Efficient and Faithful Claim Verification with Retrieval-Augmented LLMs

Adam Kuca
6377 6692
Code: https://github.com/GSXRcoder159/CSE595-FinalProject/tree/main

Abstract

Retrieval-augmented generation (RAG) enables large language models to ground their predictions in external evidence, but this comes at a high cost in latency and memory as retrieved context grows. Recent work, such as REFRAG, addresses this inefficiency through context compression and adaptive expansion, yet the impact of these performance-oriented methods on explanation quality is still largely unexplored. In this work, we study explainability–efficiency tradeoffs in RAG-based factual claim verification, where models must both predict a label and justify it with evidence.

Using the FEVER Paper Development Dataset, we evaluate several context selection strategies, including full-context generation, keyword- and semantic-based Top$k$ filtering, and a REFRAG-inspired adaptive selection method implemented with a lightweight logistic regression classifier. We evaluate each approach in terms of classification accuracy, evidence precision and recall, inference latency, and context size.

Our results show that semantic Top$k$ filtering improves both efficiency and explanation faithfulness compared to full-context decoding, but its performance is heavily dependent on the choice of k. Adaptive context selection effectively addresses this limitation by dynamically adjusting the number of selected sentences per claim. The adaptive method consistently outperforms fixed Top$k$ strategies and the full-context baseline across a wide range of thresholds, achieving higher accuracy and stronger evidence recall with fewer context sentences and lower latency.

Additionally, this work shows that even a simple approximation of REFRAG-style adaptive expansion can meaningfully improve the explainability—efficiency trade-off in RAG systems and can speed up and improve the trustworthiness of retrieval-augmented claim verification.

1 Introduction

Large Language Models (LLMs) with retrieval augmented generation (RAG) can answer questions grounded in external documentation, but this comes at a significant cost in latency and memory usage when the prompt context grows long. In particular, time-to-first-token (TTFT) grows quadratically with context length (Lin et al., 2025), and retrieving large context also requires a large key-value cache, which limits throughput. Recent work by Lin et al. (2025) introduced REFRAG (REpresentation For RAG), a new decoding framework to address this inefficiency. REFRAG compresses retrieved passages into compact embeddings and selectively expands them during decoding only when needed, reporting large improvements in TTFT and effective context length without accuracy loss.

However, how this selective context pruning affects the model's ability to explain its decisions is still unexplored. Explainability is crucial for user trust and transparency, and studying it in the context of REFRAG and similar performance-oriented approaches is essential to ensure that efficiency gains do not come at a cost of model interpretability.

The goal of this project is to investigate explainability–efficiency tradeoffs in an applied RAG setting, using REFRAG as a conceptual motivation. Specifically, the project focuses on evidence-based factual claim verification with justification. It evaluates how different context selection strategies, including an approximation of REFRAG's adaptive expansion, affect both classification accuracy and the faithfulness and completeness of generated explanations.

The impact of this project extends beyond academic contributions and is likely to be valuable in enterprise RAG deployments, where models must be both fast and trustworthy and where cost is often as important as accuracy.

2 Data

The task studied in this project is evidence-based claim verification as defined by Thorne et al. (2018). The input consists of a claim (e.g., "*Moonlight's filming began in 2015.*") along with a set of retrieved candidate evidence extracts. The output is a label: **Supported**, **Refuted**, or **Not Enough Info**, along with an evidence-based justification. That is, the model must both decide the truthfulness of the claim and select evidence to justify its decision. Importantly, this task differs from claim verification, for example, as used in REFRAG, which only considers the classification label. Evidence-based claim verification was selected because it provides a structured approach to studying the explainability—efficiency trade-offs in RAG.

2.1 FEVER Dataset

Specifically, the project uses the FEVER dataset (Thorne et al., 2018). This dataset consists of claims, ground-truth labels, and evidence annotations. The claims were generated by altering sentences from Wikipedia (June 2017) and then manually verifying them. For evaluation, the `Paper Development Dataset` with 9,999 claims is used (Thorne et al., 2018).

Claims are labeled as SUPPORTED, REFUTED, or NOT ENOUGH INFO. For the SUPPORTED and REFUTED claims, the dataset also provides ground-truth evidence annotations identifying which Wikipedia sentences support or refute each claim. These annotations allow for evaluation of each approach's ability to justify its classification decisions. Each FEVER entry contains four fields: an ID, the claim text, a label, and one or more evidence sentences. Examples are shown in Table 1.

