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 3

2 Multi-Agent Architectures for Fact-Checking 3

2.1 Foundational Multi-Agent Frameworks 3

2.2 Evolution and Current Landscape 4

3 Automatic Claim Detection in Text 4

3.1 Traditional Supervised Approaches 4

3.2 LLM-Based Claim Detection 5

3.3 Current Challenges and Hybrid Approaches 6

4 Retrieval-Augmented Generation and Information Retrieval 6

4.1 RAG Fundamentals 6

4.2 Tool Use and Web Retrieval 7

4.3 Open Web versus Closed Knowledge Bases 8

4.4 Query Optimization for Fact-Checking 8

5 LLM-Based Claim Verification Methods 9

5.1 Prompting Strategies 9

5.2 Chain-of-Thought Reasoning and Agent Loops 10

5.3 Self-Consistency and Verification 11

5.4 Current Capabilities and Limitations 11

6 Evaluation of Fact-Checking Systems 12

6.1 Veracity Classification Metrics 12

6.2 Evidence Retrieval Metrics 12

6.3 Explanation Faithfulness and Quality 13

6.4 Logical Coherence and Consistency 13

6.5 Human Evaluation 13

6.6 Computational Performance Metrics 13

7 Current Limitations and Research Opportunities 14

7.1 Lack of End-to-End Usability 14

7.2 Dependence on Structured or Closed Sources 14

7.3 Limited Accessibility for Non-Expert Users 15

7.4 Scalability and Real-Time Constraints 15

7.5 Trust and Transparency Issues 16

7.6 Lack of Comparative LLM Evaluation 16

8 Positioning of This Work **16**

8.1 End-to-End Video Fact-Checking System 17

8.2 Open-Web Evidence Retrieval 17

8.3 Practical User Interface 17

8.4 Systematic LLM Comparison 17

8.5 Modular Multi-Agent Architecture 17

8.6 Comprehensive Evaluation Framework 18

1 Introduction

The proliferation of misinformation on digital platforms has become one of the defining challenges of the modern information age. YouTube, with more than 2 billion monthly active users and over 500 hours of video uploaded every minute, has emerged as a primary source of information for millions of people worldwide. However, the absence of effective verification mechanisms allows false or misleading claims to spread at massive scale, with documented impacts on public health, democratic processes, and social cohesion [1, 2, 28].

Automated fact-checking has evolved significantly since the mid-2010s, transitioning from rule-based systems and supervised classifiers to sophisticated architectures leveraging Large Language Models (LLMs). This evolution has been driven by advances in natural language processing, the emergence of retrieval-augmented generation (RAG) paradigms, and, more recently, the development of multi-agent systems that decompose the complex fact-checking workflow into specialized, coordinated sub-tasks.

This state of the art review examines the current landscape of automated fact-checking systems, with particular emphasis on multi-agent architectures and their application to video content verification. We analyse the technological foundations that enable each component of the fact-checking pipeline—from claim detection to evidence retrieval to verdict synthesis—and identify critical gaps that limit the practical deployment of existing systems. The review covers literature primarily from 2017 to 2025, focusing on systems that combine 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. Each step is performed sequentially with explicit reasoning traces, enabling transparency and debuggability. A key advantage of this approach is its zero-shot nature—the system leverages 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] addresses complex claims requiring multi-hop reasoning through a decomposition–reasoning–evaluation loop. The system employs multiple specialised agents:

a decomposer that breaks complex claims into simpler sub-claims, reasoning agents that verify each sub-claim by retrieving evidence, and evaluator agents that ensure logical consistency and test counterfactual scenarios. If inconsistencies are detected, agents iteratively revise their reasoning. This approach explicitly targets the brittleness of single-pass verification, aiming to ensure verdicts are logically sound and robust against perturbations, albeit at the cost of additional computational expense.

Multi-agent debate systems [11, 15] introduce adversarial collaboration where distinct LLM agents are assigned opposing roles—one arguing for a claim's veracity, another against it—with a judge agent ultimately deciding the outcome. This debate mechanism surfaces contradictions and forces each agent to defend its position with evidence, thereby reducing confirmation bias and hallucinations. The adversarial setup can improve factual reliability by preventing premature convergence on incorrect conclusions, although debates can reach stalemates and overall quality depends heavily on the judge agent's ability to synthesise competing arguments.

