SANCTUARY: An Efficient Evidence-Based Automated Fact Checking System

Arbaaz Dharamvaram
Faculty of Computer Science,
University of New Brunswick, Canada
arbaaz.dm@unb.ca

Saqib Hakak
Faculty of Computer Science,
University of New Brunswick, Canada
saqib.hakak@unb.ca

Abstract

With the growing volume of misinformation online, automated fact-checking systems are becoming increasingly important. This paper presents SANCTUARY, an efficient pipeline for evidence-based verification of real-world claims. Our approach consists of three stages: Hypothetical Question & Passage Generation, a two-step Retrieval-Augmented Generation (RAG) hybrid evidence retrieval, and structured reasoning and prediction, which leverages two lightweight Large Language Models (LLMs). On the challenging AVeriTeC benchmark, our system achieves 25.27 points on the new AVeriTeC score (Ev2R recall), outperforming the previous state-of-the-art baseline by 5 absolute points (1.25× relative improvement). Sanctuary demonstrates that careful retrieval, reasoning strategies and well-integrated language models can substantially advance automated fact-checking performance.

1 Introduction

The ease with which information can be published and amplified online has intensified longstanding concerns about the spread of misinformation and disinformation (Lewandowsky et al., 2020; Schlichtkrull et al., 2024). Professional fact-checking organizations such as PolitiFact[1], FactCheck.org[2] and Snopes[3] have scaled up their efforts, yet the sheer volume and velocity of claims far outstrip human capacity (Nakov et al., 2021). Moreover, not every claim warrants fact-checking; resources should be directed toward content that can significantly impact society, such as influencing elections (Allcott and Gentzkow, 2017) or causing financial harm (Gold and Stelter, 2025). Consequently, Automated Fact-Checking (AFC) has emerged as a promising assistive technology aimed at (i) identifying check-worthy claims, (ii) retrieving or generating relevant evidence, and (iii) predicting a veracity verdict transparently to bolster public trust and adoption (Vlachos and Riedel, 2014; Thorne and Vlachos, 2018).

Figure 1 illustrates a real-world check-worthy claim, showing the kind of input and output that a fact-checking system must process and output.

Claim: Several First Nations communities in Canada have closed their borders to avoid COVID-19.
Date: 19-3-2020
Speaker: Chief David Monias
Reporting Source: Reuters news agency
Location Code: CA
Label: Supported
Justification: Multiple sources confirm some First Nations communities in Canada closed their borders or set up checkpoints to limit the spread of COVID-19 and protect vulnerable members.

Figure 1: A sample claim from the AVeriTeC dataset.

We introduce a lightweight, time-efficient pipeline for automated fact verification. The system assigns each claim to one of four verdicts – *Supported*, *Refuted*, *Not Enough Evidence (NEE)*, or *Conflicting Evidence/Cherrypicking (CE/C)* – and outputs a rationale explaining its decision, citing relevant sources. Our method involves applying claim decomposition, Large Language Models (LLMs), a Retrieval Augmented Generation (RAG) framework, hybrid retrieval, and carefully tuned prompts to produce explainable fact-checking.

We evaluate our system on the AVeriTeC dataset (Schlichtkrull et al., 2023), demonstrating a substantial accuracy improvement (5%) over the official baseline, while maintaining slightly faster performance.

2 Related Work

The 2024 AVeriTeC shared task (Schlichtkrull et al., 2024) commenced with an initial baseline system

[1] www.politifact.com
[2] www.factcheck.org
[3] www.snopes.com

Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER), pages 247–257

proposed by (Schlichtkrull et al., 2023), which utilized BM25-based sentence ranking (Trotman et al., 2014), question generation with BLOOM (Le Scao et al., 2023), and BERT (Devlin et al., 2019) for verdict prediction, achieving a modest old AVeriTeC score of 11%. Following this initial benchmark, multiple systems explored various sophisticated techniques to significantly enhance performance. For instance, HerO (Yoon et al., 2024) introduced LLM-based prompting to generate hypothetical "evidence passages" prior to iterative BM25 retrieval, re-ranking, and subsequent question generation, yielding the highest question generation score and substantially outperforming the original baseline.

