Efficient and Faithful Claim Verification with Retrieval-Augmented LLMs

Adam Kuca

EECS 595, Computer Science and Engineering, University of Michigan

Motivation: Why Efficiency Matters

Retrieval-Augmented Generation (RAG) is the default way to ground LLMs in external knowledge, but it can be expensive:

- **Time to First Token (TTFT)** grows *quadratically* with context length.
- **Memory / KV cache** grows *linearly* with the number and length of retrieved documents.

In real-time or enterprise settings, even small latency reductions yield noticeably better throughput and lower cost. The challenge is to accelerate inference *without* sacrificing grounding quality.

Two most common strategies and their trade-offs are:

- **Full context (no filtering)** High recall, but increased latency as context dominates the prompt.
- **Top-k document filtering** Faster and cheaper, but might filter out important context.

Recently, Meta introduced **REFRAG**, a RAG framework that attends to document embeddings directly and selectively expands to text (via RL), achieving over $30\times$ faster TTFT in some settings.

Goal of this work: Investigate the **efficiency–faithfulness trade-off** in an applied RAG setting (claim verification), and understand how REFRAG compares to simpler RAG approaches.

Background & Task: Evidence-Based Claim Verification

Task: Given a claim c and retrieved sentences $\mathcal{P} = \{s_1, \ldots, s_n\}$, the model must:

1. Predict a label: $y \in \{\text{Supported, Refuted, Not Enough Info (NEI)}\}$,
2. Select evidence sentences that justify the prediction.

Claim verification is uniquely suited for studying efficiency-faithfulness trade-offs as even small changes in retrieval quality have a direct impact on label correctness and explanation faithfulness.

Related work:

- **Long-context Transformers** (e.g., Longformer, CEPE, Lingua): Reduce cost with sparse attention or prompt compression
- **REFRAG**: Attends over document embeddings instead of raw text, and selectively expands only the most relevant documents during decoding.

Dataset & Experimental Setup (FEVER)

Dataset: FEVER (Fact Extraction and VERification), **Paper Development Split**:

- 9,999 claims with **balanced labels** (3,333 each).
- On average, **1.10 ground truth evidence sentences** per claim.
- For each claim, we inject **10 random noise sentences** to simulate retrieval noise.

Model:

- Qwen3-4B, served via vLLM.
- All methods use the same LLM and prompting setup; only evidence selection changes.

Methods & Evidence Selection Strategies

Baselines

- **Random**: random label + random subset of sentences.
- **Majority + Keyword**: always predict the majority class; evidence selected by keyword search.

RAG-based methods

- **Naive Full Context**: use all sentences; no filtering.
- **Keyword Filtering (Top-k)**: rank by token overlap with the claim.
- **Semantic Filtering (Top-k)**: rank by cosine similarity between dense embeddings:

$$f_{\text{sem}}(c, s_i) = \cos(\mathbf{E}(c), \mathbf{E}(s_i)) = \frac{\mathbf{E}(c) \cdot \mathbf{E}(s_i)}{\|\mathbf{E}(c)\| \, \|\mathbf{E}(s_i)\|}.$$

Visualizing the Efficiency–Faithfulness Trade-off

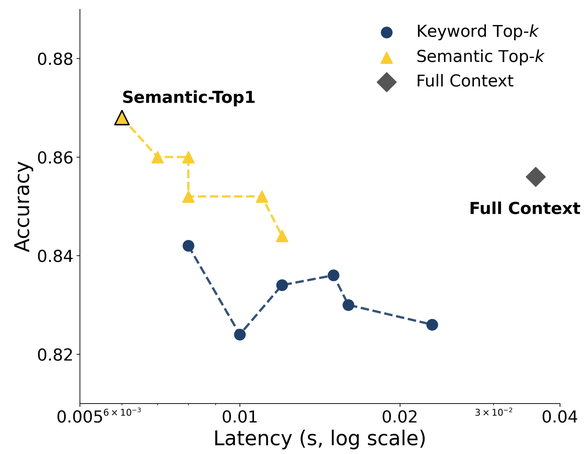


Figure 1. Efficiency–faithfulness frontier for different methods.

Trend: better retrieval corresponds to more efficient inference. Semantic (Top-1) achieves the best accuracy with the lowest latency..