2.2 Wikipedia Corpus

The FEVER dataset is accompanied by a snapshot of Wikipedia. The Wikipedia pages were obtained from the official FEVER Dataset Website. While the dataset itself required no preprocessing, the Wikipedia pages were normalized, sentence-segmented, and stored as records in an Apache Parquet file for efficient storage and retrieval. Each record includes a page ID, the processed sentence, and metadata such as page title, sentence index, and word count.

To provide a clearer picture of the `Paper Development Dataset` used for evaluation, below

Claim	Label	Evidence (page : sentence id)
Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.	SUPPORTS	`Nikolaj_Coster-Waldau : 7`
Lorelai Gilmore's father is named Robert.	REFUTES	`Lorelai_Gilmore : 3`
Henri Christophe is recognized for building a palace in Milot.	NOT ENOUGH INFO	N/A

Table 1: Examples from the FEVER dataset.

are some basic statistics:

- **Instances:** 9,999
- **Label distribution:**
  - SUPPORTED: 3,333
  - REFUTED: 3,333
  - NOT ENOUGH INFO: 3,333
- **Average number of evidence sentences (across labels):** 1.10

For the pre-processed Wikipedia corpus:

- **Number of Sentences:** 24,127,945
- **Unique pages:** 5,391,813
- **Average sentence length (characters):** 115.5

2.3 Train-Evaluation Split for Context Selection Policy

To train the adaptive context selection policy (Section 4.4), a random 5% subset of the `Paper Development Dataset` was used as a training set. This subset was used only to train the logistic regression classifier that predicts which sentence should be kept as evidence. The remaining 95% of the data was used for the final evaluation of all methods, including the baselines and the LLM-based approaches.

3 Related Work

This project involves three main areas of NLP: fact verification, efficient long-context modeling, and explainability under performance optimizations.

Claim verification has been widely studied. The FEVER dataset (Thorne et al., 2018), used in this

project, introduced large-scale claim verification on Wikipedia data and helped popularize the task. Similarly, SciFact (Wadden et al., 2020) focused on verifying scientific claims in research abstracts.

There is also much work on allowing transformers to handle long inputs more efficiently. Methods like Longformer's windowed attention (Beltagy et al., 2020) use sparse attention patterns to achieve roughly linear scaling in sequence length. Other methods, such as CEPE (Yen et al., 2024) or LongLLMLingua (Jiang et al., 2024), compress or summarize parts of the prompt to reduce length. Although these approaches improve efficiency over a naive full-context baseline, they still require the model to attend to many (or most) tokens, limiting the efficiency gains. In contrast, REFRAG (Lin et al., 2025) encodes entire chunks into fixed-size embeddings and only expands them when needed via a lightweight policy. REFRAG takes advantage of the block-sparse attention pattern inherent in RAG (unrelated documents don't need to interact), allowing it to achieve much greater latency improvements (e.g., 30× faster TTFT on benchmark tasks).

In the area of explainability, Łajewska et al. (2025) recently recognized the problem of information loss under prompt compression and proposed a framework for its evaluation. While their work examines explainability (in terms of information loss) more broadly, this project focuses on its specific implications for RAG. Additionally, Tan et al. (2024) explores faithfulness and traceability and introduces a framework to help claim verification models reason more faithfully over scientific literature. Whereas Tan et al. (2024) study faithfulness and traceability in full-context settings, this project explores whether comparable levels of explainability can be achieved under performance optimizations such as text compression in REFRAG.

Overall, prior work has not examined the impact of efficient RAG methods and their performance optimizations on explanation quality in a systematic way. Existing work has explored explainability and faithfulness in other contexts, but not specifically in the context of efficient RAG. Furthermore, works such as Lin et al. (2025) achieve impressive performance gains but do not analyze their impact on explainability and faithfulness. This project aims to address this by evaluating REFRAG-inspired adaptive context selection, along with other approaches, in a claim verification setting where both performance and explanation quality are important.

4 Methods

To investigate the tradeoffs between explainability and efficiency, several approaches for claim verification were implemented and compared. All methods share the same basic setup but differ in how they select and use context.

