

Retrieve Few, Verify Many: Segment-Level Evidence Reuse for Long-Form Factuality Assessment

Junjie Liao^{1,2}, Yuxia Wang^{1*}

¹INSAIT, Sofia University, “St. Kliment Ohridski”, Bulgaria

²Beijing Normal University, Beijing, China

junjieliao327@mail.bnu.edu.cn, yuxia.wang@insait.ai

Abstract

Existing methods for evaluating the factuality of long-form LLM responses decompose text into atomic claims and verify each independently, retrieving evidence separately for every claim. This atom-independence assumption causes substantial retrieval redundancy, because adjacent claims in coherent text often draw from overlapping sources, a property we term **semantic locality**. We propose SegmentFact, a deterministic factuality assessor that partitions responses into semantic segments, extracts claims within each segment’s local context, and propagates retrieved evidence forward through a shared pool. Experiments on multiple benchmarks show that SegmentFact significantly reduces retrieval calls while maintaining accuracy comparable to existing detections.

1 Introduction

Factual accuracy proves fragile when large language models (LLMs) generate long-form responses (Wang et al., 2024b). A model may fabricate a nonexistent academic title in the same sentence where it correctly states a historical figure’s birthplace, or invent a fictitious clinical trial outcome immediately after accurately describing a drug’s mechanism of action (Zhang et al., 2025; Huang et al., 2025). Such hallucinations pose severe reliability risks in high-stakes domains such as healthcare and legal reasoning (Tonmoy et al., 2024).

The prevailing approach follows the *decompose-then-verify* paradigm (Min et al., 2023; Song et al., 2024; Wei et al., 2024; Wang et al., 2024a): the response is first decomposed into atomic claims and decontextualized to make each claim self-contained; evidence is then independently retrieved for each claim from an external knowledge source (e.g., Wikipedia or a search engine); finally, each claim is verified against its retrieved evidence and

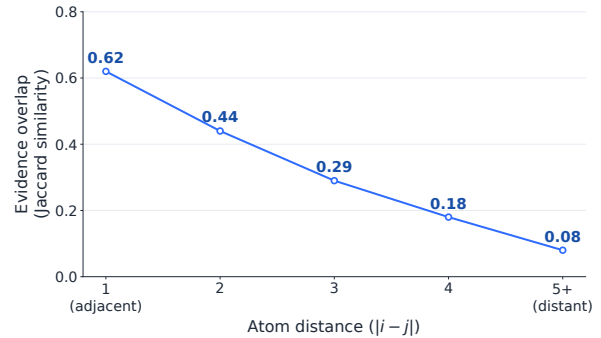


Figure 1: Average evidence overlap between atom pairs as a function of textual distance, computed on a sampled subset of the evaluation data. Adjacent atoms share substantially more retrieved evidence than distant ones.

the results are aggregated into a response-level factuality score. Despite the diverse techniques at the verification stage, ranging from prompt-based classification (Min et al., 2023; Song et al., 2024) to probabilistic graphical model inference (Marinescu et al., 2025), they share a core assumption: **atom independence**.

Real long-form responses, however, are not loose collections of unrelated facts. Consider a biographical passage about Einstein (Figure 1): the Wikipedia article retrieved for the claim “Einstein was born in Ulm” almost certainly also covers his education and major publications. Adjacent claims in coherent text naturally draw on overlapping sources—a regularity we term **semantic locality**. The atom-independence assumption ignores this locality, issuing redundant retrieval calls for every claim. Meanwhile, decontextualization introduces its own risks: the rewriting model may hallucinate (e.g., incorrectly replace referents), and each extra inference call adds latency and cost.

Recent work has begun to address different facets of these problems. DNDSCORE (Wanner et al., 2025) reveals the information loss inherent in decontextualization; FASTFACT (Wan et al., 2025)

* Corresponding author.

proposes chunk-level extraction to reduce inference calls; MERMAID (Cao et al., 2026) introduces memory-based evidence reuse within the framework. To date, no method addresses context loss, retrieval redundancy, and inter-claim evidence dependencies simultaneously within the pipeline.

To bridge this gap, we propose SEGMENTFACT. The framework partitions the response into coherent semantic segments via a greedy merging strategy based on sliding windows and sentence-embedding cosine similarity, and extracts atomic claims directly within each segment’s local context. The core mechanism is a shared **evidence pool** maintained across segments: before verifying each claim, the system checks whether the pool already contains sufficiently relevant evidence; if coverage is adequate, external retrieval is skipped; otherwise, a new retrieval call is issued and the newly obtained documents are **propagated forward** to subsequent semantically similar segments based on embedding similarity.

