State of the Art Report

Multi-Agent System for Automated Fact-Checking of YouTube Videos

Begoña Echavarren Sánchez

Tutor: Josep-Anton Mir Tutusaus

Master's Degree in Data Science
Universitat Oberta de Catalunya

PEC 2 - State of the Art

November 2025

Contents

1 Introduction

Misinformation on digital platforms has become a major challenge in the information age. YouTube, with over 2 billion monthly users and 500 hours of video uploaded every minute, is a primary information source for millions worldwide. However, without effective verification mechanisms, false or misleading claims spread at massive scale, impacting public health, democratic processes, and social cohesion [1, 2, 28].

Automated fact-checking has evolved significantly since the mid-2010s, from rule-based systems to architectures using Large Language Models (LLMs). This evolution has been driven by advances in natural language processing, retrieval-augmented generation (RAG), and multi-agent systems that break down fact-checking into specialized, coordinated tasks.

This review examines automated fact-checking systems, emphasizing multi-agent architectures for video content verification. It analyzes the technological foundations of each pipeline component, from claim detection to evidence retrieval to verdict synthesis, and identifies gaps limiting practical deployment. The review covers literature from 2017 to 2025, focusing on systems combining LLMs with information retrieval and structured reasoning.

2 Multi-Agent Architectures for Fact-Checking

2.1 Foundational Multi-Agent Frameworks

FactAgent [18] introduced an agentic workflow that explicitly emulates the methodology of human fact-checkers. Rather than fine-tuning models for specific tasks, FactAgent employs a pre-trained LLM that operates through a structured script: (1) gathering evidence via tools or internal knowledge, (2) analysing that evidence, and (3) synthesising a verdict. A key advantage of this approach is its zero-shot nature: the system uses the LLM's existing capabilities without requiring task-specific training data. However, the sequential multi-step process can introduce latency, and errors in early stages may cascade through the pipeline.

LoCal [3] handles complex claims through a decomposition–reasoning–evaluation loop. It uses specialized agents: a decomposer breaking claims into sub-claims, reasoning agents verifying each with evidence, and evaluators ensuring logical consistency and testing counterfactual scenarios. If inconsistencies arise, agents iteratively revise their reasoning. This targets single-pass verification weaknesses but increases computational cost.

Multi-agent debate systems [11, 15] use adversarial collaboration where distinct LLM agents take opposing roles: one argues for a claim's truth, another against it, while a judge decides the outcome. This exposes contradictions and forces agents to defend positions with evidence, reducing confirmation bias and hallucinations. The adversarial setup prevents premature convergence on incorrect conclusions, though debates can reach impasses and quality depends on the judge agent's synthesis ability.

2.2 Evolution and Current Landscape

Multi-agent design for fact-checking emerged recently (2023–2025). Early research used linear pipelines where a single model performed one task at a time. By 2024, works like FactAgent and LoCal began breaking verification into specialized agents [3, 18]. In 2025, researchers added debate mechanisms, self-reflection, and richer tool use [10, 15, 27]. MAD-Sherlock showed that debate-driven systems reduce hallucinations through collaborative verification [11]. Despite progress, recent evaluations reveal latency challenges, high compute requirements, and the need for careful orchestration to avoid loops or premature termination.

3 Automatic Claim Detection in Text

Identifying which statements need fact-checking is crucial for any verification pipeline. For YouTube videos, this means scanning transcripts to flag factual assertions that are verifiable and important enough to check. This task is known as claim check-worthiness detection.

3.1 Traditional Supervised Approaches

Claim detection research began in the mid-2010s, targeting political speech. **ClaimBuster** was pioneering work, the "first-ever end-to-end fact-checking system" [9]. It used a supervised model trained on human-labelled debate transcripts to score sentences for verifiable factual claims. Using feature engineering and early neural networks, it achieved sufficient accuracy for integration by fact-checking organizations like Duke Reporters' Lab [9]. ClaimBuster automated the initial triage step, helping journalists prioritize statements from debates or speeches.