4.1 Shared Task and Retrieval Setup

For each claim in the `Paper Development Dataset`, a candidate set of evidence sentences is created. In all the experiments, this candidate set consists of:

1. All ground-truth evidence sentences annotated for that claim, and
2. Additional 10 "noise" sentences sampled from the pre-processed Wikipedia corpus.

This setup simulates retrieval noise while guaranteeing that ground-truth sentences for SUPPORTED and REFUTED are present.

Originally, the retrieval involved embedding all relevant Wikipedia corpus sentences with the E5-base-v2 encoder and building a FAISS index, as this would be necessary for the full REFRAG pipeline (Lin et al., 2025)[1][2]. However, to simplify and avoid indexing overhead, all experiments reported here use the `all-MiniLM-L6-v2` Sentence-Transformer model to compute dense embeddings for both claims and candidate sentences[3]. Semantic similarity between these embeddings is calculated as their cosine similarity and used both for Semantic-Top$k$ filtering and as a feature in the adaptive selection policy.

4.2 Non-LLM Baselines

Two simple baselines were implemented and evaluated to obtain lower bounds on performance.

Random. The random classifier predicts one of the three FEVER labels uniformly at random and selects two sentences also uniformly at random from the candidate set as evidence.

Majority + Keyword. The Majority+Keyword baseline always predicts the majority class in the training data (here, it is equivalent to uniform random selection since classes are evenly distributed). For evidence selection, this baseline computes a

[1] https://github.com/facebookresearch/faiss
[2] https://huggingface.co/intfloat/e5-base-v2
[3] https://github.com/huggingface/sentence-transformers

simple keyword-overlap score between the claim and each candidate sentence and always selects the two sentences with the most overlap. This baseline provides a surprisingly strong lower bound on the evidence selection performance.

Unlike other methods, the baselines always select two sentences per claim.

4.3 LLM-Based Methods

All LLM-based methods use the same decoder model and only differ in how they select context sentences (evidence) for the prompt.

Decoder Model. For all experiments, Qwen3-4B with 8-bit quantization is used as the decoder model (Team, 2025)[4]. It is used via vLLM for efficient inference and key-value cache management (Documentation, 2023)[5]. The prompt contains the claim and all selected context sentences (number depends on Top$k$), along with a system prompt that instructs the model to decide whether the claim is SUPPORTS, REFUTES, or NOT ENOUGH INFO, and justify its decision by selecting the INDICES of the evidence sentences that support or refute the claim. For reliable output parsing vLLM's structured output with two fields, one for the label and one for the justification indices, is used.

Full Context. The full-context baseline performs no filtering and gives the decoder model all candidate sentences. It is the simplest, but also the most expensive method. This baseline represents the most common real-world approach to RAG, in which the full retrieved context is passed to the decoder.

Keyword-Top$k$. The Keyword-Top$k$ method ranks candidate sentences by keyword overlap with the claim, as in the Majority+Keyword baseline. The top k sentences are then selected as context for the decoder model and inserted into the prompt. Experiments are performed with $k \in \{1, 2, 3, 4, 6, 8\}$.

Semantic-Top$k$. In Semantic-Top$k$ filtering candidate sentences and the claim are embedded with the SentenceTransformer `all-MiniLM-L6-v2` model[6] and ranked by cosine similarity with the claim embedding. As in Keyword-Top$k$, filtering is performed by selecting the top k sentences and including them in the decoder prompt. Again, $k \in \{1, 2, 3, 4, 6, 8\}$ is used for experiments.

For all experiments, the decoder is fixed, allowing us to study the impact of context size (number of context sentences) and ranking strategy on explanation quality and efficiency.

4.4 REFRAG-Inspired Adaptive Selection

The last method is based on the REFRAG's adaptive expansion mechanism. To study this mechanism, an adaptive context selection method based on a logistic regression classifier is implemented. It is designed to be a meaningful approximation that is feasible to implement within the time frame of this project. As such, it does not perform reinforcement learning nor incorporate compressed embeddings.

Training Data and Labels. The logistic regression model is trained on examples from the 5% training split of the `Paper Development Dataset` (Section 2.3). For each claim-sentence pair, the label is defined as:

$$y = \begin{cases} 1 & \text{if ground-truth evidence,} \\ 0 & \text{otherwise.} \end{cases}$$

This creates a significantly imbalanced dataset with many evidence sentences.