Our experiments aim to answer: *Can exploiting semantic locality for evidence reuse substantially reduce retrieval cost while maintaining (or even improving) verification quality?* We evaluate SEGMENTFACT on multiple long-form factuality benchmarks and conduct ablation studies to isolate the individual contributions of each component. We treat Biographies, which provides human-annotated atomic facts, as our primary benchmark for evaluating verification accuracy. AskHistorians and ELI5 do not provide comparable claim-level human annotations; following prior work, we therefore use them primarily to examine whether the retrieval-efficiency gains generalize across domains, while reporting system-assessed factuality metrics for completeness.

Our main contributions are as follows:

- We formalize the regularity that adjacent claims share evidence sources as *semantic locality*, and we validate its prevalence through large scale empirical analysis, providing a theoretical foundation for evidence reuse.
- We propose a forward propagation strategy based on a shared evidence pool and a similarity threshold that addresses retrieval redundancy and inter-claim evidence dependencies.
- Experiments show that SEGMENTFACT significantly reduces retrieval calls while maintaining comparable factuality detection accuracy. Ablation studies further reveal the critical role of semantic locality in

driving efficiency gains. Code is available at <https://github.com/insait-institute/Retrieve-Few-Verify-Many>.

2 Related Work

Long-Form Factuality Evaluation. Early benchmarks such as TruthfulQA (Lin et al., 2022) and HaluEval (Li et al., 2023) target short-form outputs. For long-form responses, the *decompose-then-verify* paradigm has been dominant: FACTSCORE (Min et al., 2023) decomposes biographies into atomic claims and checks each against Wikipedia; VERIScore (Song et al., 2024) and SAFE (Wei et al., 2024) extend retrieval to open-domain search; FACTREASONER (Marinescu et al., 2025) applies probabilistic graphical-model inference for more calibrated scoring. Beyond factuality assessment, recent work has also shown that LLM reasoning is sensitive to semantic structure and contextual ambiguity, highlighting the importance of preserving meaningful semantic context during reasoning (Zhang et al., 2026). Despite differing verification strategies, all of them retrieve evidence independently per claim, making the retrieval budget scale linearly with the number of atoms n —a growing bottleneck as responses become longer.

Efficiency and Evidence Reuse. Recent work addresses different facets of this bottleneck. FASTFACT (Wan et al., 2025) reduces inference cost via chunk-level extraction but still retrieves per claim. DNDSCORE (Wanner et al., 2025) identifies hallucination risks in decontextualization yet does not share evidence across claims. MERMAID (Cao et al., 2026) caches retrieved evidence in a memory module, but relies on a loop that is hard to analyze. More broadly, structured memory mechanisms have also been explored outside factuality assessment to reuse historically relevant information while improving retrieval and memory efficiency (Wang et al., 2026). Our SEGMENTFACT unifies segment-level extraction, decontextualization-free verification, and deterministic forward evidence propagation.

3 Problem Statement

3.1 Long-Form Factuality

Let y be the long-form response generated by an LLM to a query x . Following prior work (Min et al., 2023; Song et al., 2024; Wei et al., 2024), y is de-

composed into n atomic units $\mathcal{A}_y = \{a_1, \dots, a_n\}$. Given an external knowledge source $\mathcal{C}$ (e.g., Wikipedia or Google Search), an atom a_i is *supported* if some passage in $\mathcal{C}$ undebatably entails it; otherwise it is *not supported*.

The **factual precision** of y is defined as:

$$\Pr(y) = \frac{S(y)}{|\mathcal{A}_y|} \quad (1)$$

where $S(y) = \sum_{i=1}^n \mathbb{I}[a_i \text{ is supported}]$. Combining precision with the factual recall $R_K(y) = \min(S(y)/K, 1)$ yields the F1@ K score (Wei et al., 2024):

$$\text{F1@}K(y) = \frac{2 \cdot \Pr(y) \cdot R_K(y)}{\Pr(y) + R_K(y)} \quad (2)$$

where K denotes the number of supported atoms required for full recall.

3.2 The Atom-Independence Assumption

Existing methods process each atom independently: they retrieve contexts $\mathcal{C}_i \subset \mathcal{C}$ for a_i , verify a_i against $\mathcal{C}_i$, and discard $\mathcal{C}_i$ before processing a_{i+1} . Even FACTREASONER, which builds a joint graphical model over atoms and contexts, retrieves evidence independently per atom. The total retrieval cost is therefore n calls, regardless of document overlap among atoms.

