Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Agent-Verified System for Resource-Constrained Query-Focused Summarisation using Small Language Models

Shanta Majumder
Daffodil International University
Suprove Chandra Sarkar
suprove.swe@diu.edu.bd

Daffodil International University

Research Article

Keywords: Agent, edge computing, Retrieval-Augmented Generation (RAG), Small Language Models (SLMs)

Posted Date: June 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9986413/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Agent-Verified System for Resource-Constrained Query-Focused Summarisation using Small Language Models

Shanta Majumder[1] and Suprove Chandra Sarkar[1*]

[1*]Department of Software Engineering, Daffodil International University, Daffodil Smart City (DSC), Birulia, Savar, Dhaka, 1216, Bangladesh.

*Corresponding author(s). E-mail(s): suprove.swe@diu.edu.bd;
Contributing authors: shanta35-1183@s.diu.edu.bd;

Abstract

Retrieval-Augmented Generation (RAG) powered by Large Language Models (LLMs) has become a prominent paradigm for knowledge-intensive natural language processing, substantially improving the factual accuracy of query-focused summarisation. However, deploying these architectures on resource-constrained edge devices remains highly challenging due to the severe latency and computational overhead of large-scale models. Small Language Models (SLMs) offer a computationally feasible alternative, yet they are considerably more prone to hallucinations and the "Relevant but Unfaithful" contradiction. To address this, an Agent-Verified, parallel RAG architecture specifically designed for edge computing is presented in this work. The architecture creates a dual-path verification loop by interrogating both an external Wikipedia agent and a local vector database simultaneously. An SLM Judge dynamically cross-references the retrieved evidence to resolve knowledge disputes mathematically before text generation. Furthermore, this work demonstrates that abstractive fact-checking systems are fundamentally penalised by conventional n-gram evaluation techniques, and presents empirical evidence to support this finding.

Keywords: Agent, edge computing, Retrieval-Augmented Generation (RAG), Small Language Models (SLMs)

1 Introduction

The rapid development of natural language processing has considerably contributed to query-focused document summarisation and question-answering systems. At the core of these advancements is a framework called RAG, which uses external knowledge to increase factual accuracy [1]. Static retrieval pipelines are increasingly being replaced by collaborative multi-agent frameworks and agentic RAG as the paradigm for managing more complex, multi-hop, long-context queries [2, 3]. By employing various agents for information filtering, planning, and document interpretation, these intelligent systems offer highly dynamic, context-aware responses [4–6]. Furthermore, data from multiple sources are frequently synthesised into logical, verifiable facts using complex abstractive summarisation algorithms [7, 8]. Integrating these advanced conversational and memory-driven capabilities into practical, real-world applications remains a significant architectural challenge [9].

Classical RAG architectures have significant disadvantages when deployed in resource-constrained environments, beyond the core benefits that multi-agent systems provide. LLMs demand considerable computational resources, making edge-device deployment unfeasible and introducing significant latency. Despite offering a computationally efficient replacement for tool calling and agentic workflows, SLMs are inherently more susceptible to the "Relevant but Unfaithful" dilemma [10]. In this case, even when the retriever accurately surfaces the appropriate context, the generative model may produce hallucinations or fail to reconcile contradictory information between internal documents and globally verified facts. Most current frameworks rely on sequential retrieval paths that neither guarantee factual faithfulness nor enable concurrent verification of external information without exceeding hardware constraints.

To address this gap, this work introduces a unique Agent-Verified, concurrent RAG architecture designed specifically for edge computing. The proposed methodology bypasses the traditional linear retrieval pipeline by implementing a simultaneous, dual-path verification loop. When a query is received, the system concurrently queries a highly optimised local vector database for domain-specific internal context and an external web agent (Wikipedia) for worldwide factual verification. Rather than relying on a single model to blindly summarise the first retrieved document, an SLM Judge actively cross-references the retrieved evidence, classifies the knowledge states, and computes distinct confidence scores. To resolve knowledge conflicts mathematically before the final content is generated, the system dynamically switches to the external web agent if the internal document lacks the information required to answer a query.

The primary contribution of this research is the validation of a highly faithful, query-focused, edge-deployable summarisation system. The system achieves strong fidelity scores and mitigates latency spikes by coordinating a multi-agent verification loop with highly optimised 3B and 4B parameter models. Additionally, this work reveals a significant flaw in widely used n-gram evaluation techniques such as ROUGE [11] and BLEU [12], demonstrating that they penalise abstractive, fact-checking architectures by design. Through extensive ablation testing against complex single-hop and multi-hop datasets including MS MARCO [13], TriviaQA [14], and HotpotQA [15], this study reveals that SLM-driven agentic conflict resolution can effectively identify

internal knowledge gaps and reduce RAG hallucinations without requiring massive computational infrastructure.

2 Related Work

Recent advances in Retrieval-Augmented Generation (RAG) have shifted from simple linear pipelines to adaptive, multi-agent systems capable of autonomous reasoning and sophisticated synthesis. However, these advances still rely on large language models whose computational and latency costs hinder edge deployment. To address this, growing research combines Small Language Models (SLMs) with graph-based retrieval for efficient, query-specific summarisation. Building on these trends, the proposed method introduces an SLM-powered, agent-validated adjudication mechanism for reliable, on-device fact checking and synthesis.

2.1 Retrieval-Augmented Generation and Summarisation

The integration of external knowledge into generative models, defined as RAG, forms the foundational architecture for reducing hallucinations in knowledge-intensive NLP tasks [1]. Early approaches focused primarily on linear retrieval and conventional text generation. However, as the demand for factual accuracy increased, researchers began employing advanced abstractive multi-document summarisation techniques, synthesising and validating facts from numerous sources using multimodal evidence and graph-based architectures [7, 8]. Iterative retrieval and planning algorithms designed specifically for multi-hop question answering have since been developed, as complex, multi-step queries remain a barrier for standard RAG pipelines despite these advances [4].

