Efficient Token Optimization

in Retrieval-Augmented Generation Pipelines for HPC

Iowa State University · CS 6250 Project

Motivation & Problem Statement

Why does token efficiency matter?

Token Bloat

Baseline RAG passes entire retrieved chunks to the LLM. Most content is irrelevant to the query yet still counted.

API Cost at Scale

HPC workloads issue thousands of queries. At $0.30 per 918 questions, costs compound quickly in production.

No Context Filtering

Standard pipelines lack query-aware compression — identical context is sent regardless of question specificity.

Proposed Method

Query-aware summarization before LLM generation

Two-Stage Context Compression

Keyword-based Filtering

Select top-k sentences from retrieved docs with highest query-word overlap. If no overlap, use first k sentences.

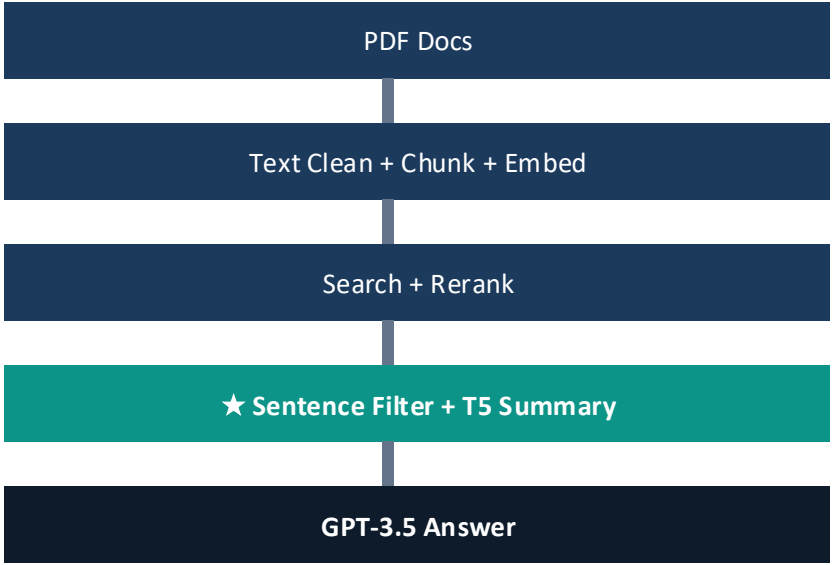
LM Summarization

A local T5 model (Small/Base/Large) condenses filtered sentences into a query-aware summary.

Compressed Prompt

Final prompt = query + summary. Sent to GPT-3.5 instead of full retrieved text.

Pipeline Flow



Experiment Setup

rag-mini-wikipedia dataset · 918 QA pairs

Dataset

rag-mini-wikipedia

918 questions · Wikipedia passages

Retrieval

FAISS + Cross-Encoder

Dense retrieval + re-ranking

Generator

GPT-3.5-Turbo

Answer generation via OpenAI API

Summarizers

T5 Small / Base / Large

3 local model sizes tested

Evaluation Metrics: Exact Match · F1 Score · BERTScore · ROUGE-1/L · Semantic Similarity · Token Count · Estimated Cost