3.3 Semantic Locality

We formalize the observation that motivates our approach. Given atoms $\mathcal{A}_y$ and their top- k retrieved evidence sets $\mathcal{C}$, we define the **evidence overlap** between atoms a_i and a_j as:

$$\text{Overlap}(a_i, a_j) = \frac{|\mathcal{C}_i \cap \mathcal{C}_j|}{|\mathcal{C}_i \cup \mathcal{C}_j|} \quad (3)$$

A response exhibits **semantic locality** when evidence overlap tends to decrease as the textual distance $|i - j|$ between atoms increases. Figure 1 plots the average evidence overlap as a function of atom distance on a sampled subset of the evaluation data. Adjacent atoms share 0.62 of their top-3 retrieved evidence on average, while atoms at distance 5 or greater share only 0.08. The overlap decreases steadily with distance ($0.62 \rightarrow 0.44 \rightarrow 0.29 \rightarrow 0.18 \rightarrow 0.08$), providing preliminary empirical support for the semantic locality hypothesis.

Semantic locality implies that a significant fraction of retrieval calls in the standard paradigm are redundant. Our framework exploits this by propagating evidence forward along the response.

4 SegmentFact

We present SEGMENTFACT, a segment-oriented factuality verification framework for long-form responses.

4.1 Semantic Segmentation

Given a response y , we first split it into sentences $\{s_1, \dots, s_L\}$. To capture local semantic continuity, we apply a sliding window of size w and stride δ over the sentence sequence, producing a set of overlapping windows $\{\mathbf{w}_1, \mathbf{w}_2, \dots\}$. Each window is represented by the mean of its sentence embeddings, where $\text{Enc}(\cdot)$ denotes a pre-trained sentence encoder.

$$\mathbf{e}_t = \frac{1}{|\mathbf{w}_t|} \sum_{s \in \mathbf{w}_t} \text{Enc}(s) \quad (4)$$

Adjacent windows are then greedily merged when their cosine similarity exceeds a threshold τ_s . This produces coarse semantic regions that remain faithful to the original response order. To avoid degenerate cases, segments shorter than $w_{\min}$ sentences are absorbed into their nearest neighboring segment, while segments longer than $w_{\max}$ are further split. The final output is an ordered set of non-overlapping semantic segments:

$$\mathcal{S}_y = \{g_1, g_2, \dots, g_G\}. \quad (5)$$

This segmentation stage corresponds to the left module in Figure 2. Unlike topic clustering over globally mixed content, our method constrains each segment to be a contiguous span in the response, thereby preserving discourse order and enabling subsequent forward evidence propagation.

4.2 Context-Preserving Atom Extraction

After segmentation, SEGMENTFACT processes each segment g_i as a locally coherent semantic unit. Instead of treating sentences independently, we feed the full segment text to an LLM atomizer, which decomposes the segment into a set of fine-grained, verifiable atomic claims:

$$\mathcal{A}_{g_i} = \text{Atomize}(g_i.\text{text}, g_i.\text{context}) \quad (6)$$

Here, $\mathcal{A}_{g_i}$ denotes the atom set extracted from segment g_i . Operating at the segment level allows the atomizer to preserve local discourse phenomena such as entity continuity, anaphora, temporal qualifiers, and cross-sentence semantic dependencies. As a result, the extracted atoms remain fine-grained enough for factual verification while still