Subsequent work refined datasets and models. **ClaimRank** expanded U.S. presidential debate data and introduced context-aware modeling, using surrounding sentences to improve detection [7]. The **CLEF CheckThat!** lab (2018–2022) released annual challenges with social media posts or political statements in multiple languages, labelled for check-worthiness [26]. These efforts established claim detection as a classification or ranking problem, with BERT and transformers becoming dominant after 2018. The key insight was that check-worthiness requires assessing both **importance** and **verifiability**. “The sky is blue” is factual but not check-worthy due to obviousness. “Unemployment rose by 15% last quarter” is both factual and check-worthy because it’s verifiable and important. Training models to capture this required carefully annotated datasets with clear guidelines.

3.2 LLM-Based Claim Detection

Recent work explores whether large language models can identify check-worthy claims without fine-tuning. **Sawinski et al. (2023) and Hyben et al. (2023)** compared fine-tuned BERT variants with GPT-3/GPT-4 in zero-shot or few-shot mode [23]. Simple zero-shot LLM prompts still underperform fine-tuned models on benchmarks. LLMs often have inconsistent internal definitions of “worthiness” and are sensitive to prompt wording, while fine-tuned models have learned explicit criteria from labelled data.

However, careful prompts can substantially improve LLM performance. **Li et al. (2023)** built a fully automated fact-checking prototype with an LLM-based claim detection module, using GPT-3 with verbose few-shot prompts [13]. While quantitative metrics weren’t reported, the work demonstrated using LLMs as drop-in replacements for claim detectors.

Ni et al. (2024) proposed a three-step prompting approach for consistent claim identification [19]. The LLM analyzes text in stages: highlighting factual statements, applying check-worthiness criteria, and ranking by importance, similar to chain-of-thought reasoning. This improved consistency but focused on verifiable claim identification rather than worthiness ranking [20].

3.3 Current Challenges and Hybrid Approaches

Key challenges include: (1) **definition ambiguity**: what constitutes a “check-worthy” claim varies by context; (2) **scalability**: scanning long transcripts with LLMs is slow and expensive; (3) **false positives**: overly aggressive detection wastes resources; and (4) **domain adaptation**: models trained on political debates may not work for scientific or economic content.

One approach combines lightweight classifiers (fine-tuned transformers) for initial filtering with LLMs performing a second pass for borderline cases. This two-stage detection aligns with industry practice where automated systems highlight candidates for human fact-checkers [9]. However, with recent advances in LLM capabilities and structured outputs, pure LLM-based approaches have become increasingly viable for end-to-end claim detection.

4 Retrieval-Augmented Generation and Information Retrieval

A core pillar for automated fact-checking is retrieval-augmented generation (RAG), which combines text generation with external information retrieval to ground outputs in verifiable sources rather than relying on parametric memory.

4.1 RAG Fundamentals

Lewis et al. (2020) formalized RAG by showing that augmenting generation with external knowledge retrieval significantly improves performance on knowledge-intensive tasks [12]. For fact-checking, RAG is essential: systems must fetch reliable sources that support or refute claims. The **FEVER** dataset exemplified this retrieve-then-verify pattern: given a claim, retrieve relevant documents (e.g., Wikipedia pages), then determine if they support or refute the claim [26]. Modern systems use LLMs for both retrieval and verification, conditioning their reasoning on retrieved evidence.

RAG addresses a critical limitation of pure LLM approaches: parametric knowledge can be outdated, incomplete, or hallucinated. By retrieving external information (from document indexes, web search, or APIs), RAG systems access current information, provide source attribution, and ground reasoning in verifiable evidence. This is crucial for fact-checking, where claims often reference recent events, specific statistics, or specialized knowledge not in LLM training data.

4.2 Tool Use and Web Retrieval

Integrating tool use with LLMs has enabled more sophisticated retrieval. The **ReAct pattern** (Reason and Act) interleaves tool use with chain-of-thought reasoning, letting LLMs decide when to call external tools like search engines [29]. This advances over fixed retrieve-then-verify pipelines by adaptively determining what information is needed and how to query for it.

WebGPT demonstrated training LLMs to use web browsers to answer questions and cite sources, greatly improving factual accuracy over vanilla GPT-3 [17]. The key innovation was teaching the model to navigate search results, click through to sources, and synthesize information from multiple pages with proper attribution.

Toolformer showed that LLMs can be fine-tuned to call external tools like search engines or calculators for factual information, reducing hallucinations by grounding answers in retrieved evidence [24]. The model learns when its parametric knowledge is insufficient and explicitly invokes tools to fill gaps.