Feature Design. For each claim-sentence pair, the following four features are used:

1. **Semantic score:** Cosine similarity between the SentenceTransformer embeddings of the claim and the sentence. This represents absolute semantic information.

2. **Keyword overlap count:** The number of overlapping non-stopword tokens between the claim and the sentence. This provides the model with lexical information.

3. **Normalized rank:** An inverse of each sentence's rank $r \in \{0, 1, 2, \dots\}$ defined as $1/(r+1)$. It assigns higher values to higher-ranked sentences. This feature is meant to provide relative semantic information.

4. **Length ratio:** The ratio between the sentence length and the length of the claim, in tokens. This feature is added to help the classifier leverage any structure based on relative length.

[4] https://huggingface.co/Qwen/Qwen3-4B
[5] https://github.com/vllm-project/vllm
[6] https://github.com/huggingface/sentence-transformers

These features were selected to provide as much information as possible while remaining simple and efficient to compute. As such, they do not rely on additional training beyond the Sentence-Transformer embeddings already used for semantic selection.

Model and Training. For simplicity, the classifier model is implemented using scikit-learn[7]. Specifically, a two-stage pipeline involving standardization and a logistic regression model is used:

```
make_pipeline(StandardScaler(),
    LogisticRegression(
    class_weight='balanced',
    solver='liblinear'))
```

The `StandardScaler` standardizes each feature to zero mean and unit variance, which is important because the features have different scales that could negatively influence optimization.

The `LogisticRegression` classifier uses `class_weight='balanced'` to account for the significant class imbalance in the data by automatically reweighting classes based on their frequencies. The `liblinear` solver is used mainly because it supports class weights.

Inference. During inference, for each claim in the evaluation split, the trained logistic regression model predicts a probability $p(\text{evidence} \mid \text{features})$ for each candidate sentence. To simulate REFRAG's selective expansion mechanism, a probability threshold T is used to select the sentences for the decoder's context. Unlike Top$k$ filtering, this mechanism selects a variable number of sentences per claim, same as in REFRAG (Lin et al., 2025).

Experiments are performed to study the impact of the probability threshold. Specifically, an ablation study is performed on this threshold with $\tau \in \{0.1, 0.2, \ldots, 0.9\}$. In practice, this policy selects an average of 2.2 sentences per claim (Section 5), which is much less than the full context.

The adaptive context selection method uses the same pipeline, decoder model, and prompt as the other LLM-based methods.

5 Evaluation and Results

All methods are evaluated on the 95% held-out evaluation part of the `Paper Development Dataset`. We evaluate each method on:

- **Classification Accuracy:** Accuracy of the predicted FEVER labels measured against ground-truth labels.
- **Evidence precision (faithfulness):** Whether the justification sentences selected by the model as evidence are correct:

$$\text{precision} = \frac{|\text{selected} \cap \text{ground-truth}|}{|\text{selected}|}.$$

  Higher precision corresponds to fewer hallucinations.
- **Evidence recall (completeness):** Whether the justification sentences selected by the model as evidence include all ground-truth evidence:

$$\text{recall} = \frac{|\text{selected} \cap \text{ground-truth}|}{|\text{ground-truth}|}.$$

  Higher recall corresponds to the model selecting more of the required evidence.
- **Inference Latency:** Average inference time of the LLM over all batches. For non-LLM baselines, latency is considered to be zero.
- **Average sentences:** For the adaptive method, the average number of sentences included in the prompt, otherwise constant.

5.1 Fixed-Context Methods

Table 2 summarizes the performance of the non-LLM baselines, the keyword- and semantic-based top-k methods, and the full-context baseline.

As expected, given the even class distribution, both non-LLM baselines achieve roughly 33% accuracy. The random baseline achieves poor precision and recall, roughly corresponding to the proportion of ground-truth sentences among the candidate sentences (≈1/11). The Majority+Keyword baseline, however, achieves a relatively high recall (0.862, indicating that simple keyword overlap is often enough to retrieve at least one correct sentence. Nevertheless, both baselines perform poorly in terms of accuracy and recall compared to LLM-based methods.