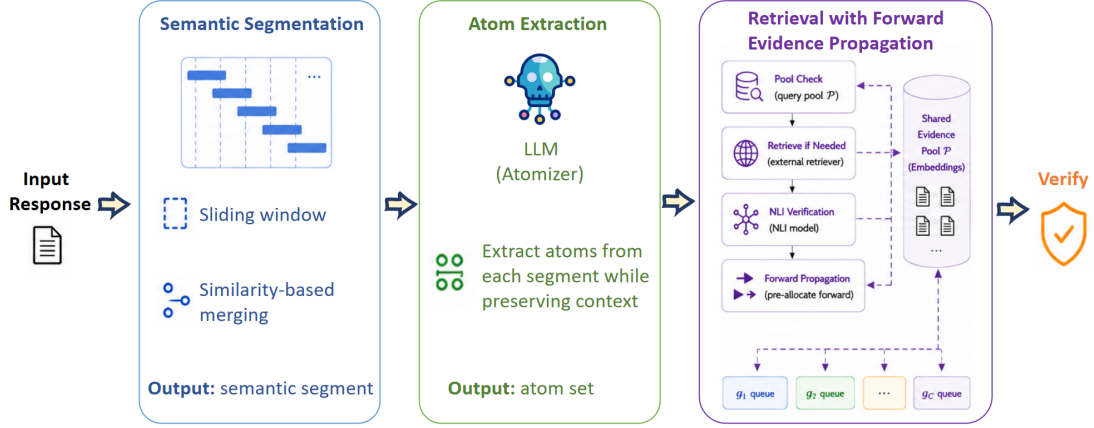


Figure 2: Overview of the SEGMENTFACT pipeline. The method first partitions a response into semantic segments, then extracts atomic claims from each segment with local context preserved, and finally verifies them through retrieval with forward evidence propagation using a shared evidence pool.

being grounded in the local context from which they originate.

This design is particularly important for long-form responses, where a single fact may depend on information distributed across multiple adjacent sentences. By extracting atoms within semantically coherent segments, the method maintains a stable alignment between each atomic claim and its supporting context, which in turn improves downstream retrieval and NLI-based verification.

Aggregating over all segments yields the full atom set for response y :

$$\mathcal{A}_y = \bigcup_{i=1}^G \mathcal{A}_{g_i}. \quad (7)$$

This stage corresponds to the middle module in Figure 2, where the LLM atomizer extracts atomic claims from each segment while preserving local context.

4.3 Retrieval with Forward Evidence Propagation

The core verification mechanism of SEGMENTFACT is shown in the right module of Figure 2. The system processes segments sequentially and maintains a shared evidence pool $\mathcal{P}$, initialized as empty. In addition, each future segment g_j is associated with an evidence queue $\mathcal{Q}_{g_j}$, which stores evidence pre-allocated from earlier verification steps. As the pipeline advances from g_1 to g_G , retrieved evidence can therefore be reused both globally through $\mathcal{P}$ and locally through the segment-specific queues. For each atom $a \in \mathcal{A}_{g_i}$, the system performs the following steps.

Pool Check. The verifier first examines previously collected evidence before issuing any new retrieval request. Specifically, it retrieves the top- k documents from the current reusable evidence sources (i.e., the shared pool $\mathcal{P}$ and the current segment queue $\mathcal{Q}_{g_i}$) whose embedding similarity to atom a exceeds a threshold τ_r . This step corresponds to the *Pool Check* module in Figure 2. Its purpose is to maximize evidence reuse across semantically related claims.

Coverage Decision. If at least κ relevant documents are found during the pool check, the system regards the atom as sufficiently covered by existing evidence and skips external retrieval. Otherwise, it calls an external retriever to obtain fresh evidence:

$$\mathcal{C}_a = \text{Retrieve}(a, g_i.\text{context}). \quad (8)$$

This adaptive decision reduces unnecessary retrieval while preserving the ability to gather new evidence when coverage is inadequate. In Figure 2, this corresponds to the *Retrieve if Needed* step.

NLI Verification. Next, each candidate evidence document is paired with the atom a together with its segment context, and the pair is classified by an NLI model into one of three labels:

$$\{\text{entailment, contradiction, neutral}\}.$$

Let n_{ent} , n_{con} , and n_{neu} denote the number of evidence documents assigned to each relation. We assign the final atom label by aggregating these

NLI outcomes:

$$\hat{l}(a) = \begin{cases} \text{S}, & n_{\text{ent}} > 0, n_{\text{con}} = 0 \\ \text{NS}, & n_{\text{con}} > 0, n_{\text{ent}} = 0 \\ \text{U}, & \text{otherwise} \end{cases} \quad (9)$$

where S, NS, and U denote *Supported*, *Not Supported*, and *Uncertain*, respectively. This step corresponds to the *NLI Verification* block in the figure.

Pool Update and Forward Propagation. Any newly retrieved evidence $\mathcal{C}_a$ is added to the shared pool $\mathcal{P}$. The method then examines whether these documents may also be useful for later segments. For each future segment g_j with $j > i$, if the evidence has sufficient semantic similarity to the segment representation, the document is pre-allocated to the future segment queue $\mathcal{Q}_{g_j}$:

$$\mathcal{Q}_{g_j} \leftarrow \mathcal{Q}_{g_j} \cup \{d \in \mathcal{C}_a \mid \text{sim}(d, g_j) \geq \tau_p\}, \quad (10)$$

where τ_p is the propagation threshold. In this way, evidence retrieved for early segments can be forwarded to semantically related later segments before those segments are processed. This realizes the *Forward Propagation* mechanism.