Subsequent entrants further evolved these techniques by incorporating advanced retrieval and generation strategies. InFact (Rothermel et al., 2024) achieved top position on the leaderboard by leveraging proprietary LLMs such as GPT-4o (Achiam et al., 2023) in conjunction with dense semantic retrievers, combined with an aggressive question-fan-out strategy to further increase evidence recall. AIC CTU (Ullrich et al., 2024) adopted a streamlined RAG (Lewis et al., 2020) approach, integrating innovative methods such as Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) and structured Chain-of-Thought (CoT) prompting. Dunamu-ML (Park et al., 2024) expanded the coverage of evidence by incorporating overlooked resources such as PDFs and video transcripts, demonstrating the benefits of richer evidence sources. Finally, Papelo (Malon, 2024) utilized a dynamic multi-hop web search approach with iterative question generation conditioned on previous results, highlighting potential advantages in scenarios where an initial fixed corpus might not be readily available.

Collectively, these systems underscore a progression from an initial shallow baseline to increasingly sophisticated methods and, heavier, proprietary models, leaving plenty of room for optimizations in terms of reducing model size, improving evidence completeness, and computational efficiency.

3 Task Description

Verifying the truthfulness of real-world claims is a complex task in natural language processing. It requires reasoning over noisy, high-volume unstructured information from diverse sources, often under strict time and resource constraints. An effective fact-checking system must not only assess the factualness of a claim, but also provide verifiable, interpretable explanations.

We formalize this claim verification task as follows:

Input

The input to the system is a tuple (c, m, e) where:

- c is an open-domain natural-language claim.
- m contains minimal metadata, including:
  - publication date t
  - speaker s
  - source URL u
  - location code ℓ
- e is the set of evidence articles collected from the Web related to the claim.

Output

Our system produces a triple (v, E, J) where:

- $v \in \{$*Supported, Refuted, NEE, CE/C*$\}$ is the system's predicted veracity label.
- E is a set of question-answer-document triples $(q_i, \{a_{i-j}\}, d_{i-j})$, representing the evidence extracted to support the label, where:
  - q_i is a fact-checking sub-question derived from the original claim c.
  - $\{a_{i-j}\}$ is a set of evidence snippets relevant to q_i.
  - d_{i-j} denotes the set of source documents from which $\{a_{i-j}\}$ are extracted.
- J is the textual justification that explains how the evidence E collectively supports the predicted veracity label v.

4 The Sanctuary System

This section outlines the architecture and workflow of Sanctuary, our end-to-end fact-checking system, as shown in Figure 2. The system consists of three main stages: Hypothetical Question & Passage Generation, a two-step Evidence-Retrieval Pipeline – Coarse and Semantic Evidence-Retrieval – and finally, Reasoning and Prediction. Each stage is designed to progressively narrow down the evidence relevant to proving the factuality of the claim and subsequent fact-checking. We describe the models, prompting strategies, and design choices used at each stage.