The Top$k$ methods show that filtering the context to the most relevant sentences leads to significant improvements in both accuracy and precision compared to the full context method. Both filtering methods also show a slight trade-off between precision and recall at Top1 and Top2 filtering.

[7] https://scikit-learn.org/stable/

Method	Accuracy	Precision	Recall	Latency (s)
Random	0.336	0.127	0.164	0.000
Majority+Key.	0.332	0.553	0.862	0.000
Keyword-Top1	0.836	0.914	0.775	0.015
Keyword-Top2	0.842	0.909	0.809	0.008
Keyword-Top3	0.824	0.856	0.780	0.010
Keyword-Top4	0.834	0.851	0.779	0.012
Keyword-Top6	0.830	0.854	0.799	0.016
Keyword-Top8	0.826	0.851	0.791	0.023
Semantic-Top1	0.868	0.973	0.827	0.006
Semantic-Top2	0.860	0.942	0.869	0.007
Semantic-Top3	0.860	0.930	0.870	0.008
Semantic-Top4	0.852	0.905	0.858	0.008
Semantic-Top6	0.852	0.902	0.850	0.011
Semantic-Top8	0.844	0.886	0.838	0.012
Full Context	0.856	0.899	0.837	0.036

Table 2: Performance of non-LLM baselines and top-k selection methods.

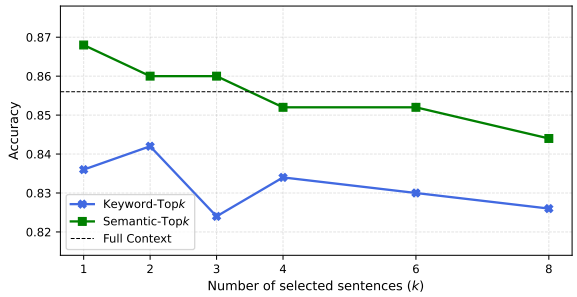


Figure 1: Accuracy as a function of the number of selected context sentences k for Keyword-Top$k$ and Semantic-Top$k$.

As visualized in Figure 1, the Semantic-Top$k$ method consistently outperforms the keyword-based method at the same k. Semantic-Top1 achieves the best overall accuracy (0.868) and precision (0.973), suggesting that selecting just one most semantically similar sentence is very often enough for correct and faithful claim verification. As k increases to 2 and 3, recall improves but at the cost of accuracy and precision. Importantly, all Semantic-Top$k$ methods achieve a lower latency than the full-context baseline.

Figure 1 and Table 2 also clearly show that full-context method always outperforms Keyword-Top$k$, but achieves lower accuracy and precision than Semantic-Top$k$ with $k \leq 3$ while having significantly higher latency. This suggests that simply adding more context does not necessarily improve classification performance or explanation quality, unless there is no effective way to filter retrieved context (e.g., keyword filtering).

5.2 Adaptive Context Selection

Table 3 shows results for the REFRAG-inspired adaptive selection policies implemented with logistic regression.

Method	Accuracy	Precision	Recall	Lat. (s)	Avg sents.
Random	0.312	0.130	0.169	0.000	2.000
Majority	0.324	0.560	0.863	0.000	2.000
Keyword-Top1	0.836	0.914	0.775	0.015	1.000
Keyword-Top2	0.842	0.909	0.809	0.008	2.000
Semantic-Top1	0.868	0.973	0.827	0.006	1.000
Semantic-Top2	0.860	0.942	0.869	0.007	2.000
REFRAG (τ=0.1)	0.872	0.908	0.846	0.024	4.896
REFRAG (τ=0.2)	0.870	0.928	0.868	0.008	3.070
REFRAG (τ=0.3)	0.880	0.955	0.889	0.008	2.384
REFRAG (τ=0.4)	0.874	0.959	0.892	0.007	1.970
REFRAG (τ=0.5)	0.878	0.964	0.897	0.007	1.754
REFRAG (τ=0.6)	0.878	0.967	0.898	0.007	1.576
REFRAG (τ=0.7)	0.882	0.972	0.901	0.007	1.468
REFRAG (τ=0.8)	0.880	0.972	0.897	0.006	1.376
REFRAG (τ=0.9)	0.878	0.975	0.893	0.006	1.300
Full Context	0.856	0.899	0.837	0.036	11.100

Table 3: Performance of REFRAG-inspired adaptive selection. "Avg sents." is the average number of sentences included in the LLM input.