Quantitative Results

Method	Acc	Prec	Recall	Latency (s)
Random Baseline	0.306	0.127	0.164	≈0.00
Majority + Keyword	0.322	0.553	0.862	0.005
Keyword (Top-1)	0.836	0.914	0.775	0.015
Keyword (Top-2)	0.842	0.909	0.809	0.008
Keyword (Top-4)	0.834	0.851	0.779	0.012
Semantic (Top-1)	**0.868**	**0.973**	0.827	0.006
Semantic (Top-2)	0.860	0.942	**0.869**	0.007
Semantic (Top-4)	0.852	0.905	0.858	0.008
Full Context	0.856	0.899	0.837	0.036

- **Semantic (Top-1) achieves the highest accuracy** while keeping latency 1/6 of full context.
- **Adding more sentences hurts both methods**. The average number of evidence sentences in the dataset is 1.10; therefore, anything above that is just noise.
- **Keyword filtering is too weak**; in this setting full context is always better.
- **Latency increases with context length**. The more sentences, the slower decoding becomes.
- **Full Context is better than expected**. Even though it is the slowest, it still performs competitively.

Take-Home Message

- The **efficiency–faithfulness trade-off** in RAG is determined not just by model size or context compression but also largely by retrieval quality.
- **Semantic-Top1** strategy can **outperform full context** on FEVER while being $\approx 6\times$ faster.
- **Keyword-only filtering is not strong enough**: if you cannot do good semantic retrieval, you will be better off using the full context.

RAG efficiency is not just about compressing context; it's about retrieving the right context.

Why Keyword Filtering Fails: Qualitative Analysis

Keyword search uses exact token matches, which cannot handle synonyms, implicit relationships, and phrasing differences.

Example 1

Claim: *"Savages was exclusively a German film."*
Gold label: Refuted

- **Keyword search:** misses the key evidence (no token overlap between "German film" and "American crime thriller") and predicts *NEI*.
- **Semantic search:** retrieves *"Savages is a 2012* ***American*** *crime thriller film..."* and correctly predicts *Refuted*.

Example 2

Claim: *"The series finale of Make It or Break It ended in 2013."*
Gold label: Refuted

- **Keyword search:** fails to connect the claim year to the evidence year.
- **Semantic search:** retrieves *"The series finale aired on May 14,* ***2012****..."* and correctly predicts *Refuted*.

Semantic filtering works better here because it incorporates **meaning**, not just keyword matching.

Retrieval Quality

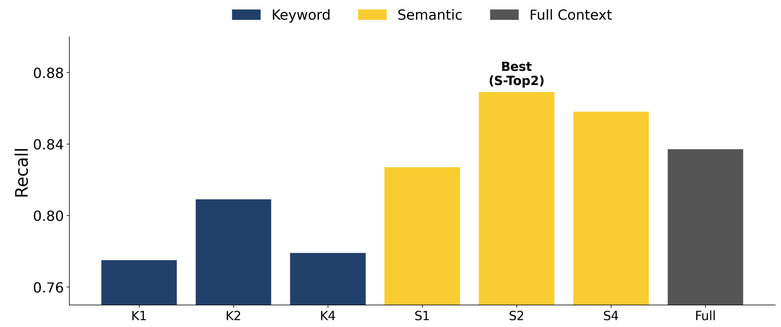


Figure 2. Semantic retrieval consistently captures more gold evidence than keyword search.

- **Semantic retrieval consistently selects more ground truth evidence** across all budgets.
- **Semantic (Top-2) achieves best recall**, surpassing full context with a fraction of sentences.
- Keyword recall is noticeably lower and less stable (sensitivity to synonyms and phrasing).
- Semantic (Top-2) outperforms Semantic (Top-1) in recall, which shows that Top-1 is more selective and precise, whereas Top-2 captures more evidence at the cost of some noise.

Future Work: Toward REFRAG

Semantic filtering is strong but still has limitations. Specifically, it can **miss important evidence** and generally requires **fixed** k.

REFRAG addresses these shortcomings by attending over **all document embeddings** (cheap), not full text and **selectively expanding** only the most relevant documents into text during decoding.

Next steps in this project:

- Implement an approximation of REFRAG in this FEVER setup.
- Compare its efficiency–faithfulness frontier to full-context, filtering methods, and baselines.
- Test the hypothesis that REFRAG can match or exceed Semantic (Top-1) accuracy with only a small latency degradation.

adamkuca@umich.edu