Chern et al. (2023) proposed using Google Search, Google Scholar, code interpreters, and other tools to fact-check LLM-generated text, verifying outputs against external sources [4]. **Cheung and Lam (2023)** combined search-engine retrieval with LLaMA to predict claim veracity, using retrieved web information rather than relying solely on training data [5]. These tool-augmented methods address LLMs' inherent knowledge limitations, which can be outdated or incomplete [4].

4.3 Open Web versus Closed Knowledge Bases

Most academic fact-checking systems restrict retrieval to trusted corpora (primarily Wikipedia) to simplify evaluation and ensure evidence quality. This yields high precision in closed-domain settings but severely limits real-world coverage [6]. Economic claims might require World Bank reports, medical claims need CDC guidelines, and breaking news requires recent articles, none available in static Wikipedia dumps.

The FEVER dataset, while influential, exemplifies this limitation by assuming all verifiable claims can be checked against a June 2017 Wikipedia snapshot. This breaks down for recent events, specialized domains, or claims requiring synthesis across multiple specialized sources.

Tian et al. (2024) integrated web-retrieval agents into an LLM pipeline and demonstrated improved misinformation detection compared to standalone LLMs [27]. However, open-web retrieval introduces challenges: (1) **source credibility**: not all websites are reliable; (2) **information quality**: web content varies in accuracy; (3) **ranking complexity**: identifying relevant sources among millions of candidates; and (4) **dynamic nature**: content changes, affecting reproducibility.

Current best practices include prioritizing sources with high domain authority (established news organizations, academic institutions, government agencies), cross-referencing multiple independent sources, explicitly evaluating source credibility using metadata (publication date, author credentials, institutional affiliation), and maintaining transparency by exposing retrieved sources to users. Systems like FactAgent incorporate evidence retrieval as a dedicated step, using search tools to query the web and filtering results based on relevance and credibility [18].

4.4 Query Optimization for Fact-Checking

A critical but often overlooked component of RAG systems is query formulation. The same claim can be verified or refuted depending on how search queries are constructed. Query quality significantly impacts downstream task performance [9, 15].

Effective query optimization involves several strategies. **Keyword extraction** identifies salient terms likely to appear in relevant sources, filtering stop words and focusing on entities and key concepts. **Query expansion** generates multiple variants to capture different phrasings; for example, expanding "unemployment rate increased" to also search for "jobless claims rose" or "labour market deterioration" [12]. **Entity**

recognition identifies named entities (people, organizations, locations) that should be included in queries, as these serve as strong signals for retrieval. **Temporal awareness** incorporates time constraints when claims reference specific periods, adding the relevant year when verifying recent events.

Recent multi-agent systems often dedicate a specialized agent to query generation, recognizing this step significantly impacts retrieved evidence quality [18]. Poor queries may miss relevant sources or retrieve irrelevant information, degrading overall performance regardless of verification model quality. FactAgent includes explicit query formulation as one of its agent steps, using the LLM to generate search-optimized queries [18].

5 LLM-Based Claim Verification Methods

Once claims are identified and evidence retrieved, systems must determine veracity, labelling claims as supported (true), refuted (false), or not enough evidence. Traditional approaches treated this as textual entailment, using neural classifiers to determine if evidence entails or contradicts claims. With LLMs, a new approach emerged: using models to perform verification through natural language reasoning.

5.1 Prompting Strategies

Zero-shot and few-shot prompting involves providing an LLM with a claim and evidence, asking it to decide veracity and explain why: "Claim: X. Evidence: [text]. Based on the evidence, is the claim true or false?" [30]. In zero-shot mode, the LLM relies on internal reasoning and evidence interpretation. In few-shot mode, the prompt includes examples of claims with evidence and the correct verdict to guide the model.

GPT-4 and similar models show surprising capability at this task, often correctly interpreting whether evidence supports statements. However, LLMs can be overly agreeable, sometimes hallucinating justifications or defaulting to "Supported" even when evidence is insufficient [30]. This confirmation bias stems from models' training to be helpful and provide answers, even when saying "I don't know" would be more appropriate.

Careful prompt engineering can help. Effective strategies include explicitly instructing the model to answer "Not Enough Evidence" when information is insufficient, adding system messages emphasizing accuracy over helpfulness, requesting citation of specific evidence sentences supporting its verdict, using temperature settings near zero to reduce randomness, and implementing multi-pass verification where the model first generates a verdict then critiques its own reasoning.