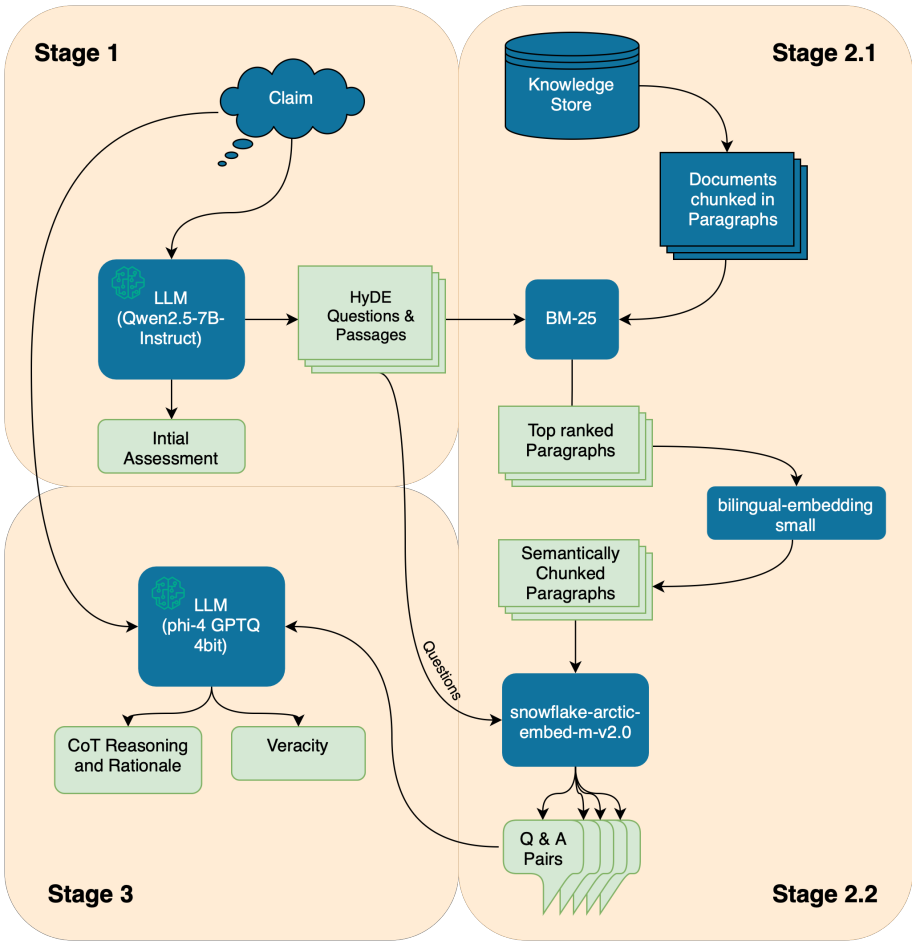

Figure 2: Overview of the SANCTUARY system

4.1 Stage 1: Hypothetical Question and Passage Generation

Fact-checkers typically approach verification by formulating multiple nuanced questions, exploring various angles to reach an informed conclusion (Silverman, 2014). Inspired by this journalistic practice, our system first employs an LLM – specifically, Qwen2.5-7B-Instruct (Yang et al., 2024) – to categorize a claim into one of three predefined veracity labels: *Supported*, *Refuted*, or *CE/C*. This classification step follows a methodology similar to that of (Park et al., 2024), with the notable exclusion of the *NEE*. This ensures that the model explicitly posits either supporting or opposing evidence; abstention would starve later stages of evidence. This initial assessment step capitalizes on the LLM's extensive pretraining and reasoning capabilities to detect factual inconsistencies based on its learned internal knowledge.

Following this classification, and drawing inspiration from Hypothetical Document Embeddings (HyDE) (Gao et al., 2023) as well as the works by (Yoon et al., 2024; Rothermel et al., 2024), we prompt the LLM to generate a diverse set of hypothetical question-document pairs using a detailed prompting template (illustrated in Figure 3). This process aims to enhance evidence recall in later stages by utilizing these hypothetical documents as queries to retrieve relevant evidence, rather than relying solely on the claim's text.

The rationale behind conducting an initial assessment prior to generating HyDE content is to condition the LLM to produce more targeted and contextually relevant hypothetical documents, leveraging its internal understanding of the claim, events, and related patterns.

Therefore, this step closely simulates the investigative process employed by human fact-checkers, who typically formulate investigative queries and the kind of evidential responses or documents they would likely obtain from online sources. By explicitly prompting the model to generate plausible yet fictional questions, associated passages, relevant entities, and alternative event scenarios, we aim to fully exploit the LLM's internal knowledge base

and reasoning strengths.

4.2 Stage 2.1: Coarse Evidence-Retrieval

After generating a set of hypothetical question-answer pairs, we use the BM25 algorithm (Trotman et al., 2014) to retrieve relevant textual evidence from the knowledge store. Standard BM25 retrieval often suffers when documents vary significantly in length, as longer documents can disproportionately penalize or dilute term frequency scores. To mitigate this, we segment each document into fixed-length segments of approximately 500 tokens using a sliding-window approach with zero overlap, in line with previous methodologies (Malon, 2024; Park et al., 2024; Ullrich et al., 2024). Each document in the corpus is chunked in this manner to construct the BM25 index.

For each generated question-answer pair (treated as a single query), we retrieve the top 125 ranked document chunks from the index, a cutoff chosen based on the development set balancing recall and runtime. Given the substantial volume and length of the source documents – averaging 794 documents per claim in the development set – we preprocess (tokenize, segment, and index) them in parallel using multiprocessing to improve efficiency.