2.2 Evolution and Current Landscape

The shift to multi-agent design for fact-checking is recent (2023–2025). Early fact-checking research typically followed linear pipelines where a single model performed one task at a time. By 2024, works such as FactAgent and LoCal began modularising verification steps into specialised agents [3, 18]. In 2025, researchers integrated debate mechanisms, self-reflection capabilities, and richer tool use to improve reliability [10, 15, 27]. MAD-Sherlock demonstrated that debate-driven agent systems reduce hallucinations through collaborative verification [11]. Despite clear progress, empirical evaluations still reveal latency challenges, high compute requirements, and the need for rigorous orchestration strategies to avoid loops or premature termination in complex verification scenarios.

3 Automatic Claim Detection in Text

Identifying which statements in content are "claims" worthy of fact-checking is a crucial first step in any verification pipeline. In the context of YouTube videos, this means scanning transcripts to flag factual assertions that are verifiable and sufficiently important to check. This task is known as claim check-worthiness detection.

3.1 Traditional Supervised Approaches

Academic interest in claim detection began in the mid-2010s, primarily targeting political speech. **ClaimBuster** was pioneering work, touted as the "first-ever end-to-end fact-checking system" [9]. ClaimBuster provided a supervised model trained on human-labelled debate transcripts to score each sentence for likelihood of containing a verifiable factual claim. The system used feature engineering and early neural networks (pre-transformer era) to detect check-worthy statements, achieving sufficient accuracy to be integrated by fact-checking organisations such as Duke Reporters' Lab [9]. ClaimBuster effectively automated the initial triage step, helping journalists prioritise which statements from live debates or speeches merited human verification.

Subsequent research created refined datasets and models. The **ClaimRank** dataset expanded on U.S. presidential debate data and introduced context-aware modelling, considering surrounding sentences to improve detection [7]. The **CLEF CheckThat!** lab (2018–2022) annually released challenges with datasets of social media posts or political statements in multiple languages, labelled for check-worthiness [26]. These efforts established claim detection as a text classification or ranking problem, with BERT and other transformer models becoming dominant after 2018. The key insight from this era was that check-worthiness is not just about identifying factual statements, but also about assessing their **importance** and **verifiability**. A statement like "the sky is blue" is factual but not check-worthy due to its obviousness. Conversely, "unemployment rose by 15% last quarter" is both factual and check-worthy because it is verifiable and consequential. Training supervised models to capture this nuance required carefully annotated datasets with clear labelling guidelines.

3.2 LLM-Based Claim Detection

Recent work has explored 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 for claim detection and check-worthiness classification [23]. Their findings indicate that naïve zero-shot LLM prompts still underperform fine-tuned smaller models on established benchmarks. LLMs often have inconsistent internal definitions of "worthiness" and are sensitive to prompt wording, whereas fine-tuned models have learned explicit criteria from labelled data.

However, carefully crafted prompts can improve LLM performance substantially. **Li et al. (2023)** built a fully automated fact-checking prototype that included an LLM-based claim detection module, using GPT-3 with verbose few-shot prompts to identify claims in input text [13]. While quantitative metrics for this step were not reported, the work demonstrated feasibility of using LLMs as drop-in replacements for dedicated claim detectors.

Ni et al. (2024) proposed a three-step prompting approach to improve consistency in claim identification [19]. The method has the LLM analyse text in stages—first highlighting all factual statements, then applying check-worthiness criteria, and finally ranking by importance—essentially decomposing the decision similar to chain-of-thought reasoning. This approach improved consistency but focused on verifiable claim identification (whether a claim is objectively checkable) rather than worthiness ranking [20].

3.3 Current Challenges and Hybrid Approaches

Key challenges in claim detection include:

1. **Definition ambiguity** — what constitutes a "check-worthy" claim varies by context and application.
2. **Scalability** — scanning long transcripts with LLMs can be slow and expensive.
3. **False positives** — overly aggressive detection wastes verification resources.
4. **Domain adaptation** — models trained on political debates may not generalise to scientific or economic content.

A hybrid approach appears promising for practical systems: lightweight classifiers (fine-tuned transformers) can provide initial filtering for efficiency, with LLMs performing a second pass for nuanced judgement on borderline cases. This two-stage detection aligns with industry practice where automated systems highlight candidates for human fact-checkers to make final decisions [9]. Such architectures balance the speed and consistency of fine-tuned models with the flexibility and reasoning capabilities of LLMs.