REFRAG-inspired adaptive selection outperforms the full-context baseline across all evaluated thresholds on the held-out evaluation set. The adaptive context selection method achieves the highest accuracy and recall with $\tau = 0.7$ (0.882 and 0.091, respectively). Additionally, precision increases with the threshold, validating the effectiveness of the adaptive context selection. Inference latency remains low, as with Top$k$, since it depends on the number of retrieved sentences. As expected, the number of selected sentences decreases as the threshold increases.

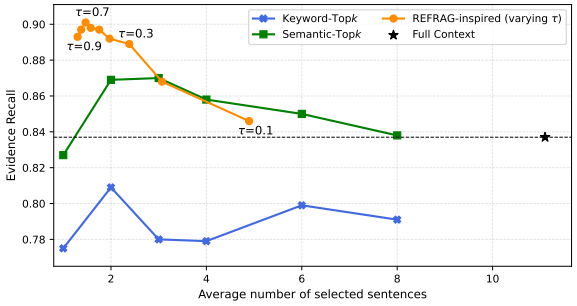


Figure 2: Explainability–efficiency tradeoff between evidence recall and average number of selected context sentences.

Compared to the best Semantic-Top$k$ method (Semantic-Top1), the adaptive $\tau = 0.7$ selection

improves recall (0.901 vs. 0.827). Furthermore, as visualized in Figure 2, the adaptive selection with $\tau \in [0.4, 0.9]$ achieves better recall than any Semantic-Top$k$ filtering (0.901 vs. 0.869) while selecting less context. This suggests that the logistic regression classifier successfully combines its semantic, keyword, rank, and length features into a more effective and fine-grained selection than fixed-k filtering. This performance also validates the generalization of the learned selection policy to unseen data, as it achieves an almost perfect training accuracy of 0.946 and a very strong performance on the validation data.

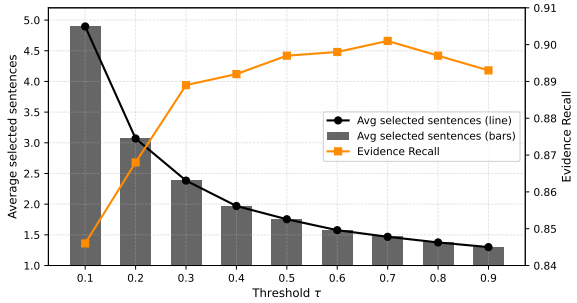


Figure 3: Sensitivity of the adaptive selection policy to the probability threshold τ.

Figure 2 also shows that while Top$k$ filtering methods are spread out with fixed intervals, the adaptive method smoothly interpolates over the continuous interval. Furthermore, with adaptive context selection, the points are clustered around the ideal context size while bad context sizes are avoided entirely. Figure 3 further illustrates this property of adaptive selection. As τ increases, the number of selected sentences decreases rapidly at first, but then much more slowly around the optimal context size. It also shows that the method is not overly sensitive to the exact choice of threshold and, in fact, achieves superior recall to any other method for $\tau \in [0.3, 0.9]$.

Overall, these results show that a simple feature-based classifier, trained on 5% of the development data, can approximate REFRAG-like adaptive expansion and yield a strictly better explainability-efficiency tradeoff than naive full-context decoding or semantic filtering.

6 Discussion

Based on the experimental results, we can make several observations about context size, explainability, and efficiency in RAG-style claim verification.

First, semantic similarity is an effective and reliable signal for evidence selection. The Semantic-Top$k$ method, by incorporating semantic meaning, consistently outperforms keyword-based filtering at the same context size in both accuracy and explainability (evidence precision and recall). This result aligns well with the common practice of using dense retrieval in RAG systems and confirms that even simple semantic filtering can improve both prediction quality and explanation faithfulness.

Second, more context is not always better. The full-context baseline, which naively gives the LLM all candidate sentences, is slower and achieves lower performance than the best semantic filtering approaches (Top1-Top3). This suggests that too much context can actually hurt explanation faithfulness by introducing irrelevant or distracting information. In other words, explanation quality depends less on the quantity of available evidence and more on the ability to reliably select the most relevant context.