Finally, we group the top-retrieved chunks by their originating web article, using the document's URL as the key. This step intuitively narrows down the retrieved content to those segments within Web articles most relevant to answering queries associated with fact-checking the claim.

4.3 Stage 2.2: Semantic Evidence-Retrieval

In this stage, we further refine the evidence segments using a semantic, sentence-based chunking strategy (Kamradt, 2024) followed by semantic document retrieval, shifting the focus from keyword matching to semantic relevance. By applying semantic filtering, we narrow the scope of evidence to textual snippets that precisely address the hypothetical questions generated in Stage 1.

Initially, for each retrieved evidence document, we consolidate the segments selected in the previous step. We then divide this combined text into individual sentences and encode each sentence into semantic vector representations. Sentences exhibiting high semantic similarity are grouped together, creating coherent, semantically themed textual chunks. This step further refines the segments obtained previously into compact semantic units, improving the granularity and relevance of the evidence.

For sentence embeddings, we use the *bilingual-embedding-small* model (Conneau et al., 2019; Nils Reimers, 2019; Thakur et al., 2020), which offers strong performance across a range of NLP tasks despite its modest size (117M parameters) and currently ranks #30 on the HuggingFace MTEB Multilingual leaderboard.

Next, both the semantically formed chunks and the previously generated Stage 1 queries are embedded using a retrieval embedding model, treating chunks as documents, and queries as search input. For this step, we adopt the *snowflake-arctic-embed-m-v2.0* model (Yu et al., 2024), which provides a strong balance of retrieval accuracy and computational efficiency. To avoid redundancy, and reduce the size of the final evidence list, chunks identified as semantically duplicate (cosine similarity greater than 0.9) are filtered out, ensuring that each query is answered by distinct evidence. Therefore, this stage ensures that only the most semantically relevant and distinct evidence is retained to address each query.

4.4 Stage 3: Reasoning and Prediction

Rather than refining questions via sub-query generation (Rothermel et al., 2024), or generating new questions after initial coarse evidence-retrieval (Yoon et al., 2024), or reframing retrieved answers (Park et al., 2024), or generating question-answer pairs from initially retrieved coarse evidence (Ullrich et al., 2024) – each of which increases computational complexity – we opt for simplicity by adhering to our initial queries from Stage 1.

For each query, we select up to 8 evidence chunks from Stage 2.2, while imposing a hard cosine similarity threshold of 0.52. Our internal subjective evaluations indicated that this threshold effectively filters out less relevant evidence.

We then incorporate the selected question-answer blocks into a carefully crafted veracity prompt template, as shown in Figure 4, which includes detailed instructions and guidelines specific to the task of fact-checking. To further guide the model's understanding of the task, we include four-shot examples, one for each veracity label. The prompt is then fed into Microsoft's Phi-4 14B LLM (Abdin et al., 2024), using its GPTQ 4-bit quantized version (Frantar et al., 2022), chosen to maintain computational efficiency while remaining competitive. Although we had the option to use

Qwen2.5-7B for this stage, we found that Phi-4 is better suited for the task, owing to its larger parameter count and better handling of nuanced reasoning, which is critical for accurate veracity classification.

Additionally, to promote structured reasoning, groundedness, and explainability in our predictions, we employ a Chain-of-Thought (CoT) prompting strategy (Wei et al., 2022), instructing the model to articulate its reasoning explicitly, citing relevant Q-A pairs before producing a veracity label.

5 Experimental Results

5.1 Dataset

The AVeriTeC dataset provides three splits of real-world fact-checking claims containing both the textual claim and rich metadata. The training set comprises **3,068** claims, the development set **500** claims, and the blind test set **1,000** claims. Although the train/dev splits are drawn from fact-checks published up to 2019, the test set only contains claims posted *from 2024 onward*, making it a strictly out-of-distribution evaluation for temporal generalization. For every claim, a bundle of crawled web pages is provided as a *knowledge store*.

5.2 Evaluation Criteria

The new scoring mechanism, as of 2025, replaces the previous Hungarian METEOR (Banerjee and Lavie, 2005) string matching-based approach with the atomic reference-based **Ev2R atomic scorer** (Akhtar et al., 2024). An LLM is used to decompose predicted and reference questions and evidence into minimal *atomic facts*. For every atomic fact in the reference-set the scorer asks whether it is supported by the prediction, computing a *recall* value.