2.2 Multi-Agent Frameworks and Agentic RAG

To overcome the shortcomings of static retrieval, research has shifted swiftly towards the Agentic RAG framework [2]. Recent work demonstrates that long-context tasks and comprehensive document understanding can be significantly enhanced through interactive chains of agents, in which several language models manage autonomous reasoning, retrieval, and synthesis activities [3, 5]. Multi-agent filtering algorithms have been effectively applied to parse noisy data whilst maintaining contextual relevance [6]. These systems have further evolved into conversational agents, moving beyond simple RAG to handle long-term memory (LTM) and engage in dynamic dialogue [9].

2.3 Resource Constraints and Small Language Models

Multi-agent architectures can achieve state-of-the-art accuracy, but their fundamental reliance on massive LLMs introduces computational and latency constraints that prevent deployment on edge devices. SLMs have consequently received considerable research attention as a potential remedy. With targeted fine-tuning, SLMs can outperform larger models in efficient agentic tool calling [10]. However, the implementation of fully autonomous, multi-agent conflict-resolution loops under strict hardware constraints remains far from complete. Current lightweight systems often lack the

architectural capacity to perform dual-path retrieval and fact verification simultaneously without exceeding memory constraints. The proposed Agent-Verified solution directly addresses this gap by developing an SLM-driven adjudicator capable of performing complex query-focused summarisation entirely on the edge.

2.4 Faithfulness, Evaluation Metrics, and Benchmarks

Ensuring that generated summaries accurately reflect their source material has long been a priority in NLP. The distinction between what a model attends to and what it verbalises was codified by foundational work on faithfulness, which established interpretability criteria that remain crucial to system evaluation [16]. The n-gram overlap metrics ROUGE [11] and BLEU [12], widely used in automatic assessment, have enabled systematic cross-system comparisons in summarisation and generation. Regarding retrieval, the relevance-assessment principles that drive both sparse and dense retrieval today were defined by early information retrieval theory [17]. HotpotQA [15] and the multi-hop QA dataset of Ho et al. [18] have further established challenging benchmarks that require multi-step reasoning, providing essential evaluation frameworks for systems such as the one proposed here.

2.5 Long-Context Challenges and Prompt-Level Remedies

A related field of study examines how well language models make use of the context they are given, rather than assuming effective utilisation. Empirical research has shown that models sometimes neglect tokens distant from the start or end of lengthy inputs, a phenomenon commonly referred to as the "lost-in-the-middle problem" [19]. Architectural responses include positional-interpolation techniques that increase the effective context window without retraining [21], and training-time strategies that specifically motivate models to utilise all available context [20]. At inference time, prompt-compression techniques such as LongLLMLingua [22] reduce redundancy before tokens are passed to the model, lowering latency without compromising essential information. These long-context studies were made possible by the foundational few-shot capabilities demonstrated at scale by Brown et al. [23], whose GPT-3 work showed that general-purpose language models could handle retrieval-adjacent tasks with minimal task-specific supervision. Systematic benchmarking has since examined the successes and failures of RAG pipelines in terms of retrieval faithfulness, context integration, and answer correctness [24], providing an empirical foundation for the SLM-driven approach proposed here.

2.6 Query-Focused Summarisation and Graph-Structured Retrieval

A specialised area of work addresses query-focused summarisation, which extends beyond passage-level retrieval and requires that the output be shaped by a specific information need rather than offering a general document digest. Conditioning on the query at generation time significantly increases topical precision, as demonstrated by sequence-to-sequence models augmented with query-relevance signals and length constraints [25]. Subsequently, deep communicating agents were proposed to distribute

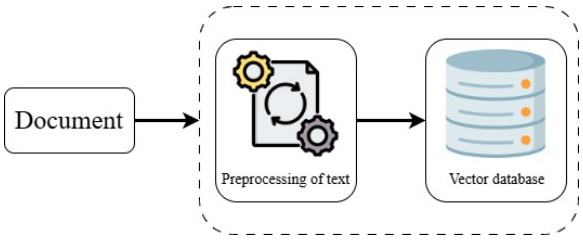
the summarisation load across specialist sub-modules, each responsible for a different component of source coverage [26], an architectural idea that closely foreshadows the multi-agent designs discussed above. Structured topic templates and convolutional decoders provide further evidence that enforcing explicit discourse scaffolding improves coherence in multi-document outputs [27]. At the retrieval layer, GraphRAG [28] extends the concept of flat document ranking by encoding inter-sentence and inter-document relationships as edges, enabling coverage-aware selection of complementary evidence and facilitating global, query-focused synthesis via community-level knowledge graphs [8]. In domain-specific contexts such as customer-service QA, grounding retrieval in explicit knowledge graphs has also shown promise where entity relationships provide precision that dense retrieval alone cannot match [29].

2.7 Infrastructure, Tooling, and the SLM Frontier

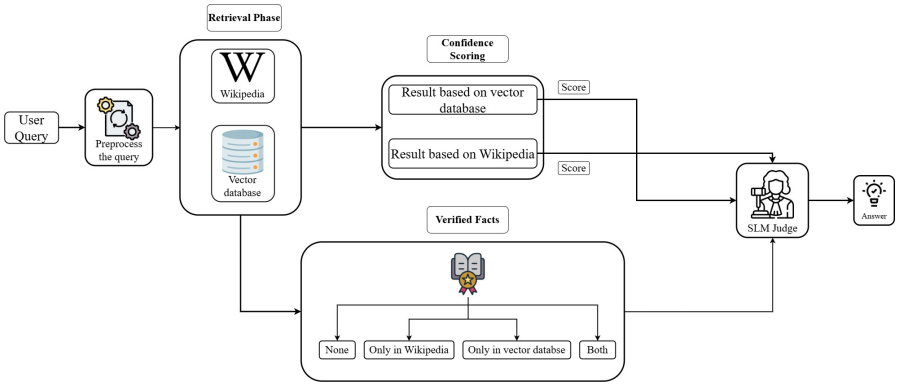
Effective vector search and orchestration infrastructure is crucial for the practical implementation of agentic RAG systems. The proposed system makes direct use of both the high-throughput approximate nearest-neighbour search provided by the FAISS library [30] and the graph-based execution model for coordinating heterogeneous agent workflows provided by LangGraph [31]. Early open-domain retrieval pipelines that read Wikipedia to provide answers demonstrated the viability of the retrieve-then-read paradigm at scale [32], defining the engineering blueprint that agentic systems currently follow. Recent work suggests that Small Language Models represent a genuinely superior architecture for agentic workloads where latency, privacy, and edge deployment are paramount [33]. Collaborative multi-agent ensembles of SLMs have already demonstrated the ability to solve challenging reasoning problems and synthesise new domain knowledge [34], supporting the direction taken by the proposed Agent-Verified adjudicator.