Despite these techniques, pure prompting still lags behind specialized models on complex datasets, particularly for subtle cases requiring deep domain knowledge or multi-step reasoning.

5.2 Chain-of-Thought Reasoning and Agent Loops

More advanced approaches use **chain-of-thought reasoning** or implement the LLM as an agent in a loop. The **ReAct pattern** has the LLM explicitly reason step-by-step while using tools [29]. For verification, this might involve: (1) breaking the claim into parts, (2) querying a search engine for each part, (3) evaluating each piece of evidence, and (4) synthesizing a conclusion.

FactAgent's structured workflow exemplifies this: the LLM follows a script where each step (search, read results, extract evidence, cross-check, formulate verdict) is explicit and logged for transparency [18]. This approach is more flexible than fixed pipelines; if initial evidence is inconclusive, the agent can trigger refined searches. A major benefit is zero-shot operation without training, mimicking human fact-checker processes [18].

Chain-of-thought provides transparency (each reasoning step is explicit and inspectable), debuggability (identifying which step failed), adaptability (adjusting strategy based on intermediate results), and explainability (the reasoning trace serves as a natural language explanation). However, multiple LLM calls for each sub-step can be slow and expensive, and errors compound across stages. If claim decomposition is incorrect, subsequent reasoning will be compromised regardless of retrieval and evaluation quality.

5.3 Self-Consistency and Verification

SelfCheckGPT introduced self-consistency checking for hallucination detection [16]. The method generates multiple independent answers to the same query and checks if factual assertions agree across responses. If answers diverge on details, those details likely represent hallucinations [16, 28]. While SelfCheckGPT doesn't use external evidence, the principle extends to verification: an LLM can double-check its own claims.

The self-consistency approach operates on the principle that hallucinated information will vary across samples (since it's essentially random), while information grounded in training data will be consistent. By generating multiple explanations and checking for factual consistency, systems can identify low-confidence or potentially hallucinated components.

LLM-as-a-judge approaches use one model to generate answers and another (or the same model in a different mode) to verify them. For fact-checking, this could mean using GPT-4 to generate a verdict, then prompting it to validate: "Given the claim, evidence, and explanation, is the explanation correct and does it truly support the claim?" This metacognitive step can catch errors before presenting results [21, 22].

Cross-model checking uses different models to verify outputs. For example, GPT-4 might generate a verdict, then a smaller model fine-tuned on FEVER validates it against evidence. If they disagree, the system might abstain or ask GPT-4 to reconsider [1]. This ensemble approach can improve reliability by detecting flawed reasoning. Another relevant task is **stance detection**: classifying evidence snippets as supporting, refuting, or not mentioning the claim. Many verification pipelines have dedicated stance models, which can also be implemented via prompting [26].

5.4 Current Capabilities and Limitations

Carefully prompted LLMs (especially GPT-4-class models) can achieve near state-of-the-art performance on tasks like FEVER [26]. However, they still make mistakes, especially on ambiguous or complex claims requiring specialized knowledge or multi-step reasoning.

A noted limitation is the tendency to default to "Supported": models sometimes erroneously agree claims are true if any related evidence is found (confirmation bias), rather than truly verifying the exact claim [30]. For example, given the claim "Paris has a population of 10 million" and evidence stating "Paris metropolitan area has 12 million residents," an LLM might incorrectly mark this as supported due to numerical proximity without recognizing the distinction between city proper and metropolitan area.

Designing prompts or agent behaviors to be appropriately skeptical and output "Not Enough Info" when evidence is lacking remains important. This requires careful calibration: the system should neither be too trusting (accepting weak evidence) nor too skeptical (rejecting valid evidence due to minor inconsistencies).

6 Evaluation of Fact-Checking Systems

Evaluating automated fact-checking systems requires assessing accuracy, explanation quality, evidence usage, and practical usability. This section examines evaluation methodologies from recent literature.

6.1 Veracity Classification Metrics

When framed as classification (true/false or support/refute), standard metrics include accuracy, F1-score, and precision/recall. The FEVER challenge introduced the **FEVER score**, defined as the accuracy of the claim label and provision of at least one correct supporting evidence [26]. This metric penalizes systems that get labels right without proper reasoning or evidence grounding. For multi-class truth scales (e.g. “true”, “mostly true”, “half-true”, “mostly false”, “false”, “pants on fire” as used by PolitiFact), accuracy within each class or Cohen’s kappa for ordinal scales can be used, with disagreement severity varying by class distance.