If the **Q + A (Ev2R recall)** > 0.50, then the system's veracity label is compared with gold, producing *new AVeriTeC score*. The Recall scores are published for Q-only and Q+A, along with the final AVeriTeC score.

Why the change? Hungarian METEOR is sensitive to surface form and treats any unannotated but valid evidence as "wrong" (Akhtar et al., 2024). The Ev2R scorer rewards factual coverage regardless of wording, is robust to alternate evidence chains, and correlates better with human judgments of coverage and relevance.

5.3 Baseline

Our reference is the **HerO** system (Yoon et al., 2024), *Herd of Open LLMs*, ranked 2nd in the 2024 AVeriTeC challenge. HerO adopts a multi-stage approach combining retrieval and generative reasoning: First, it uses open-source language models to produce hypothetical fact-checking passages. Next, these passages, along with the claim guide a retrieval process, identifying relevant evidence from a large-scale, per-claim knowledge store. The retrieved evidence is further filtered through semantic embedding models to retain only the most contextually meaningful excerpts. Then, HerO generates structured questions explicitly connecting evidence back to the original claim. Finally, a language model utilizes these questions and evidence snippets to classify the claim.

5.4 Hyperparameter Choices

We present the hyperparameter configurations employed at each stage of our pipeline below:

Stage 1: We utilize `Qwen2.5-7B-Instruct`, running under the vLLM[4] inference engine with its weights cast to bfloat16 precision. To balance diversity and instruction-following, we fix the sampling parameters as follows: `temperature` = 0.5, `top_p` = 0.8, `min_p` = 0.1, and `max_tokens` = 2048. We perform batch inference with four claims processed concurrently.

Stage 2: This stage leverages two distinct models: one optimized for Semantic Textual Similarity (STS) and another for document retrieval. For STS, we use `bilingual-embedding-small`[5] with the python library `Chonkie`[6], using the following chunking parameters: `min_chunk_size` = 30, `chunk_size` = 140, `min_sentences` = 2, and `similarity_window` = 1. Since 1–2 sentences average approximately 30 tokens (OpenAI, 2025), we enforce a minimum chunk size of two sentences (30 tokens) and a maximum of approximately 4–6 sentences (140 tokens). The `similarity_window` parameter controls the number of adjacent sentences considered during the similarity threshold computation. These choices ensure that short sentences are grouped to preserve contextual integrity, while longer meaningful segments are prevented from being fragmented arbitrarily.

[4] https://github.com/vllm-project/vllm

[5] https://huggingface.co/Lajavaness/bilingual-embedding-small

[6] https://github.com/chonkie-inc/chonkie

System	Q only	Q + A	AVeriTeC Score	Avg. time/claim (s)
CTU AIC	**0.2003 ± 0.0066**	**0.4774 ± 0.0035**	**0.3317 ± 0.0015**	53.67
HUMANE	0.1933 ± 0.0048	0.4299 ± 0.0006	0.2707 ± 0.0040	**29.19**
SANCTUARY	0.1561 ± 0.0057	0.4098 ± 0.0077	0.2527 ± 0.0051	31.71
Baseline	0.2723 ± 0.0006	0.3362 ± 0.0036	0.2023 ± 0.0068	33.88

Table 1: Performance of participating systems on the AVeriTeC 2025 task. Each system is evaluated on Question-only (Q), Question + Answer (Q+A), and the unified AVeriTeC score. Average inference time per claim (in seconds) is also reported.

LLM Combination (Stages 1 and 3)	Q only	Q + A	AVeriTeC Score
Qwen2.5-7B and Phi-4 GPTQ	0.3427	0.5167	0.29
Gemini 2.5 Flash (Both Stages)	0.4004	0.6106	0.33

Table 2: Ev2R evaluation comparing our default LLM combination of Qwen2.5-7B and Phi-4 GPTQ with Gemini Flash 2.5. Metrics reported are recall scores on question-only (Q), question-plus-answer (Q+A), and the final AVeriTeC score on 100 balanced development claims.

`snowflake-arctic-embed-m-v2.0`[7] is used for retrieval, using the eager attention implementation. Document and query texts are encoded with a batch size of 512.