Third, while semantic Top$k$ filtering performs well, choosing an appropriate value of k remains non-trivial. As shown in the results, only a small range of k values achieves strong performance, and increasing k outside this range results in inferior performance relative to the full-context baseline. This sensitivity motivates adaptive context selection that can dynamically adjust the amount of context rather than relying on a fixed heuristic.

The REFRAG-inspired adaptive context selection demonstrates that many of the benefits of more sophisticated frameworks like REFRAG can be achieved with a surprisingly simple approach. By training a logistic regression classifier on four simple, cheap features, the model learns to effectively select context and consistently outperforms even the already strong semantic filtering. The adaptive context selection policy achieves higher accuracy and better explainability scores than any other approach, while using fewer sentences and having lower latency. This result is somewhat surprising, given the popularity of dense semantic retrieval for evidence selection. One possible explanation is that the ability to dynamically adjust the number of selected sentences allows the model to avoid introducing noise from unnecessary context while ensuring all the relevant context is present. Additionally, for fact verification, there may be useful structure in simple features such as keyword overlap, sentence length, and rank that dense similarity does not make use of.

Finally, the fact that these improvements are achieved without any context compression, encoder-decoder alignment, or reinforcement learning is promising for the applicability of such a method. Many real-world RAG systems lack the capacity to retrain embeddings or fine-tune large language models, but they can often add a lightweight classifier with off-the-shelf embeddings. The experimental results suggest that something as simple as a basic logistic regression classifier with four simple, cheap features can meaningfully improve both efficiency and explainability in RAG-based systems.

7 Conclusion

This project explored explainability-efficiency tradeoffs in RAG-based factual claim verification. Although the original goal of implementing the full REFRAG framework could not be completed within the scope of this project, focusing on adaptive context selection was both feasible and surprisingly informative.

Using the FEVER `Paper Development Dataset`, the following methods were compared: (i) non-LLM baselines, (ii) an LLM-based full-context approach, (iii) keyword- and semantic-based Top$k$ filtering, and (iv) a REFRAG-inspired adaptive selection with logistic regression. All methods were evaluated on classification accuracy, evidence precision and recall, inference latency, and the number of context sentences.

The key findings are:

- Semantic retrieval (using SentenceTransformer `all-MiniLM-L6-v2`) consistently outperforms keyword-based retrieval on all metrics at the same k.
- Using all retrieved context is neither necessary nor optimal. Semantic-Top[1,2,3] all achieve better accuracy and higher evidence precision than the full-context baseline with significantly lower latency.
- A simple logistic regression model, trained on just 5% of the development data using four cheap features, can successfully perform adaptive context selection that outperforms all other methods on both effectiveness and explainability, while using fewer sentences and achieving lower latency.

These results show that while semantic filtering already provides clear benefits over full-context decoding, selecting the right fixed context size remains challenging. Only a small subset of k values consistently improves over the full-context approach, making fixed-k strategies hard to use in practice.

In contrast, adaptive context selection effectively addresses this limitation. By dynamically adjusting the number of context sentences based on a probability threshold, the adaptive method outperforms the full-context baseline across **all** evaluated thresholds and metrics, including accuracy, explainability, and latency. Moreover, it is substantially less sensitive to hyperparameter choice than Semantic-Top$k$ filtering and achieves a strictly better explainability–efficiency tradeoff across most thresholds.

Overall, the results suggest that a lightweight selection model on top of off-the-shelf embeddings can achieve a large portion of the benefits of the full REFRAG framework. While fully implementing REFRAG would require significantly more architectural complexity, data, and compute, even a simple approximation, such as the one evaluated in this project, can already provide tangible practical improvements in RAG-based claim verification systems.

8 Other Things We Tried

Several additional parts were implemented or explored during the project, but were not used in the final experiments.

FAISS Index and E5-base-v2. The initial plan for retrieval was to embed the entire preprocessed Wikipedia corpus using the pre-trained E5-base-v2 model and store the embeddings in a FAISS index for fast approximate nearest-neighbor search. A prototype of this approach was implemented and tested. However, testing revealed that adding all 24 million sentences to FAISS was infeasible, as index construction and updates became too slow after approximately 3 million sentences.