6.2 Evidence Retrieval Metrics

A critical aspect is whether systems find appropriate evidence. **Recall@k** (e.g. Recall@5, Recall@10) measures whether correct evidence appears in the top k retrieved documents or sentences [26]. **Precision** measures what proportion of selected evidence is relevant, indicating how much noise accompanies signal. **Mean Average Precision (MAP)** accounts for both relevance and ranking order, rewarding systems that place relevant documents higher. End-to-end evaluations often credit systems only when they retrieve human-identified supporting or refuting evidence, though this can be restrictive as multiple valid sources may exist for a claim.

6.3 Explanation Faithfulness and Quality

When systems produce textual explanations, evaluation becomes challenging. Ideally, explanations should be **faithful** (reflect actual reasoning without introducing external information) and **factually consistent** with evidence. Automatic metrics like BLEU or ROUGE compare to reference explanations but don’t measure factuality well [21]. More sophisticated approaches include FactCC [25], Q^2 (question-answering-based verification), and entailment checks that determine whether evidence and claim entail the explanation.

LLM-as-a-judge has become popular: using GPT-4 or similar models to score explanation coherence and factuality. Prompting GPT-4 with criteria such as factual accuracy, logical coherence, appropriate use of evidence, and absence of unsupported claims yields scores that correlate reasonably with human judgments when properly calibrated [21, 22]. Nevertheless, judge models may have biases, can be persuaded by confident but incorrect explanations, and may struggle with specialized domains, so validation against human assessments remains essential.

6.4 Logical Coherence and Consistency

Stance consistency checks whether evidence stances align with final verdicts. If a system claims a statement is true but all evidence is marked “Refutes”, that indicates retrieval problems or reasoning errors. LoCal explicitly evaluates logical consistency by checking if composed solutions imply claim veracity [3]. Secondary models can perform entailment checks to verify each claim against cited sources, creating a verification layer where the system's reasoning is itself verified.

6.5 Human Evaluation

Human judgment remains the gold standard for real-world systems. Researchers conduct user studies or expert assessments on output samples. Fact-checking experts may rate verdict correctness and reasoning soundness, while lay users can evaluate whether explanations are convincing and understandable. Human evaluation can also assess readability, perceived trust, actionability, and completeness: whether the system addressed all relevant aspects of the claim [15, 28].

6.6 Computational Performance Metrics

For practical deployment, computational efficiency matters. Relevant metrics include latency (time from claim input to verdict output), throughput (claims processed per unit time), monetary cost (API or infrastructure expenditure), and resource utilization (memory, CPU, GPU). Although rarely reported in academic papers, these metrics are critical for operational systems, particularly those targeting near-real-time analysis of streaming video content [18, 27].

7 Current Limitations and Research Opportunities

Despite rapid progress, automated fact-checking systems have significant limitations constraining practical deployment. This section examines these gaps and identifies research opportunities.

7.1 Lack of End-to-End Usability

Most research prototypes focus on isolated components rather than seamless end-to-end tools. Some excel at claim detection but assume manual verification [9], while others verify claims but require human identification. Even ClaimBuster, dubbed "end-to-end", only highlighted claims without verification [9]. Complete systems need to integrate detection, verification, and source tracing, but existing systems typically address only one or two steps [14]. For YouTube videos, true end-to-end systems should handle transcription extraction, claim detection, evidence retrieval, verification, and user-friendly presentation—integration rarely achieved [14].

7.2 Dependence on Structured or Closed Sources

Much research restricts evidence to structured knowledge bases, primarily Wikipedia. While this yields cleaner evaluation, it severely limits applicability [1]. Real misinformation often involves domains where Wikipedia lacks coverage—economic claims need World Bank reports, medical claims need CDC guidelines, breaking news requires recent articles. Systems benchmarked on FEVER tend to be over-fitted to Wikipedia [26]. In practice, fact-checkers must handle a heterogeneous open web including news sites, scientific papers, and government databases, introducing challenges of source credibility and information quality. Many systems also don't handle multilingual content well [1].