3 Methodology

The proposed framework introduces a novel Agent-Verified, parallel RAG architecture that improves factual faithfulness whilst operating under strict hardware limitations. To accomplish this, the system replaces conventional linear retrieval with a simultaneous, dual-path verification loop, enabling an SLM Judge to actively resolve knowledge conflicts before final text generation. The following diagrams illustrate the overall architecture:



(a) Document pre-processing to vector database



(b) Agentic framework with dual verification loop

Fig. 1: The Framework of an Agent-Verified System for Resource-Constrained Query-Focused Summarisation

The framework introduces an Agent-Verified, parallel RAG architecture that operates under limited hardware whilst maximising faithfulness. Figure 1a illustrates the document pre-processing pipeline, and Figure 1b illustrates the system with a dual-path verification loop combined with an SLM Judge that verifies facts, resolves conflicts, and generates a final answer.

3.1 Document Pre-processing

The framework is initiated when a document is ingested into the system. To ensure high-fidelity semantic retrieval and maximise computational efficiency during the embedding phase, the full NLP pre-processing pipeline includes tokenisation, lemmatisation, stop-word removal, named entity recognition (NER), part-of-speech (POS) tagging, dependency parsing, noun-phrase checking, and text cleaning using regular expressions.

3.2 Query Pre-processing

The framework is initiated when a user submits a query. During the pre-processing phase, the query is cleaned, tokenised, and converted into dense vector embeddings. This query vector is mathematically optimised for semantic similarity matching during the subsequent retrieval phase, thereby increasing computational efficiency.

3.3 Dual Retrieval Phase

To overcome the knowledge gap inherent in standard RAG systems, the framework implements a dual-path retrieval mechanism:

1. **Internal Retrieval (vector database):** The framework searches the vector database to find precise, domain-specific, internal context.
2. **External Retrieval (Wikipedia):** Concurrently, an external web agent queries Wikipedia to retrieve broader, verifiable facts.

This phase operates in parallel rather than linearly, thereby mitigating latency bottlenecks in resource-constrained environments.

3.4 Knowledge State Classification and Confidence Scoring

After retrieving from both sources, the system evaluates the information before generating the final answer:

1. **Verified facts categorisation:** The retrieved information is categorised based on its presence in the source material. The system classifies the available knowledge into four states: *None*, *only in the vector database*, *only in Wikipedia*, or *both*.
2. **Confidence scoring:** The framework independently calculates a confidence score for the vector database result and the Wikipedia result. The scoring mechanism is evaluated based on the semantic alignment and information density of the retrieved information relative to the user query.

3.5 The SLM Judge and Final Answer Generation

The most critical phase of the architecture is the SLM Judge. Rather than naively summarising the retrieved document, the SLM acts as an adjudicator: it interprets the query, then analyses the retrieved information on the basis of confidence scoring and verified facts relevant to that query.

After analysing all retrieved evidence, the SLM Judge resolves contradictions between internal and external information. If the internal document lacks sufficient detail (resulting in a low internal confidence score), the Judge dynamically shifts reliance to the verified external facts, thereby eliminating the model's need to hallucinate missing information. The output is a highly faithful, query-focused summarisation grounded in verifiable evidence.

4 Algorithm

Algorithm 1 Agentic Conflict-Resolution and Dual-Path Retrieval

Input: Q (Query), V_{db} (FAISS), W_{ext} (Wikipedia), M_{slm} (Judge), τ (Threshold)
Output: A (Verified Summary)

$Q_{emb} \Leftarrow \text{PreprocessAndEmbed}(Q)$
Phase 1: Parallel Dual-Path Retrieval
  $C_{int} \Leftarrow \text{Search}(V_{db}, Q_{emb})$ ▷ Internal Search
  $C_{ext} \Leftarrow \text{Search}(W_{ext}, Q)$ ▷ External Search
Phase 2: Confidence Scoring
$Score_{int} \Leftarrow M_{slm}.\text{CalculateConfidence}(Q, C_{int})$
$Score_{ext} \Leftarrow M_{slm}.\text{CalculateConfidence}(Q, C_{ext})$
Phase 3: Dynamic Conflict Resolution (The SLM Judge)
if $Score_{int} \geq \tau$ **then**
  $C_{opt} \Leftarrow C_{int}$
else if $Score_{ext} > Score_{int}$ **then**
  $C_{opt} \Leftarrow C_{ext}$
else if $Score_{int} < \tau \wedge Score_{ext} < \tau$ **then**
  return $A \Leftarrow$ "Insufficient verifiable data"
end if
Phase 4: Final Generation
$A \Leftarrow M_{slm}.\text{GenerateSummary}(Q, C_{opt})$
return A

4.1 Agentic Conflict-Resolution Framework

Algorithm 1 presents an agentic Retrieval-Augmented Generation (RAG) framework that dynamically resolves conflicts between internal and external knowledge sources. The system employs parallel dual-path retrieval followed by confidence scoring using a small language model (SLM) judge to select optimal context.

The algorithm processes each query Q through four distinct phases. Phase 1 executes simultaneous retrieval: internal search leverages FAISS vector database (V_{db}) for document retrieval, while external search queries Wikipedia (W_{ext}) for web-based evidence. This parallel approach ensures comprehensive context coverage within a single processing cycle.