4 Retrieval-Augmented Generation and Information Retrieval

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

4.1 RAG Fundamentals

Lewis et al. (2020) formalised RAG by showing that augmenting generation with a non-parametric memory (a document index) significantly improves performance on knowledge-intensive tasks [12]. In fact-checking, RAG is nearly essential: to verify a claim, systems must fetch reliable sources that either support or refute it. The original **FEVER** dataset exemplified this retrieve-then-verify pattern: given a claim, the system retrieved Wikipedia pages, then a classifier determined if the claim was supported or refuted by those pages [26]. Modern systems replace classifiers with LLMs but maintain the same fundamental architecture of conditioning verification on retrieved evidence.

The RAG paradigm addresses a critical limitation of pure LLM-based approaches: parametric knowledge can be outdated, incomplete, or hallucinated. By retrieving and conditioning on external documents, RAG systems can access current information, provide source attribution, and ground their reasoning in verifiable evidence. This is particularly crucial for fact-checking, where claims often reference recent events, specific statistics, or domain-specific knowledge that may not be well represented in an LLM's training data.

4.2 Tool Use and Web Retrieval

The integration of tool use with LLMs has enabled more sophisticated retrieval strategies. The **ReAct pattern** (Reason and Act) interleaves tool use with chain-of-thought reasoning, allowing LLMs to decide when to call external tools like search engines based on their current reasoning state [29]. This pattern represents a significant advance over fixed retrieve-then-verify pipelines, as the LLM can adaptively determine what information it needs and how to query for it.

WebGPT demonstrated the effectiveness of 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 not just to search, but to navigate search results, click through to promising sources, and synthesise information from multiple pages while maintaining proper attribution.

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

Chern et al. (2023) proposed a framework 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)** similarly combined search-engine retrieval with LLaMA to predict claim veracity, with the LLM using retrieved web information to make judgements rather than relying solely on training data [5]. These tool-augmented methods were motivated by limitations of LLMs' inherent knowledge, which can be outdated or incomplete for real-world claims [4].

4.3 Open Web versus Closed Knowledge Bases

Most academic fact-checking systems restrict retrieval to trusted corpora (Wikipedia being predominant) to simplify evaluation and ensure evidence quality. This approach yields high precision in closed-domain settings but severely limits real-world coverage [6]. For example, 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 assumption breaks down for claims about recent events, specialised domains not well-covered in Wikipedia, or claims requiring synthesis of information across multiple specialised 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 significant challenges:

- **Source credibility**: not all websites are equally reliable.
- **Information quality**: web content varies in accuracy and completeness.
- **Ranking complexity**: systems must identify the most relevant sources among millions of candidates.
- **Dynamic nature**: content changes over time, affecting reproducibility.

Current best practices for web retrieval in fact-checking include prioritising sources with high domain authority (established news organisations, academic institutions, government agencies), cross-referencing multiple independent sources to corroborate claims, 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 then filtering results based on relevance and credibility [18].

4.4 Query Optimization for Fact-Checking

An often overlooked but critical component of RAG systems is query formulation. The same claim can be verified or refuted depending on how search queries are constructed. Research on question generation for information retrieval has shown that query quality significantly impacts downstream task performance [9, 15].

Effective query optimisation for fact-checking involves several key strategies. **Keyword extraction** identifies the most salient terms in a claim that are likely to appear in relevant sources, filtering out stop words and focusing on entities and key concepts. **Query expansion** generates multiple query variants to capture different phrasings or perspectives on the claim—for example, expanding "unemployment rate increased" to also search for "jobless claims rose" or "labour market deterioration" [12]. **Entity recognition** identifies named entities (people, organisations, locations) that should be included in queries, as these often serve as strong signals for retrieval systems. **Temporal awareness** incorporates time constraints when claims reference specific periods, such as adding the relevant year when verifying recent events.

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

5 LLM-Based Claim Verification Methods

Once claims are identified and relevant evidence retrieved, systems must determine veracity—typically labelling claims as supported (true), refuted (false), or not enough evidence. Traditional approaches treated this as recognising textual entailment, using neural classifiers to determine if evidence entails or contradicts claims. With LLMs, a new paradigm has 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 retrieved evidence, asking it to decide veracity and explain why. For example: "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's style and criteria.

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 appears to stem from models' training to be helpful and provide answers, even when saying "I don't know" would be more appropriate.

Careful prompt engineering can mitigate these issues. Effective strategies include:

- Explicitly instructing the model to answer “Not Enough Evidence” when information is insufficient.
- Adding system messages emphasising the importance of accuracy over helpfulness.
- Requesting that the model cite specific evidence sentences supporting its verdict.
- Using temperature settings near zero to reduce randomness and increase consistency.
- Implementing multi-pass verification where the model first generates a verdict and then critiques its own reasoning.