7.3 Limited Accessibility for Non-Expert Users

Most solutions are research demonstrations, not polished products for public use. Code is often research-grade (notebooks, command-line scripts) that average users can't operate [14]. Google Fact Check Explorer is public but only searches existing fact-checks rather than performing new verification [8]. User experience is often lacking: systems output labels and confidence scores that non-experts find unactionable. Many advanced systems require heavy compute not feasible without powerful hardware or costly API access [14].

7.4 Scalability and Real-Time Constraints

Checking long videos with many claims stresses any system. If verification takes minutes per claim, videos with 20 claims become impractical for interactive use. Current research often doesn't address runtime performance [14]. Multi-agent systems can theoretically operate in parallel, but sequential dependencies limit parallelization. Cost is also a factor: using commercial LLMs like GPT-4 for every step can be prohibitively expensive at scale.

7.5 Trust and Transparency Issues

Users may be reluctant to trust AI verdicts without understanding how they were derived. Many systems have been criticized as "black boxes" [28]. Multi-agent systems and chain-of-thought approaches attempt to address this via explicit reasoning traces, but if explanations are generated by the same model making judgments, there's risk of post-hoc rationalization or hallucinated justifications. Maintaining clear separation between evidence and reasoning is critical yet challenging. LLMs have a documented tendency to confidently present false information, which in fact-checking contexts could undermine the system's purpose.

7.6 Lack of Comparative LLM Evaluation

Despite many LLM options (commercial models like GPT-4 or Claude; open-source models like LLaMA, Qwen, DeepSeek), there's limited systematic comparison of their suitability for fact-checking tasks. Research tends to use whichever model is most accessible without careful comparison of trade-offs in cost, accuracy, and latency [21].

8 Positioning of This Work

This thesis addresses the critical gaps identified above by developing a complete multi-agent system for YouTube video fact-checking, making the following contributions.

8.1 End-to-End Video Fact-Checking System

The system implements a complete pipeline integrating all stages from transcription extraction to claim detection, query generation, evidence retrieval, and verdict synthesis, processing raw YouTube URLs autonomously. Unlike research prototypes assuming pre-processed inputs, this handles the full workflow from video to verified claims, addressing the end-to-end usability gap [14].

8.2 Open-Web Evidence Retrieval

Moving beyond Wikipedia and closed knowledge bases, the system retrieves evidence from the live web using search engines, incorporates source credibility evaluation, and checks results across multiple sources. This addresses the limitation of systems constrained to structured sources and enables verification of claims about recent events, specialized domains, and topics not well covered in encyclopedic sources [1, 26].

8.3 Practical User Interface

A web-based interface with real-time streaming of intermediate results using Server-Sent Events makes the verification process transparent and accessible to non-expert users. This bridges the gap between research prototypes and usable products [14], demonstrating that academic advances can be packaged for practical use.

8.4 Systematic LLM Comparison

The thesis conducts comparative evaluation of different LLM configurations (commercial versus open-source models, different model sizes), analyzing trade-offs in cost, latency, and quality across pipeline components. This fills a gap in the literature where such trade-offs are rarely studied explicitly, providing practical guidance for deployment decisions [21].

8.5 Modular Multi-Agent Architecture

The system implements five specialized agents with structured data schemas (using Pydantic) enabling transparency, maintainability, and future updates. Each agent can be evaluated and optimized independently, and the modular design facilitates experimentation with different models and techniques, showing the practical benefits of multi-agent approaches in a production-oriented context [3, 15, 18].

8.6 Comprehensive Evaluation Framework

The evaluation combines quantitative metrics (technical performance, LLM-specific metrics like faithfulness and consistency), LLM-as-a-judge evaluation, and qualitative case studies across different video topics. This multifaceted approach addresses the evaluation challenges discussed earlier and provides a realistic assessment of system capabilities and limitations [21, 22].

By combining state-of-the-art techniques from multi-agent systems, RAG, and LLM-based verification with explicit focus on practical deployment, this work advances automated fact-checking from research prototype toward usable tool. The system is designed not to replace human fact-checkers but to empower both professionals and lay users by automating information gathering and initial verification, allowing human judgment to focus on complex cases requiring expertise, contextual understanding, and ethical consideration.