Phase 2 utilises the SLM judge (M_{slm}) to independently evaluate confidence scores for both contexts ($Score_{int}$, $Score_{ext}$) against the original query. Phase 3 implements threshold-based conflict resolution with threshold τ: internal context prevails if $Score_{int} \geq \tau$; external context is selected if $Score_{ext} > Score_{int}$; both failing threshold returns "Insufficient verifiable data".

Phase 4 generates the verified summary A using the optimally selected context C_{opt}. This agentic architecture enhances traditional RAG systems by incorporating autonomous judging, confidence calibration, and robust fallback mechanisms that prioritise verifiable evidence over hallucination risk.

Algorithm 2 Verified Facts Categorisation and Confidence Scoring

Input:
Q: User Query
C_{int}: RetrievedInternalContext(VectorDatabase)
C_{ext}: RetrievedExternalContext(Wikipedia)
Output:
K_{state}: Categorised knowledge state $\in$ {None, Only Int, Only Ext, Both}
$Score_{int}$, $Score_{ext}$: Evaluated confidence scores

Step 1: Fact and Claim Extraction
Internal Factors: $F_{int} \Leftarrow$ ExtractClaims(C_{int})
External Factors: $F_{ext} \Leftarrow$ ExtractClaims(C_{ext})
Step 2: Verified Facts Categorisation
if $F_{int} = \emptyset \wedge F_{ext} = \emptyset$ **then**
    $K_{state} \Leftarrow$ None
else if $F_{int} \neq \emptyset \wedge F_{ext} = \emptyset$ **then**
    $K_{state} \Leftarrow$ Only in Vector Database
else if $F_{int} = \emptyset \wedge F_{ext} \neq \emptyset$ **then**
    $K_{state} \Leftarrow$ Only in Wikipedia
else
    $K_{state} \Leftarrow$ Both
end if
Step 3: Confidence Scoring (Semantic Alignment & Density)
$Score_{int} \Leftarrow M_{slm}$.Evaluate_Alignment($Q, F_{int}$)
$Score_{ext} \Leftarrow M_{slm}$.Evaluate_Alignment($Q, F_{ext}$)
return $K_{state}, Score_{int}, Score_{ext}$

4.2 Verified Facts Categorisation Framework

Algorithm 2 implements a systematic approach for categorising verified facts extracted from dual knowledge sources and computing their confidence scores relative to the user query. This mechanism enables precise knowledge state determination before final response generation.

The algorithm first extracts claims from both internal vector database context (C_{int}) and external Wikipedia context (C_{ext}), producing fact sets F_{int} and F_{ext} respectively. It then categorises the knowledge state K_{state} into four mutually exclusive conditions: None (both empty), Only in Vector Database, Only in Wikipedia, or Both (facts present in both sources).

Step 3 employs the SLM judge (M_{slm}) to evaluate semantic alignment between the original query Q and extracted facts from each source. Confidence scores $Score_{int}$ and $Score_{ext}$ quantify relevance and factual density, providing a quantitative basis for subsequent context selection and agentic decision-making.

This verification layer complements Algorithm 1 by establishing ground truth knowledge availability before confidence-based conflict resolution. The categorical output enables nuanced handling of partial verification scenarios whilst maintaining robust fallback mechanisms for insufficient evidence cases.

5 Experimental Setup

To assess the performance, latency, and factual reliability of the Agent-Verified architecture, a controlled evaluation environment was created. The setup was explicitly designed to test system performance on resource-constrained edge devices using established benchmark datasets.

5.1 Hardware and Software Environment

All experiments were performed on a local MSI system featuring an Intel Core i5 12th Gen H processor and an NVIDIA RTX 4050 GPU (45 W), restricted to a maximum of 6 GB of VRAM, reflecting realistic edge-device constraints and ensuring data privacy. The software workflow was managed using Python, with local model inference facilitated by the Ollama framework.

5.2 Benchmark Datasets

A variety of benchmark datasets, covering both single-hop and multi-hop question-answering scenarios, were selected in order to thoroughly assess the proposed system. These datasets represent different degrees of reasoning difficulty, document diversity, and factual richness, providing a consistent and repeatable assessment framework aligned with current NLP research standards.

5.2.1 TriviaQA

TriviaQA [14] contains question-answer pairs sourced from trivia enthusiasts and Wikipedia, requiring reasoning across question, answer, and evidence documents. Questions demand multi-document comprehension beyond simple keyword matching.

5.2.2 WebQuestions

WebQuestions contains complex questions drawn from Google search logs, paired with Freebase entity answers. It tests open-domain single-hop retrieval and entity linking from real user queries.

5.2.3 MS MARCO

MS MARCO [13] (Microsoft Machine Reading Comprehension) focuses on passage ranking and extractive QA from noisy web data.

5.2.4 HotpotQA

HotpotQA [15] is a Wikipedia-based question-answering dataset requiring multi-hop reasoning across multiple documents. Key features include finding and combining distant facts, diverse reasoning patterns (comparison, bridging), and sentence-level supporting facts for explainability.

5.2.5 2WikiMultiHop

2WikiMultiHop [18] derives multi-hop questions from WikiHop by composing two single-hop questions, testing reasoning chains that require two sequential passages.

5.3 Model Selection

The system was evaluated across a range of SLMs to identify the best trade-off between factual accuracy and inference latency on the constrained GPU. Models assessed included Gemma 3 (4B) [35], Llama 3.2 (3B) [36], Llama 3.1 [36], and Ministral 3 (3B) [37], alongside significantly larger models: DeepSeek-R1 (7B) [38] and Qwen 3 (8B) [39] to establish latency failure thresholds.