Despite these techniques, pure prompting still lags behind specialised models on complex datasets, particularly for subtle cases requiring deep domain knowledge or multi-hop 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) synthesising 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 orchestrated 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].

The chain-of-thought approach provides several advantages for verification:

- **Transparency**: each reasoning step is explicit and inspectable.
- **Debuggability**: when errors occur, it is possible to identify which step failed.
- **Adaptability**: the system can adjust its strategy based on intermediate results.
- **Explainability**: the reasoning trace serves as a natural language explanation for the verdict.

However, the downside is efficiency: multiple LLM calls for each sub-step can be slow and expensive, and errors compound across stages. If the 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 or false facts [16, 28]. While SelfCheckGPT does not use external evidence, the principle extends to verification: an LLM can be prompted to explicitly double-check its own claims.

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

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 (or another model) 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 to users [21, 22].

Cross-model checking uses different models or the same model with different knowledge to verify outputs. For example, a powerful 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 cases where one model's reasoning is flawed. 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

Evaluation results from various benchmarks suggest that 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 specialised 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 recognising the distinction between city proper and metropolitan area.

Designing prompts or agent behaviours to be appropriately sceptical and output “Not Enough Info” when evidence is lacking remains important. This requires careful calibration—the system should neither be too credulous (accepting weak evidence) nor too sceptical (rejecting valid evidence due to minor inconsistencies).

6 Evaluation of Fact-Checking Systems

Evaluating automated fact-checking systems is multifaceted, requiring assessment of 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**—accuracy of the claim label and provision of at least one correct supporting evidence [26]. This metric penalises 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 of evaluation 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 actually relevant, indicating how much noise accompanies signal. **Mean Average Precision (MAP)** provides a single metric that accounts for both the relevance of retrieved documents and their ranking order, rewarding systems that place relevant documents higher in result lists. End-to-end evaluations often credit systems only when they retrieve human-identified supporting or refuting evidence, though this can be restrictive because multiple valid sources may exist for a claim.

6.3 Explanation Faithfulness and Quality

When systems produce textual explanations or rationales, 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 do not 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 increasingly popular: using GPT-4 or similar models to score explanation coherence and factuality. For example, 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 judgements when properly calibrated [21, 22]. Nevertheless, judge models may have biases, can be persuaded by confident but incorrect explanations, and may struggle with specialised 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 either 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 subject to verification.

6.5 Human Evaluation

Human judgement remains the gold standard for systems meant for real-world use. 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 (number of claims processed per unit time), monetary cost (API or infrastructure expenditure), and resource utilisation (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, contemporary automated fact-checking systems have significant limitations that constrain practical deployment and real-world impact. This section examines these gaps critically and identifies specific research opportunities they present.

7.1 Lack of End-to-End Usability

Limitation: most research prototypes focus on isolated problem slices rather than providing seamless end-to-end tools. Some systems excel at claim detection but assume verification is done manually [9]. Others verify given claims but require humans to identify them first. This fragmentation means few truly autonomous fact-checkers exist that users can feed raw content (such as videos) and receive comprehensive verified analyses.

Even ClaimBuster, while dubbed "end-to-end", essentially provided claim highlighting and left verification to humans [9]. **Fact-Audit** points out that complete systems need to integrate detection, verification with reasoning, and source tracing, but existing systems typically address only one or two of these steps [14]. For YouTube videos specifically, true end-to-end systems should handle transcription extraction, claim detection from long transcripts, evidence retrieval from diverse sources, verification with reasoned explanations, and presentation of results in user-friendly formats—a level of integration rarely achieved in current research [14].

Research opportunity: develop complete pipelines that integrate all stages from raw video input to verified claims with user-friendly presentation. This requires not just technical integration but careful attention to error propagation across stages and system-level optimisation.

7.2 Dependence on Structured or Closed Sources

Limitation: a large portion of fact-checking research restricts evidence to structured, highly reliable knowledge bases—primarily Wikipedia. While this yields cleaner evaluation and reduces web noise, it severely limits applicability [1]. Real misinformation often involves domains or topics where Wikipedia lacks coverage. Economic claims might require World Bank reports, medical claims need CDC guidelines, and breaking news requires recent articles—none available in static snapshots.