Stage 3: We use `Microsoft's Phi-4-14B GPTQ 4bit`[8] quantized variant, also run under the vLLM inference engine with weights cast to half precision. Inference parameters are set as: `temperature` = 0.9, `top_p` = 0.7, `top_k` = 1, and `max_tokens` = 2048. We batch four claims per inference run. Setting `top_k` to 1 enforces determinism, thereby ensuring reproducibility.

5.5 Constraints

The participating systems were required to comply with the following conditions:

1. Avoid the usage of proprietary LLMs.
2. Process each claim in under 60 seconds on average. The evaluation system was equipped with an NVIDIA A10G GPU (23 GB VRAM), 8 vCPUs, 32GB RAM, and 450GB file-system.
3. Capture the source of the article of each evidence used in fact-checking to facilitate manual auditing.

5.6 Challenge Results

Table 1 presents the top three submissions from the 2025 AVeriTeC challenge leaderboard (excluding the baseline), ranked according to their AVeriTeC scores on the test set. The Sanctuary system (codename **yellow_flash**) secured third place, achieving an AVeriTeC score approximately 5% higher than the baseline, with an average execution time of 31.71 seconds per claim.

On the development set, our system achieved scores of 0.2454 (Q-only), 0.5152 (Q+A), and 0.376 (AVeriTeC score), placing third on the dev leaderboard. However, a direct comparison with other systems on the development set is limited, as these evaluations were not conducted in a uniform, time-controlled environment.

However, the development and experimentation of our system was conducted on a different machine with a less powerful GPU compared to the challenge environment. Specifically, it featured an Intel(R) Xeon(R) Platinum 8253 CPU @ 2.20GHz (32 cores), an Nvidia Quadro RTX 8000 Turing GPU (48 GB VRAM), and 32 GB of RAM. On this setup, our system processed the 500 claims in the development set with an average execution time of approximately 55 seconds per claim.

6 Analysis

To assess the impact of backbone language models on overall system performance, we conducted an ablation study comparing our default LLM configuration – Qwen2.5-7B-Instruct for HyDE query generation and Phi-4 GPTQ for final reasoning – with a variant that uses Google's Gemini Flash 2.5 (Preview 04-17) (Team et al., 2023) in both stages 1 and 3. The Gemini model was constrained to a 1024-token "thinking" budget. All other pipeline

[7] https://huggingface.co/Snowflake/snowflake-arctic-embed-m-v2.0

[8] https://huggingface.co/jakiAJK/microsoft-phi-4_GPTQ-int4

LLM Combination (Stages 1 and 3)	Accuracy	Macro F1	Supported	Refuted	NEE	CE/C
Qwen2.5-7B and Phi-4 GPTQ	**0.535**	**0.486**	**0.649**	0.645	0.222	**0.429**
Gemini 2.5 Flash (Both Stages)	0.507	0.457	0.598	**0.655**	**0.351**	0.224
Gemini 2.5 Flash and Phi-4 GPTQ	0.490	0.438	0.586	0.609	0.223	0.333
Qwen2.5-7B and Gemini 2.5 Flash	0.466	0.421	0.642	0.555	0.278	0.207

Table 3: Local Classification Performance comparing our default combination of Qwen2.5-7B and Phi-4 GPTQ vs. Gemini Flash 2.5 on the same 100 development claims. Reported metrics include accuracy, macro F1, and per-class F1 scores.

components, parameters and prompts were kept fixed to ensure a controlled comparison.

For this experiment, we sampled 100 claims from the development set while ensuring an equal representation of each veracity class to reduce bias in performance estimation.

As shown in Table 2, substituting with Gemini resulted in notable gains across all Ev2R metrics. Specifically, we observed improvements of +5.77 points in Q-only recall, +9.39 points in Q+A recall, and a +4 point increase in the AVeriTeC score. These gains suggest that a stronger reasoning model enhances both the quality of generated questions and the utility of retrieved evidence, ultimately leading to better veracity predictions.

However, Table 3 presents a more nuanced picture. Despite Gemini's higher recall, our original configuration achieves a slightly higher classification accuracy (+2.8%) and macro F1 (+2.9%). It particularly excels in the *Supported* and *CE/C* classes, while Gemini notably performs better on the *NEE* class – indicating a more conservative stance when evidence is ambiguous or lacking.