5.4 Evaluation Metrics and Architectural Validation

Evaluating an agent-driven, multi-source RAG system requires metrics that reflect factual grounding rather than mere text replication. The following metrics were monitored:

5.4.1 Faithfulness

Evaluates the degree to which generated responses are factually correct relative to the input context, specifically penalising unsubstantiated assertions. Higher values on a 0–1 scale indicate stronger grounding in the retrieved evidence.

$$\text{Faithfulness Score} = \frac{|C_{\text{sup}}|}{|C_{\text{total}}|} \tag{1}$$

Equation 1 defines the Faithfulness Score as the ratio of supporting claims ($|C_{\text{sup}}|$) to total claims ($|C_{\text{total}}|$) extracted from the generated response. This metric quantifies factual grounding by measuring the proportion of verifiable statements against the complete set of claims made.

5.4.2 Context Relevance

Assesses how relevant the retrieved passages are to the query and quantifies the proportion of irrelevant noise.

$$\text{CR} = \frac{|\{s \in D \mid s \text{ is relevant to } q\}|}{|D|} \tag{2}$$

This equation defines the Context Relevance (CR) as the proportion of statements s in document set D that are relevant to query q. The metric measures retrieval quality by calculating the fraction of relevant content retrieved from the total document collection.

5.4.3 Semantic Similarity

Captures meaning beyond exact word matches by computing the cosine similarity between the generated response and the gold reference.

$$\text{SS} = \cos\theta = \frac{\vec{e}_g \cdot \vec{e}_r}{\|\vec{e}_g\| \, \|\vec{e}_r\|} \tag{3}$$

where $\vec{e}_g$ and $\vec{e}_r$ are the dense embedding vectors of the generated response and the gold reference, respectively.

5.4.4 Pipeline Latency

The average inference time in seconds per query, serving as a measure of real-world usability on edge hardware.

$$L_{\text{total}} = L_{\text{ret}} + L_{\text{verify}} + L_{\text{gen}} \tag{4}$$

where L_{ret} is retrieval latency, L_{verify} is the SLM Judge verification latency, and L_{gen} is the generation latency, all measured in milliseconds.

5.4.5 Source Attribution

Monitors whether the model accurately references supporting details from the retrieved documents.

$$\text{SA} = \frac{\left|\{c_i \in \hat{S} \mid c_i \text{ is traceable to a source document}\}\right|}{\left|\hat{S}\right|} \tag{5}$$

where $\hat{S}$ is the set of all claims in the generated summary and c_i is an individual claim.

5.4.6 ROUGE-1

Evaluates the overlap of unigrams between the generated summary and the reference text, primarily capturing content coverage and informational recall.

$$\text{ROUGE-1} = \frac{\sum_{s \in R} \sum_{w \in s} \mathbb{1}\left[w \in \hat{S}\right]}{\sum_{s \in R} |s|} \tag{6}$$

This equation defines ROUGE-1 as the fraction of words from reference sentences R appearing in generated summary $\hat{S}$. The metric computes unigram overlap precision by summing matching word indicators across all reference sentence words, normalised by total reference word count.

5.4.7 ROUGE-2

Measures the overlap of bigrams (pairs of consecutive words) to assess the fluency and structural consistency of the generated output against the ground truth.

$$\text{ROUGE-2} = \frac{\sum_{s \in R} \sum_{b \in \text{bigrams}(s)} \mathbb{1}\left[b \in \hat{S}\right]}{\sum_{s \in R} |\text{bigrams}(s)|} \tag{7}$$

This equation defines ROUGE-2 as the fraction of bigrams from reference sentences R appearing in generated summary $\hat{S}$. The metric computes bigram overlap precision by summing matching bigram indicators across all reference sentence bigrams, normalised by total reference bigram count.

5.4.8 ROUGE-L

Utilises the Longest Common Subsequence (LCS) to identify the longest co-occurring in-sequence n-grams, reflecting how well the model preserves sentence structure and semantic flow.

$$\text{ROUGE-L} = \frac{(1 + \beta^2) \cdot P_{lcs} \cdot R_{lcs}}{R_{lcs} + \beta^2 \cdot P_{lcs}} \tag{8}$$

Equation 8 defines ROUGE-L as the F1-measure of the longest common subsequence (LCS) between the reference R and generated summary $\hat{S}$. The metric combines LCS precision P_{lcs} and recall R_{lcs} using a weighting parameter β, prioritising recall when $\beta > 1$.

5.4.9 BLEU

A precision-oriented metric that calculates n-gram matches between the candidate and reference answers, incorporating a brevity penalty to ensure that generated responses are of appropriate length.

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{9}$$

This equation defines BLEU score as brevity penalty BP multiplied by weighted geometric mean of n-gram precisions p_n. The metric combines precision across multiple n-gram orders ($n = 1$ to N) with exponentially weighted logarithm, penalising overly short generations through BP. where the Brevity Penalty is:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{otherwise} \end{cases} \tag{10}$$

This equation defines the Brevity Penalty (BP) in BLEU score calculation. It applies no penalty when candidate length c exceeds reference length r, but exponentially

penalises overly short candidates via $\exp(1 - r/c)$ when $c \leq r$. The modified n-gram precision is:

$$p_n = \frac{\sum_{\mathcal{C} \in \text{Candidates}} \sum_{n\text{-gram} \in \mathcal{C}} \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{\mathcal{C} \in \text{Candidates}} \sum_{n\text{-gram} \in \mathcal{C}} \text{Count}(n\text{-gram})} \tag{11}$$

This equation represents the modified n-gram precision, p_n, which is a foundational component of the BLEU metric used to evaluate machine translation systems. It calculates the proportion of n-grams in the candidate texts that appear in the reference translations, with the matching counts "clipped" to ensure repeated words do not artificially inflate the score.

6 Results

To evaluate the Agent-Verified system, experiments were performed using multiple Small Language Models (SLMs) and benchmarked against five datasets: TriviaQA, WebQuestions, MS MARCO, HotpotQA, and 2WikiMultiHop with 10 samples per dataset (50 total). The evaluation prioritised metrics specifically designed for retrieval-augmented architectures, including Faithfulness, Answer Relevance, Context Relevance, Semantic Similarity, Latency, ROUGE-1/2/L F1, BLEU, Average Retrieval Similarity, Number of Verified Facts, and Number of Conflicts.