Further analysis showed that only 3,493 unique sentences (0.014477% of the Wikipedia corpus) are used as ground-truth evidence in the `Paper Development Dataset`. This suggested that the full corpus indexing was unnecessary for the project. However, further testing revealed that the FAISS index pipeline with the pre-trained E5-base-v2 encoder was unnecessarily complex for the scope of this project. As a result, the pipeline was abandoned, and SentenceTransformer `all-MiniLM-L6-v2` was used directly to create em-

beddings for only the candidate sentences needed per claim.

Continual Pre-Training and Encoder-Decoder Alignment. The REFRAG paper uses continual pre-training (CPT) to align the encoder and decoder representations and enable the use of embedding from the RAG encoder as a form of compressed context. The initial plan was to perform next-paragraph prediction on Wikipedia to align the chosen encoder (E5-base-v2) with the decoder (Qwen3-4B). This approach was also partially implemented and tested. However, due to time constraints and instability of the continual pre-training, the project was simplified to avoid performing CPT.

9 What You Would Have Done Differently or Next

All things considered, the project went quite smoothly. However, there are a few things that I would have done a bit differently if I were to start over. Most importantly, I would spend more time testing the different parts of the project before starting to work on a solution. Many of the things that were tried and ultimately abandoned could have been avoided by performing simple feasibility tests or by doing more research into the technology before starting implementation. Specifically, both the issues with the FAISS index and the CPT could have been spotted much earlier. The other option would be to start small and grow the project incrementally. By first looking for the simplest possible solution to a problem, the project implementation and execution could have been simplified. By starting with a complex, broad implementation, a lot of unnecessary work was done implementing features that were never actually used.

As for future work, the project has several limitations that would make for interesting extensions.

Full REFRAG Implementation. Most importantly, the project does not implement the full REFRAG architecture. There is no representation-level compression of context, no encoder-decoder aliment via CPT, and no reinforcement learning expansion policy. Instead, the project focuses on a relatively simple adaptive selection policy applied to uncompressed sentences. Future work should implement at least a small-scale version of REFRAG to understand the benefit of representation-level compression.

Limited Retrieval Setup. The current experiments assume that the candidate set already contains the ground-truth evidence sentences plus sampled noise. This simplifies the problem to just context selection but does not fully capture the challenges of end-to-end retrieval. Future work should investigate a realistic retrieval and evaluate how retrieval quality improves with adaptive context selection.

Limited Models and Hyperparameters. All LLM-based experiments were performed with a single decoder model (Qwen3-4B) and two hyperparameters (k and the probability threshold τ). Future work could extend this study further to:

- Explore different features and their combinations for the adaptive policy, including more complex features like attention scores.
- Evaluate larger or smaller LLMs to understand how model capacity affects the context selection quality.

Single Task. Finally, all experiments were performed on the claim verification task using the FEVER `Paper Development Dataset`. While this dataset was well-suited to studying the efficiency-faithfulness trade-offs, it constrains the results to a single task. It is not clear how well the results generalize to other tasks, such as multi-hop reasoning or long-document QA. Extending the current method to other tasks would therefore be an interesting area for future work.

Despite these limitations, the project's results show that even simple adaptive context selection can improve the explainability–efficiency trade-off in claim verification RAG systems, and that many of the benefits of the more complex REFRAG framework can be achieved without significant additional training.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Qwen Documentation. 2023. Qwen deployment with vllm. Accessed: 2025-11-06.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 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), pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Weronika Łajewska, Momchil Hardalov, Laura Aina, Neha Anna John, Hang Su, and Lluis Marquez. 2025. Understanding and improving information preservation in prompt compression for LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17520–17541, Suzhou, China. Association for Computational Linguistics.

Xiaoqiang Lin, Aritra Ghosh, Bryan Kian Hsiang Low, Anshumali Shrivastava, and Vijai Mohan. 2025. Refrag: Rethinking rag based decoding. *Preprint*, arXiv:2509.01092.

Neset Özkan Tan, Niket Tandon, David Wadden, Oyvind Tafjord, Mark Gahegan, and Michael Witbrock. 2024. Faithful reasoning over scientific claims. In *Proceedings of the AAAI 2024 Spring Symposium Series, Stanford, CA, USA, March 25-27, 2024*, pages 263–272. AAAI Press.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-context language modeling with parallel context encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2588–2610, Bangkok, Thailand. Association for Computational Linguistics.