These findings highlight an important trade-off: while Gemini – leveraging its advanced reasoning capacity and thinking mode – significantly improves factual recall and alignment with the Ev2R metrics, our Qwen–Phi pipeline delivers more balanced veracity classification despite scoring lower on the official metrics and operates under far lighter computational demands.

7 Conclusion and Future Work

We introduced SANCTUARY, an open-source time-efficient fact-checking pipeline that keeps model and hardware footprints modest while still closing much of the performance gap to heavier proprietary systems. By coupling lightweight question generation, a coarse-to-fine hybrid retriever, and quantized reasoning, the system attains a **new AVeriTeC** score of 25.27, an absolute 5–point gain over the shared-task baseline, yet verifies each claim in 31.71s on a single A10 GPU, 1.1s faster than the baseline.

Ablation results confirm that *reasoning capacity drives factual recall*: swapping in Gemini 2.5 Flash, a more capable LLM, lifts Ev2R Q-only, Q + A, and overall scores, at the cost of higher resources, yet achieves a slightly lower macro-F1 compared to our pipeline. Crucially, the study also exposes our main bottleneck, **question generation**. *Phi-4* still delivers strong label accuracy and reasoning when fed questions generated by either Qwen or Gemini; in fact, it scores better with Qwen-generated questions, despite their lower recall. This suggests that better-formed sub-questions could unlock further gains without enlarging the downstream reasoning model.

Our future work will therefore focus on increased knowledge-aware Q-generation and adaptive retrieval windows, in order to push recall higher while ensuring the same lean footprint that makes SANCTUARY efficient. We release our code and prompts to facilitate further research and reproducibility[9]

Limitations

Despite its efficiency and strong performance, Sanctuary has three main constraints. First, our HyDE Question Generation stage can miss multi-hop subquestions – key background links (e.g., Company X → Subsidiary Y) or tertiary events may go unexplored, capping recall. Second, Ev2R may penalize correct evidence not covered in the gold references, which means that valid but unannotated sources are treated as "misses"; this issue warrants further investigation. Third, our retrieval budget is fixed; complex claims or a larger evidence corpus may require adaptive context windows to avoid dropping crucial passages.

[9] https://github.com/arbaaz-abz/Sanctuary

Finally, we did not adopt the latest open-source LLMs at the time, such as Qwen3 (Yang et al., 2025), which many of the participants used. It is very likely that swapping in a more capable model, evidenced by our experiments with Google's Gemini, while still respecting our 60 s/claim budget, could yield even higher Ev2R recall and overall AVeriTeC scores.

Acknowledgment

This research is supported by NSERC RGPIN-2025-04608 and DGECR-2025-00153.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mubashara Akhtar, Michael Schlichtkrull, and Andreas Vlachos. 2024. Ev2r: Evaluating evidence retrieval in automated fact-checking. *arXiv preprint arXiv:2411.05375*.

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate post-training compression for generative pretrained transformers. *arXiv preprint arXiv:2210.17323*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Hadas Gold and Brian Stelter. 2025. Fake news x post caused market whiplash.

Greg Kamradt. 2024. The 5 levels of text splitting for retrieval. https://www.youtube.com/watch?v=8OJC21T2SL4.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and 1 others. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Stephan Lewandowsky, John Cook, Ullrich Ecker, Dolores Albarracin, Michelle A Amazeen, Panayiota Kendou, Doug Lombardi, Eryn Newman, Gordon Pennycook, Ethan Porter, and 1 others. 2020. *The debunking handbook 2020*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Christopher Malon. 2024. Multi-hop evidence pursuit meets the web: Team papelo at fever 2024. *arXiv preprint arXiv:2411.05762*.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. *arXiv preprint arXiv:2103.07769*.

Iryna Gurevych Nils Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *https://arxiv.org/abs/1908.10084*.

OpenAI. 2025. What are tokens and how to count them? Accessed: 2025-06-22.

Heesoo Park, Dongjun Lee, Jaehyuk Kim, ChoongWon Park, and Changhwa Park. 2024. Dunamu-ml's submissions on averitec shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 71–76.

Mark Rothermel, Tobias Braun, Marcus Rohrbach, and Anna Rohrbach. 2024. Infact: A strong baseline for automated fact-checking. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos. 2024. The automated verification of textual claims (AVeriTeC) shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36:65128–65167.