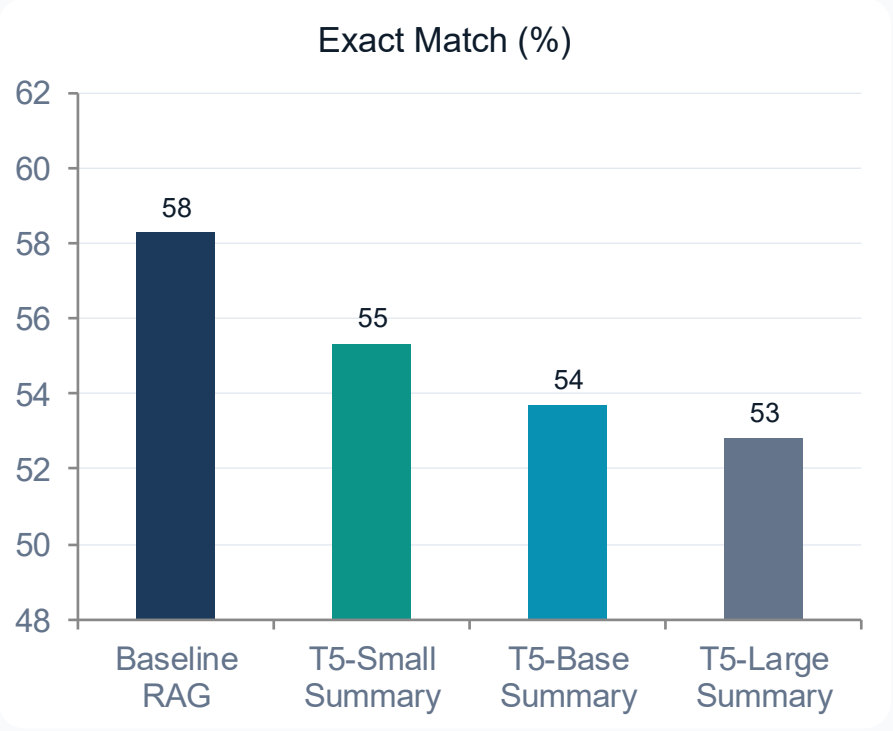
Results: Overall Performance

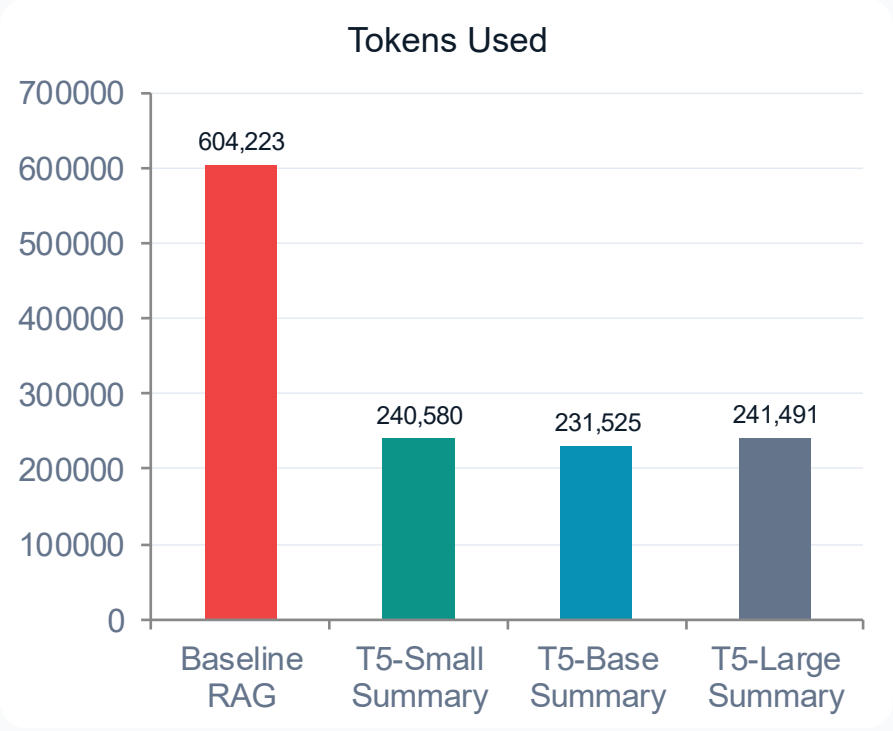
Evaluated on 918 QA pairs from rag-mini-wikipedia

Model	Exact Match	F1 Score	BERTScore	Tokens Used	Est. Cost
Baseline RAG	**58.28% ★**	**0.704 ★**	**0.9474 ★**	604,223	
T5-Small Summary ⭐	55.34%	0.640	0.9389	**240,580 ↓60%**	
T5-Base Summary	53.70%	0.633	0.9397	**231,525 ↓62%**	
T5-Large Summary	52.83%	0.614	0.9360	241,491	