6.1 Faithfulness and Agentic Verification

The system's core novelty is the dual-path verification loop, which was evaluated using the Faithfulness metric a measure of the factual accuracy of the generated answer relative to the input context that penalises hallucinations.

Table 1: Faithfulness metrics of SLM models

Models	TriviaQA (Single Hop)	WebQuestions (Single Hop)	MS MARCO (Single Hop)	HotpotQA (Multi Hop)	2WikiMultiHop (Multi Hop)
Llama 3.2 (3B)	0.69	0.74	**0.81**	**0.67**	0.69
Gemma 3 (4B)	0.32	0.78	0.39	0.18	0.32
Qwen 3 (8B)	0.30	0.55	0.30	0.40	0.30
Llama 3.1 (Latest)	**0.73**	**0.83**	0.50	0.36	**0.73**
Ministral 3 (3B)	0.46	0.49	0.35	0.20	0.46
DeepSeek-R1 (7B)	0.61	0.58	0.56	0.32	0.61

Note. Bold values indicate the highest faithfulness score achieved for each dataset. Scores range from 0 to 1, where higher values represent greater adherence to the source context.

For certain SLMs, integrating external knowledge with local vector databases produced strong results. As shown in Table 1, Llama 3.2 (3B) and Llama 3.1 clearly

dominated factual grounding across both single-hop and multi-hop datasets, whereas models such as Ministral 3 and Qwen 3 struggled to maintain reliability.

The system was also tested exclusively against local documents, and the results are consistent with the full-pipeline findings. The performance differences between SLMs persisted in this setting. Table 2 further validates the 3B and 4B Llama and Gemma architectures as the strongest foundation for this framework.

Table 2: Average System Faithfulness (Local Document Processing)

Model	**Parameter Size**	**Average Faithfulness**	**Performance Tier**
Llama 3.1 (Latest)	8B	80%	Robust
Gemma 3	4B	78%	Robust
Qwen 3	8B	33%	Poor Factual Grounding
Ministral 3	3B	30%	Poor Factual Grounding

Note. Faithfulness scores reflect the model's ability to generate responses strictly grounded in the provided local documents, minimising hallucinations. Models scoring above 75% are classified as "Robust," demonstrating reliable factual adherence. Conversely, scores below 50% fall into the "Poor Factual Grounding" tier, indicating a high propensity to generate information outside the given context.

Table 3: Agentic Conflict-Resolution (Confidence Score Asymmetry)

Query Instance	Evaluated Model	Internal (PDF) Confidence	External (Web) Confidence	Agent Action
Query 1	Llama 3.1	0.50	0.90	Shifted to External
Query 2	Qwen 3	0.00	0.50	Shifted to External

Note. This table illustrates scenarios where the routing agent detected a significant asymmetry between confidence scores. Because the external web context yielded a higher confidence than the internal PDF data, the agent autonomously shifted its retrieval strategy to prioritise the external source.

6.2 Performance Across Complex Datasets

From Table 1 and Table 4, the framework underwent additional stress testing using intricate single-hop and multi-hop datasets. On the MS MARCO dataset, which comprises question-answer pairs extracted from noisy web data, Llama 3.2 (3B) achieved a strong Faithfulness of 0.81 and a Semantic Similarity of 0.75.

The system also maintained consistent contextual correctness on multi-hop reasoning datasets that require synthesising information across distant passages, such as

Table 4: Core RAG Performance Metrics

Models	Dataset	Category	Answer Rel.	Context Rel.	Source Att.	Semantic Sim.
Llama 3.2 (3B)	TriviaQA	Single Hop	0.81	0.82	0.80	0.10
	WebQuestions	Single Hop	0.71	0.75	0.77	0.35
	MS MARCO	Single Hop	0.77	0.66	0.77	0.75
	HotpotQA	Multi Hop	0.71	0.81	0.77	0.32
	2WikiMulti Hop	Multi Hop	0.81	0.82	0.80	0.10
Gemma 3 (4B)	TriviaQA	Single Hop	0.81	0.82	0.66	0.08
	WebQuestions	Single Hop	0.77	0.75	0.71	0.33
	MS MARCO	Single Hop	0.83	0.66	0.67	0.75
	HotpotQA	Multi Hop	0.67	0.81	0.68	0.30
	2WikiMulti Hop	Multi Hop	0.81	0.82	0.66	0.08
Qwen 3 (8B)	TriviaQA	Single Hop	0.85	0.82	0.65	0.08
	WebQuestions	Single Hop	0.79	0.75	0.57	0.33
	MS MARCO	Single Hop	0.67	0.66	0.55	0.73
	HotpotQA	Multi Hop	0.55	0.81	0.80	0.30
	2WikiMulti Hop	Multi Hop	0.85	0.82	0.65	0.08
Llama 3.1 (Latest)	TriviaQA	Single Hop	0.79	0.82	0.77	0.11
	WebQuestions	Single Hop	0.81	0.75	0.70	0.36
	MS MARCO	Single Hop	0.79	0.66	0.73	0.75
	HotpotQA	Multi Hop	0.80	0.81	0.72	0.33
	2WikiMulti Hop	Multi Hop	0.79	0.82	0.79	0.11
Ministral 3 (3B)	TriviaQA	Single Hop	0.88	0.82	0.47	0.08
	WebQuestions	Single Hop	0.72	0.75	0.47	0.30
	MS MARCO	Single Hop	0.82	0.66	0.38	0.68
	HotpotQA	Multi Hop	0.70	0.81	0.39	0.31
	2WikiMulti Hop	Multi Hop	0.88	0.82	0.47	0.08

HotpotQA and 2WikiMultiHop. Specifically, Llama 3.2 (3B) achieved a Context Relevance of 0.81 on HotpotQA. Gemma 3 (4B) attained the same Context Relevance of 0.81 on HotpotQA, further confirming that the retrieval architecture reliably surfaces relevant passages with minimal irrelevant noise.