Craig Silverman. 2014. Verification handbook. a definitive guide to verifying digital content for emergency coverage. european journalism centre. open access publication.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv e-prints*, pages arXiv–2010.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. *arXiv preprint arXiv:1806.07687*.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, pages 58–65.

Herbert Ullrich, Tomáš Mlynář, and Jan Drchal. 2024. Aic ctu system at averitec: Re-framing automated fact-checking as a simple rag task. *arXiv preprint arXiv:2410.11446*.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yejun Yoon, Jaeyoon Jung, Seunghyun Yoon, and Kunwoo Park. 2024. Hero at averitec: The herd of open large language models for verifying real-world claims. *arXiv preprint arXiv:2410.12377*.

Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed 2.0: Multilingual retrieval without compromise. *arXiv preprint arXiv:2412.04506*.

Appendix A: Stage 1 prompt

You are a highly capable, thoughtful, and precise fact-checker.

Given a claim and its metadata, first classify the claim to the best of your knowledge and then generate relevant Question and Answer pairs. You must assign one of the following labels:

* **Supported:** The claim is fully backed by clear, consistent evidence.

* **Refuted:** The claim is directly contradicted by reliable evidence, or there is no concrete evidence to support the claim.

* **Conflicting Evidence/Cherrypicking:** The evidence has both supporting and opposing arguments for the claim or is selectively presented to favor the claim.

Guidelines:

– Generate anywhere between 5 to 10 Q-A pairs. Generate more questions for complex and multi-fact claims.

– Ensure questions have context. For example, given the claim *"Every week, at least 12 doctors leave Nigeria to seek employment in the UK."*, a better question is *"When was the claim made regarding Nigerian doctors moving to the UK to seek employment?"* rather than the overly generic *"When was the claim made?"*

– A question can have multiple sub-answers derived from various sources.

– Keep your answers brief (not more than 3 sentences).

– Fabricate random facts, figures, statements, or arguments, if and when needed.

Output Format:

```
"label": "Supported | Refuted | Conflicting Evidence/Cherrypicking",
"evidence": {Question-1: [Answer-1, Answer-2, ...], Question-2: [Answer-1], ...}
```

1-SHOT EXAMPLE (omitted for brevity)

Now process this claim:

```
{Claim}
{Claim Date}
{Claim Speaker}
{Location ISO Code}
{Reporting Source}
```

Figure 3: Prompt template for Hypothetical Question and Passage Generation.

Appendix B: Stage 3 prompt

You are a highly capable, thoughtful, and precise fact-checker.

Given a claim, its metadata, question and evidence pairs (each question can be addressed by several related evidences), your goal is to analyze the claim and how the evidence aligns with it. Then, you must label the claim using one of the following:

* **Supported**: The claim is fully backed by clear, consistent evidence with no significant contradictions.
* **Refuted**: The claim is directly contradicted by reliable evidence, or there is no evidence to support the claim.
* **Not Enough Evidence**: The evidence is insufficient to either support or refute the claim.
* **Conflicting Evidence/Cherrypicking**: The evidence both supports and opposes the claim or is selectively presented to favor the claim.

Guidelines:

– Take note of the claim date, the time period it refers to, speaker identity, geographical location, and reporting source for contextualization.

– Evaluate evidence within the relevant timeline and location constraints.

– Consider trustworthiness and title/URL source of evidence.

– For numerical claims, focus on data points; for events, check occurrence and timing; for position statements, consider the speaker's intent.

4-SHOT EXAMPLES (omitted for brevity)

Fact-check this claim:

```
{Claim}
{Claim Date}
{Claim Speaker}
{Location ISO Code}
{Reporting Source}
{Queries and Evidence}
```

You must carefully reason step-by-step using the context and evidence to determine the final label of the claim.

OUTPUT FORMAT:

```
Reasoning:
1. <concise rewrite of claim and intent; Note the timeline, location, statistics, numbers, quotes, events>
2. <evidence assessment>
3. <contradictions / gaps / biases noted>
4. <why the balance of evidence leads to the chosen label>

Label:
<Supported | Refuted | Not Enough Evidence | Conflicting Evidence/Cherrypicking>
```

Figure 4: Prompt template for veracity prediction.