★ *Best in category* · ⭐ *Recommended* · ↓ *Reduction vs Baseline*

Accuracy vs. Token Efficiency Trade-off

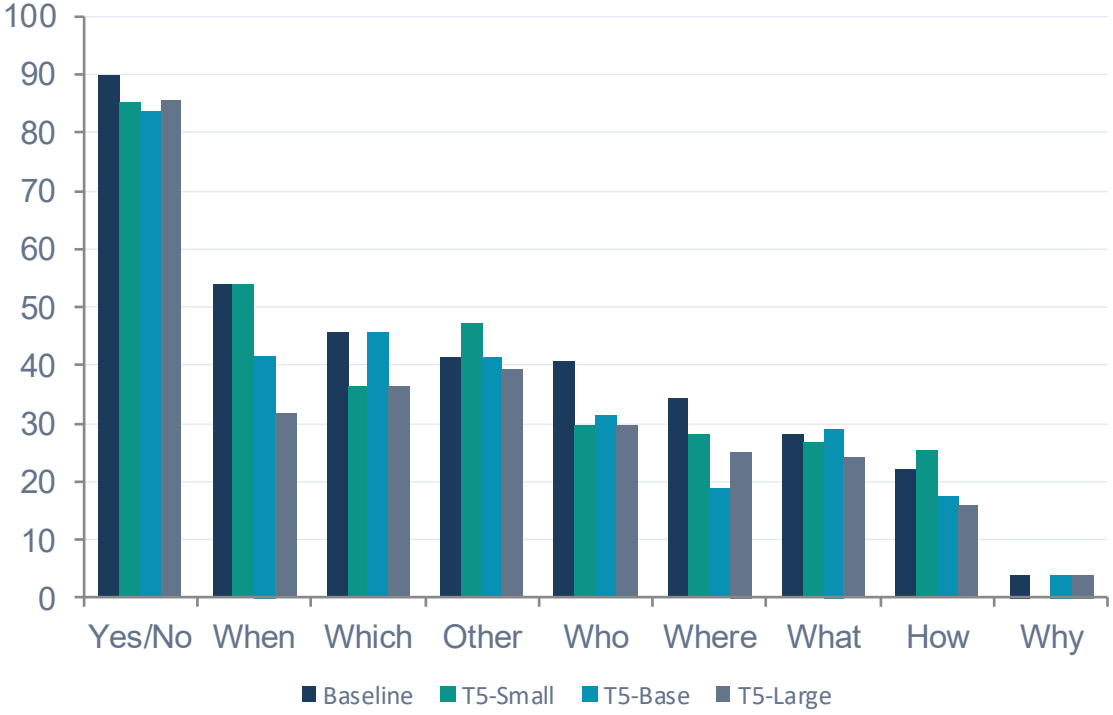




Baseline has highest accuracy (+5%) but consumes 2.6× more tokens and costs 2.5× more than T5-Summary variants.

Performance by Question Type

9× gap between best (Yes/No) and worst (Why) categories



Strong: Yes/No

>83% across all models (420 questions)

Moderate: What

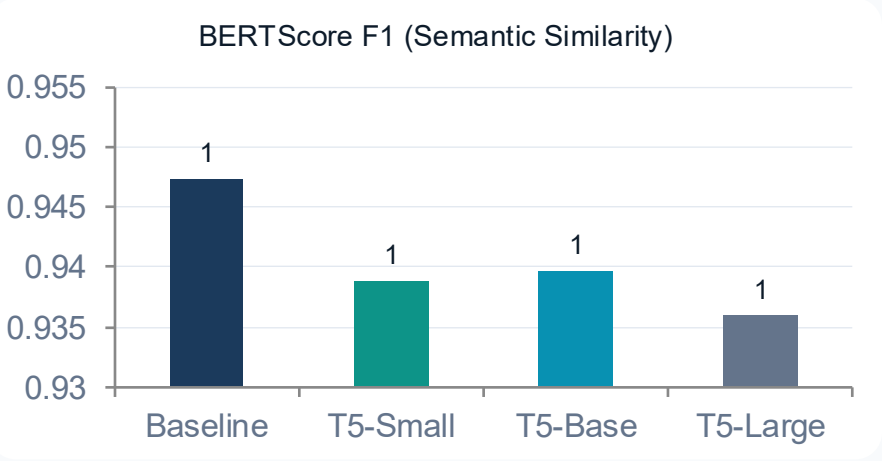
~27% avg, yet 24% of dataset — high impact