6.3 Evaluation Metric Limitations: The Fallacy of N-Gram Scoring

The traditional exact phrase-matching metrics, ROUGE-1/2/L and BLEU, are reported in Table 5 and consistently produced near-zero scores across all models and datasets. Gemma 3 achieved an ROUGE-1 F1 of only 0.01 on TriviaQA, whilst Llama 3.2 scored 0.00 on the same dataset.

These results do not reflect a failure in factual generation; rather, they expose a fundamental limitation of n-gram overlap metrics when applied to abstractive, generative RAG systems. ROUGE and BLEU heavily penalise outputs that diverge from the exact phrasing of the reference text, yet the proposed SLM-Judge architecture produces highly concise, synthesised answers drawn from multiple retrieved passages. The system's true capability is more accurately captured by the high Semantic Similarity and Faithfulness scores reported in Table 1 and Table 4, which measure meaning preservation and factual grounding beyond superficial keyword overlap.

Table 5: Baseline N-Gram Evaluation

Models	Dataset	Category	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
Llama 3.2 (3B)	TriviaQA	Single Hop	0.00	0.00	0.00	0.00
	WebQuestions	Single Hop	0.02	0.00	0.02	0.00
	MS MARCO	Single Hop	0.09	0.04	0.08	0.03
	HotpotQA	Multi Hop	0.01	0.00	0.01	0.00
	2WikiMultiHop	Multi Hop	0.00	0.00	0.00	0.00
Gemma 3 (4B)	TriviaQA	Single Hop	0.01	0.00	0.01	0.00
	WebQuestions	Single Hop	0.03	0.00	0.02	0.01
	MS MARCO	Single Hop	0.12	0.05	0.10	0.03
	HotpotQA	Multi Hop	0.01	0.00	0.01	0.00
	2WikiMultiHop	Multi Hop	0.01	0.00	0.01	0.00
Qwen 3 (8B)	TriviaQA	Single Hop	0.00	0.00	0.00	0.00
	WebQuestions	Single Hop	0.01	0.00	0.01	0.00
	MS MARCO	Single Hop	0.10	0.03	0.08	0.03
	HotpotQA	Multi Hop	0.01	0.00	0.01	0.00
	2WikiMultiHop	Multi Hop	0.00	0.00	0.00	0.00
Llama 3.1 (Latest)	TriviaQA	Single Hop	0.00	0.00	0.00	0.00
	WebQuestions	Single Hop	0.01	0.00	0.01	0.00
	MS MARCO	Single Hop	0.06	0.01	0.05	0.01
	HotpotQA	Multi Hop	0.00	0.00	0.00	0.00
	2WikiMultiHop	Multi Hop	0.00	0.00	0.00	0.00
Ministral 3 (3B)	TriviaQA	Single Hop	0.00	0.00	0.00	0.00
	WebQuestions	Single Hop	0.02	0.00	0.02	0.01
	MS MARCO	Single Hop	0.12	0.03	0.09	0.03
	HotpotQA	Multi Hop	0.01	0.00	0.01	0.00
	2WikiMultiHop	Multi Hop	0.00	0.00	0.00	0.00

6.4 Hardware-Constrained Latency Analysis

As shown in Figure 2, models exceeding the 5B parameter threshold experienced substantial latency spikes that render them unusable for real-time edge applications. Qwen 3 (8B) recorded a total pipeline latency of 122.81 seconds, and DeepSeek-R1 (7B) recorded 122.93 seconds, confirming that both large models far exceed acceptable latency bounds for edge deployment.

The most optimised SLMs performed well within the hardware limitations. Llama 3.2 (3B) achieved the lowest latency at 20.88 seconds, with the retrieval phase consuming only 0.25 seconds. Gemma 3 (4B) also performed efficiently, with a total pipeline latency of 30.37 seconds and a retrieval time of 4.52 seconds. These results validate that 3B and 4B parameter models represent the optimal architecture for balancing computational cost and real-time responsiveness on constrained hardware.

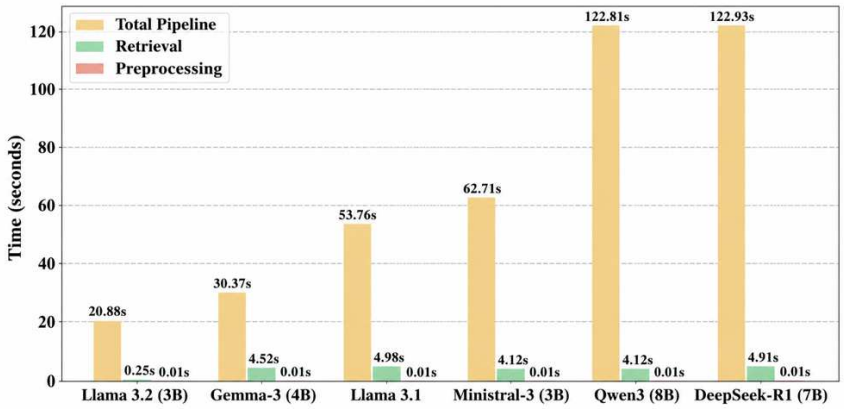


Fig. 2: Pipeline, Retrieval, and Pre-processing Latency by Model

7 Conclusion

Ensuring the reliability of generative models in limited environments has become a concern as SLMs get increasingly integrated into essential uses. Architectures that can independently verify facts without depending on computationally expensive infrastructure are required to address this, and to avoid factual hallucinations in query-focused summarisation on an edge device, this paper illustrates an Agent-Verified Retrieval-Augmented Generation system.

For dynamic conflict handling, it uses a parallel dual-path architecture which combines Wikipedia agent verification with local retrieval and is resolved by an SLM Judge.

