473.

[49] R. J. Pittman. 2017. *Cracking the Code on Conversational Commerce*. eBay Inc. https://www.ebayinc.com/stories/news/cracking-the-code-on-conversational-commerce/ Accessed: 2025-04-16.

[50] J. Pujara, E. Augustine, and L. Getoor. 2017. Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. ACL, 1751–1756. doi:10.18653/v1/d17-1184

[51] Umair Qudus, Michael Röder, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. 2025. Fact Checking Knowledge Graphs – A Survey. *ACM Comput. Surv.* (July 2025). doi:10.1145/3749838 Just Accepted.

[52] Qwen-Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/

[53] Daniel Russo, Stefano Menini, Jacopo Staiano, and Marco Guerini. 2024. Face the Facts! Evaluating RAG-based Fact-checking Pipelines in Realistic Settings. *CoRR* abs/2412.15189 (2024). https://doi.org/10.48550/arXiv.2412.15189

[54] Philipp Schoenegger, Indre Tuminauskaite, Peter S. Park, Rafael Valdece Sousa Bastos, and Philip E. Tetlock. 2024. Wisdom of the silicon crowd: LLM ensemble prediction capabilities rival human crowd accuracy. *Science Advances* 10, 45 (2024), eadp1528. doi:10.1126/sciadv.adp1528

[55] Farzad Shami, Stefano Marchesin, and Gianmaria Silvello. 2025. Fact Verification in Knowledge Graphs Using LLMs. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 3985–3989. doi:10.1145/3726302.3730142

[56] Chetan Sharma and Jan Overgoor. 2018. *Scaling Knowledge Access and Retrieval at Airbnb*. The Airbnb Tech Blog. https://medium.com/airbnb-engineering/scaling-knowledge-access-and-retrieval-at-airbnb-665b6ba21e95 Accessed: 2025-04-16.

[57] Baoxu Shi and Tim Weninger. 2016. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-Based Systems* 104 (2016), 123–133. doi:10.1016/j.knosys.2016.04.015

[58] Prashant Shiralkar, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2017. Finding Streams in Knowledge Graphs to Support Fact Checking. In *2017 IEEE International Conference on Data Mining (ICDM)*. 859–864. doi:10.1109/ICDM.2017.105

[59] Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs?. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 311–325. doi:10.18653/v1/2024.naacl-long.18

[60] Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs?. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (Eds.). Association for Computational Linguistics, 311–325. doi:10.18653/V1/2024.NAACL-LONG.18

[61] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2019. Unsupervised Discovery of Corroborative Paths for Fact Validation. In *The Semantic Web – ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I* (Auckland, New Zealand). Springer-Verlag, Berlin, Heidelberg, 630–646. doi:10.1007/978-3-030-30793-6_36

[62] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2018. FactCheck: Validating RDF Triples Using Textual Evidence. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1599–1602. doi:10.1145/3269206.3269308

[63] Zafar Habeeb Syed, Nikit Srivastava, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2019. COPAAL – An Interface for Explaining Facts using Corroborative Paths. In *Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas)*, Mari Carmen Suárez-Figueroa, Gong Cheng, Anna Lisa Gentile, Christophe Guéret, Maria Keet, and Abraham Bernstein (Eds.), Vol. 2456. Springer International Publishing, 201–204. https://papers.dice-research.org/2019/ISWC2019_COPAAL_Demo/public.pdf

[64] Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2025. Reasoning Aware Self-Consistency: Leveraging Reasoning Paths for Efficient LLM Sampling. 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)*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 3613–3635. doi:10.18653/v1/2025.naacl-long.184

[65] Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language Models are Open Knowledge Graphs. arXiv:2010.11967 [cs.CL] https://arxiv.org/abs/2010.11967

[66] Magdalena Wysocka, Oskar Wysocki, Maxime Delmas, Vincent Mutel, and André Freitas. 2024. Large Language Models, scientific knowledge and factuality: A framework to streamline human expert evaluation. *Journal of Biomedical Informatics* 158 (2024), 104724. doi:10.1016/j.jbi.2024.104724

[67] Mingfeng Xue, Dayiheng Liu, Wenqiang Lei, Xingzhang Ren, Baosong Yang, Jun Xie, Yidan Zhang, Dezhong Peng, and Jiancheng Lv. 2023. Dynamic Voting for Efficient Reasoning in Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 3085–3104. doi:10.18653/v1/2023.findings-emnlp.203

[68] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671* (2024).

[69] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen-tau Yih, and Xin Luna Dong. 2024. CRAG - Comprehensive RAG Benchmark. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 10470–10490. https://proceedings.neurips.cc/paper_files/paper/2024/file/1435d2d0fca85a84d83ddcb754f58c29-Paper-Datasets_and_Benchmarks_Track.pdf

[70] Zhenrui Yue, Huimin Zeng, Lanyu Shang, Yifan Liu, Yang Zhang, and Dong Wang. 2024. Retrieval Augmented Fact Verification by Synthesizing Contrastive Arguments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 10331–10343. doi:10.18653/v1/2024.acl-long.556

[71] Danna Zheng, Mirella Lapata, and Jeff Z. Pan. 2024. How Reliable are LLMs as Knowledge Bases? Re-thinking Facutality and Consistency. arXiv:2407.13578 [cs.CL] https://arxiv.org/abs/2407.13578

[72] Zhengzuo Zhengzuo, Zhengzuo Liu, Lanyu Li, Ling Fu, Jing Li, Tianrui Sun, and Xiaonan Wang. 2023. Knowledge Graph for Low Carbon Power and Energy Systems. doi:10.46855/energy-proceedings-10361