References

[1] Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: Fact extraction and verification over unstructured and structured information. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[2] Yochai Benkler, Robert Faris, and Hal Roberts. *Network Propaganda: Manipulation, Disinformation, and Radicalization in American Politics*. Oxford University Press, New York, 2018.

[3] Wei Chen, Ling Hu, Ming Zhang, and Rui Zhao. LoCal: Logical and causal fact-checking with llm-based multi-agents, 2024. OpenReview preprint.

[4] Alice Chern, Luis Prieto, and Riya Gupta. A tool-enabled framework for fact-checking language model outputs, 2023. Preprint manuscript.

[5] Wendy Cheung and Victor Lam. Augmenting llm fact-checking with web retrieval, 2023. Technical report.

[6] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16477–16508, 2023.

[7] Pepa Gencheva, Preslav Nakov, Georgi Karadzhov, Alberto Barrón-Cedeño, and Lluís Màrquez. ClaimRank: Detecting check-worthy claims in political debates. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 543–552, 2017.

[8] Google for Developers. Google fact check tools api. https://developers.google.com/fact-check/tools/api, 2024. Accessed: 2025-10-12.

[9] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. ClaimBuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948, 2021.

[10] ICWSM Workshop Committee. Proceedings of the icwsm 2025 workshop on agentic fact-checking, 2025. Workshop proceedings.

[11] Sonu Lakara, Karan Iyer, and Priya Subramanian. MAD-Sherlock: Multi-agent debate for fact verification, 2025. Preprint.

[12] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[13] Jiawei Li, Han Wu, and Ming Zhou. Automated fact-checking with llm-based claim detection, 2023. Preprint.

[14] Zheng Lin, Maya Patel, and Alicia Roberts. Fact-Audit: Requirements for trustworthy automated fact-checking systems, 2025. White paper.

[15] Liang Ma, Shiyu Hu, Wei Zhang, Hang Sun, and Yan Chen. Guided and knowledgeable multi-agent debate for fact verification. *Expert Systems with Applications*, 238:121857, 2025.

[16] Potsawee Manakul, Adian Liusie, and Mark JF Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, 2023.

[17] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Igor Babuschkin, Aakanksha Chowdhery, Sharad Amanpour, Pasha Wu, Jeffrey Jiang, Angela Jia, Shantanu Chen, et al. Webgpt: Browser-assisted question answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2022.

[18] Hieu Nguyen, Minh Pham, and Quang Tran. FactAgent: Towards robust fact-checking with multi-agent systems and advanced evidence retrieval. *arXiv preprint arXiv:2506.17878*, 2025. Preprint.

[19] Angela Ni and Samuel Carter. Structured prompting for consistent claim identification, 2024. Preprint.

[20] Angela Ni and Samuel Carter. Verifiable claim identification with large language models, 2024. Technical report.

[21] Sebastian Raschka. Llm evaluation: Four practical approaches. https://sebastianraschka.com/blog/2025/llm-evaluation-4-approaches.html, 2025. Accessed: 2025-10-12.

[22] Sebastian Ruder. The evolving landscape of llm evaluation. https://www.ruder.io/the-evolving-landscape-of-llm-evaluation/, 2025. Accessed: 2025-10-12.

[23] Marcin Sawiński, Michal Hyben, and Tomasz Wesołowski. Assessing large language models for claim detection tasks. In *Proceedings of the 7th Workshop on Fact Extraction and VERification*, pages 210–221, 2024.

[24] Timo Schick, Daniel Dwivedi-Yu, Roberta Raileanu, Nicolas Kramer, Sebastian Ruder, et al. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

[25] Skywork AI. How to avoid hallucinations: Editorial fact-check workflow for ai writing. https://skywork.ai/blog/how-to-avoid-hallucinations-ai-writing-fact-check-guide/, 2024. Accessed: 2025-10-12.

[26] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: A large-scale dataset for fact extraction and verification. In *Proceedings*

of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 809–819, 2018.

[27] Rui Tian, Ming Xie, and Hao Wang. Web-retrieval agents for misinformation detection, 2024. Preprint.

[28] Claire Wardle and Hossein Derakhshan. Information disorder: Toward an interdisciplinary framework for research and policy making. Technical report, Council of Europe, Strasbourg, France, 2017.

[29] Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Tom Griffiths, Graham Neubig, and Yongchao Cao. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 1–23, 2023.

[30] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.