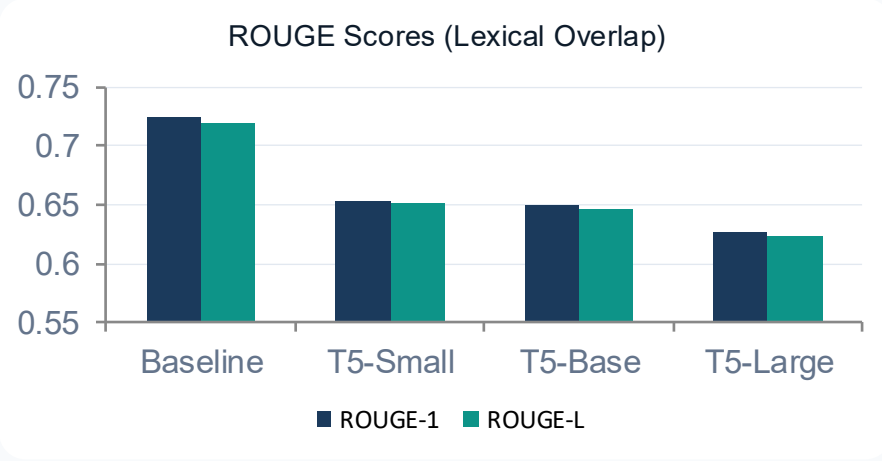
Critical Gap: Why

<5% average — needs chain-of-thought prompting

Semantic Quality: BERTScore & ROUGE

All models maintain high semantic quality despite lower exact match





Key Insight

All four models achieve BERTScore F1 > 0.93, indicating that T5-compressed answers remain semantically correct even when they don't exactly match the ground truth string. The exact match gap between Baseline and T5 variants is partially a surface-level artifact of answer phrasing, not a deep semantic failure.

Cost-Benefit Analysis

T5-Small delivers 95% of accuracy at 40% of the cost

4.61
T5-Small Efficiency Score (Accuracy / Cost — highest)

2.6×
More tokens used by Baseline vs T5 variants

60%
Token reduction with T5-Small (363K saved)

$0.18
Saved per 918 questions with T5-Small vs Baseline

Decision Matrix

Use Case	Recommended Model	Reason
Accuracy-critical (research/benchmark)	Baseline RAG	Best exact match (58.28%), highest BERTScore
Cost-sensitive / high-volume	T5-Base Summary	Lowest token count (231K), cheapest at scale
Balanced production workload	T5-Small Summary ★	Best efficiency ratio (4.61), 95% of accuracy
Avoid	T5-Large Summary	Lowest accuracy, no cost benefit over T5-Small

Key Findings & Future Work

Key Findings

- T5-Small achieves 95% of baseline accuracy at 40% of the cost — clear winner for production.
- Summarization trades ~5% exact match for 60%+ token savings, while maintaining >0.93 BERTScore.
- Larger T5 models do NOT improve quality — over-compression likely discards key facts.
- Yes/No questions perform strongly (>83%) across all models; Why questions remain a critical weakness (<5%).
- The method is model-agnostic and suitable for HPC batch pipelines with high query volumes.

Future Work

High Priority	Chain-of-thought prompting for Why/How questions (currently <5% accuracy)
High Priority	Hybrid routing: Baseline for complex questions, T5-Small for simple ones
Medium Priority	Named-entity-aware filtering to improve Who/Where accuracy
Medium Priority	Fine-tune T5 on HPC domain documents for better summarization quality
Long-term	Explore semantic-similarity-based sentence selection instead of word overlap

Conclusion

- Query-aware T5 summarization significantly reduces token usage in RAG pipelines without critical accuracy loss.
- T5-Small is the recommended default — best balance of cost (40%), accuracy (95% of baseline), and semantic quality.
- The approach is model-agnostic and ready for large-scale HPC deployment.
- Remaining challenge: open-ended reasoning questions (Why/How) need future work.

60%

token reduction

with T5-Small

at only −2.9% accuracy

Thank you