Systems benchmarked on FEVER or similar datasets tend to be over-fitted to Wikipedia as the source of truth, using wiki-specific retrieval heuristics or assuming single correct pages exist for claims [26]. In practice, fact-checkers must handle a heterogeneous open web including news sites, blogs, scientific papers, government databases, and multimedia content. This introduces challenges of source credibility, data variety, and information quality variations. Additionally, many systems do not handle multilingual or non-English content well, limiting global accessibility [1].

Research opportunity: develop robust open-web retrieval methods with explicit source credibility evaluation, multi-source corroboration strategies, and techniques for synthesising information from conflicting sources. This includes handling diverse evidence formats and supporting multilingual fact-checking.

7.3 Limited Accessibility for Non-Expert Users

Limitation: most current solutions are research demonstrations or internal tools, not polished products for public use. Academic papers propose models and report accuracy metrics, but code is often research-grade (notebooks, command-line scripts) that average users cannot operate. The general public, who could benefit most from automated fact-checking, seldom interacts with these systems directly [14].

One exception is Google Fact Check Explorer, which is public but only searches existing fact-check articles rather than performing new verification [8]. User experience is often lacking—systems output labels and confidence scores that non-experts find unactionable or unconvincing. Computational accessibility is another concern: many advanced systems require heavy compute not feasible without powerful hardware or costly API access [14].

Research opportunity: bridge the gap between research prototypes and usable products through intuitive interfaces, real-time feedback mechanisms, and cost-efficient architectures. This includes exploring cost-quality trade-offs to make systems economically viable for public deployment.

7.4 Scalability and Real-Time Constraints

Limitation: checking long videos with many claims stresses any system. If verification takes minutes per claim, a video with 20 claims becomes impractical for interactive use. Current research often does not address runtime performance, evaluating models on single queries or small datasets without considering throughput or latency [14]. Multi-agent systems can theoretically operate in parallel on different claims, but sequential dependencies (evidence retrieval preceding verification) limit parallelisation opportunities. Cost is also a scalability factor: using commercial LLMs like GPT-4 for every step can be prohibitively expensive at scale.

Research opportunity: perform systematic analysis of computational efficiency and cost–quality trade-offs in multi-agent systems. Questions such as which steps benefit most from powerful models, whether claim difficulty can guide routing, and how to parallelise agent operations effectively remain largely unexplored.

7.5 Trust and Transparency Issues

Limitation: users and fact-checkers may be reluctant to trust AI verdicts without understanding how they were derived. Many deep learning systems have been criticised 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 judgements, there is risk of post-hoc rationalisation or hallucinated justifications.

Maintaining clear separation between evidence and reasoning—ensuring systems quote actual sources rather than fabricating them—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.

Research opportunity: develop verification mechanisms for system outputs themselves, such as secondary validation of cited sources, consistency checks between evidence and explanations, and methods to detect when LLMs hallucinate rather than reason from evidence. Communicating uncertainty and limitations transparently to users is equally important.

7.6 Lack of Comparative LLM Evaluation

Limitation: despite the proliferation of LLM options (commercial models such as GPT-4 or Claude; open-source models like LLaMA, Qwen, DeepSeek), there is limited systematic comparison of their suitability for different fact-checking tasks. Research tends to use whichever model is most accessible without rigorous comparison of trade-offs in cost, accuracy, and latency across the full pipeline [21].

Research opportunity: conduct comprehensive comparative evaluations of different LLMs across fact-checking pipeline components, measuring not just accuracy but also cost, latency, consistency, and propensity toward hallucination. This would inform practical decisions about which models to use for which tasks.

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. The work makes the following contributions to the field.

8.1 End-to-End Video Fact-Checking System

We implement a complete pipeline integrating all stages—from transcription extraction to claim detection, query generation, evidence retrieval, and verdict synthesis—that processes raw YouTube URLs autonomously. Unlike research prototypes that assume pre-processed inputs, this system 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 handles multi-source corroboration. This addresses the limitation of systems constrained to structured sources and enables verification of claims about recent events, specialised domains, and topics not well covered in encyclopaedic sources [1, 26].

8.3 Practical User Interface

We develop a web-based interface with real-time streaming of intermediate results using Server-Sent Events, making 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), analysing 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

We implement five specialised agents with structured data schemas (using Pydantic) that enable transparency, maintainability, and future extensibility. Each agent can be evaluated and optimised independently, and the modular design facilitates experimentation with different models and techniques. This demonstrates 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 such as 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 judgement 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.