8 Declarations

- **Funding:** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
- **Conflicts of interest/Competing interests:** The authors have no relevant financial or non-financial interests to disclose.
- **Availability of data and materials:** The datasets analysed during the current study (TriviaQA, WebQuestions, MS MARCO, HotpotQA, and 2WikiMultiHop) are publicly available in standard natural language processing repositories.
- **Code availability:** The local SLM inference and Agent-Verified routing scripts used during the current study are available from the corresponding author on reasonable request.
- **Authors' contributions:** Shanta Majumder conceived the presented idea, developed the framework, performed the computations, and drafted the manuscript. Suprove Chandra Sarkar supervised the research, verified the analytical methods, and reviewed the manuscript. Both authors read and approved the final manuscript.

References

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.

[2] Singh, A., Ehtesham, A., Kumar, S., & Khoei, T. T. (2025). Agentic retrieval-augmented generation: A survey on agentic RAG. *arXiv preprint arXiv:2501.09136*.

[3] Zhang, Y., Sun, R., Chen, Y., Pfister, T., Zhang, R., & Arik, S. (2024). Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37, 132208–132237.

[4] Jiang, Z., Sun, M., Liang, L., & Zhang, Z. (2025). Retrieve, summarise, plan: Advancing multi-hop question answering with an iterative approach. In *Companion Proceedings of the ACM on Web Conference 2025* (pp. 1677–1686).

[5] Han, S., Xia, P., Zhang, R., Sun, T., Li, Y., Zhu, H., & Yao, H. (2025). MDocAgent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*.

[6] Chang, C. Y., Jiang, Z., Rakesh, V., Pan, M., Yeh, C. C. M., Wang, G., ... & Zou, N. (2025). MAIN-RAG: Multi-agent filtering retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2607–2622).

[7] Chen, T. C., Tang, C. W., & Thomas, C. (2024). MetaSumPerceiver: Multimodal multi-document evidence summarisation for fact-checking. *arXiv preprint arXiv:2407.13089*.

[8] Li, W., Xiao, X., Liu, J., Wu, H., Wang, H., & Du, J. (2020). Leveraging graph to improve abstractive multi-document summarisation. *arXiv preprint arXiv:2005.10043*.

[9] Zhang, D., Feng, Y., Liu, H., Sun, C., Luo, J., Chen, X., & Li, X. (2025). Conversational agents: From RAG to LTM. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region* (pp. 447–452).

[10] Jhandi, P., Kazi, O., Subramanian, S., & Sendas, N. (2025). Small language models for efficient agentic tool calling: Outperforming large models with targeted fine-tuning. *arXiv preprint arXiv:2512.15943*.

[11] Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* (pp. 74–81).

[12] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311–318).

[13] Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., ... & Wang, T. (2016). MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

[14] Joshi, M., Choi, E., Weld, D. S., & Zettlemoyer, L. (2017). TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1601–1611).

[15] Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., & Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 2369–2380).

[16] Jacovi, A., & Goldberg, Y. (2020). Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4198–4205).

[17] Lesk, M. E., & Salton, G. (1968). Relevance assessments and retrieval system evaluation. *Information Storage and Retrieval*, 4(4), 343–359.

[18] Ho, X., Nguyen, A. K. D., Sugawara, S., & Aizawa, A. (2020). Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 6609–6625).

[19] Sun, S., Krishna, K., Mattarella-Micke, A., & Iyyer, M. (2021). Do long-range language models actually use long-range context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 807–822).

[20] An, S., Ma, Z., Lin, Z., Zheng, N., Lou, J. G., & Chen, W. (2024). Make your LLM fully utilise the context. *Advances in Neural Information Processing Systems*, 37, 62160–62188.

[21] Chen, S., Wong, S., Chen, L., & Tian, Y. (2023). Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

[22] Jiang, H., Wu, Q., Luo, X., Li, D., Lin, C. Y., Yang, Y., & Qiu, L. (2024). LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1658–1677).

[23] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

[24] Chen, J., Lin, H., Han, X., & Sun, L. (2024). Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 16, pp. 17754–17762).

[25] Baumel, T., Eyal, M., & Elhadad, M. (2018). Query focused abstractive summarisation: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.

[26] Celikyilmaz, A., Bosselut, A., He, X., & Choi, Y. (2018). Deep communicating agents for abstractive summarisation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 1662–1675).

[27] Perez-Beltrachini, L., Liu, Y., & Lapata, M. (2019). Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5107–5116).

[28] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., ... & Larson, J. (2024). From local to global: A graph RAG approach to query-focused summarisation. *arXiv preprint arXiv:2404.16130*.

[29] Xu, Z., Cruz, M. J., Guevara, M., Wang, T., Deshpande, M., Wang, X., & Li, Z. (2024). Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th International ACM SIGIR*

Conference on Research and Development in Information Retrieval (pp. 2905–2909).

[30] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P. E., ... & Jégou, H. (2025). The FAISS library. *IEEE Transactions on Big Data*.

[31] Wang, J., & Duan, Z. (2024). Agent AI with LangGraph: A modular framework for enhancing machine translation using large language models. *arXiv preprint arXiv:2412.03801*.

[32] Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1870–1879).

[33] Belcak, P., Heinrich, G., Diao, S., Fu, Y., Dong, X., Muralidharan, S., ... & Molchanov, P. (2025). Small language models are the future of agentic AI. *arXiv preprint arXiv:2506.02153*.

[34] Ni, B., & Buehler, M. J. (2024). MechAgents: Large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge. *Extreme Mechanics Letters*, 67, 102131.

[35] Gemma Team, Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., ... & Google DeepMind. (2025). Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

[36] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., ... & Meta AI. (2024). The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[37] Liu, A. H., Khandelwal, K., Subramanian, S., Jouault, V., Rastogi, A., Sadé, A., ... & Mistral AI. (2026). Ministral 3. *arXiv preprint arXiv:2601.08584*.

[38] DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., ... & Liang, W. (2025). DeepSeek-R1: Incentivising reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

[39] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., ... & Qwen Team. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.