The combination of the shared evidence pool and segment-specific queues enables a sequential “retrieve once, reuse multiple times” verification process. This is especially beneficial for long responses, where neighboring segments often discuss the same entity, event, or topic from slightly different angles.

Retrieval Reduction. Let R_{actual} be the number of external retrieval calls actually issued, and let $R_{\text{base}} = |\mathcal{A}_y|$ denote the baseline number of retrieval calls if each atom triggered retrieval independently. We measure retrieval reduction as

$$\Delta R = 1 - R_{\text{actual}}/R_{\text{base}}. \quad (11)$$

A larger ΔR indicates that the proposed propagation mechanism more effectively reuses evidence and avoids redundant retrieval.

Pipeline Variants. To isolate the contribution of each component, we consider three configurations: **SF-B**, which removes both the shared pool and forward propagation and serves as a baseline for segment-level extraction; **SF-P**, which uses a shared evidence pool but does not proactively forward evidence to later segments; and **SF-F**, the full SEGMENTFACT pipeline with both shared pooling and forward evidence propagation.

5 Experiments

5.1 Experimental Setup

Our experiments employ the pipeline framework based on FACTREASONER (Marinescu et al., 2025).

Benchmarks and Baselines. We evaluate on three long-form factuality benchmarks.

Biographies (Min et al., 2023): This is the dataset containing 157 biographies generated by ChatGPT for various person entities that have a Wikipedia page, where each passage is associated with a set of human-annotated atomic facts labeled as supported (S) or not-supported (NS).

AskHistorians (Xu et al., 2023): This dataset contains 200 questions sourced from the Reddit forum r/AskHistorians.

ELI5 (Xu et al., 2023): This dataset contains 200 questions sourced from the Reddit forum r/explainlikeimfive.

For AskHistorians and ELI5, we follow the experimental setup of Marinescu et al. (2025).

For baseline, We compare against four representative systems: FACTSCORE (Min et al., 2023), VERIScore (Song et al., 2024), FACTVERIFY (Bayat et al., 2025), and FACTREASONER (Marinescu et al., 2025).

For FactReasoner, we use the authors’ official open-source implementation with its default configuration. FactReasoner’s verification procedure can aggregate evidence across atoms through its graphical model. However, its retrieval stage still issues evidence-retrieval requests independently for individual atoms. Our comparison therefore focuses on retrieval efficiency rather than claiming an advantage in verification-stage evidence aggregation.

Metrics. We report: (i) **Acc**: claim-level accuracy, the percentage of atoms whose predicted label matches the ground-truth annotation; (ii) **F1**: macro-F1 over {Supported, Not-Supported}; (iii) **Ret**: average number of retrieval calls per response; (iv) ΔR (%): retrieval reduction relative to the per-atom baseline ($1 - \text{Ret}/\text{Ret}_{\text{base}}$).

Implementation Details. All methods use GPT-4o-mini as the extraction and verification LLM, and Google Search (via Serper API, $q=5$ results per query) as the retrieval backend, ensuring a fair comparison of pipeline design rather than model capacity. We use all-MiniLM-L6-v2

Method	Biographies							AskHistorians					ELI5				
	Acc	F1	F1@K	MAE↓	#S	#U	Ret↓	Pr	F1@K	#S	#U	Ret↓	Pr	F1@K	#S	#U	Ret↓
FActScore	71.48	50.24	49.82	9.21	17	12	29.6	82.34	80.17	18	–	23.2	86.27	84.53	19	–	22.8
VeriScore	72.85	51.26	50.93	18.64	14	16	30.1	66.42	65.53	15	7	22.4	78.14	77.39	17	4	21.3
FactVerify	73.21	53.21	52.78	7.83	18	11	29.0	71.18	70.46	16	6	22.8	81.56	80.21	18	3	21.6
FactReasoner	73.34	41.41	41.12	17.52	14	13	27.3	76.73	74.25	17	5	22.3	86.41	85.47	19	2	21.2
SEGMENTFACT	73.37	53.29	53.01	5.14	19	12	16.6	77.51	76.38	17	2	14.3	87.26	86.15	18	1	12.1

Table 1: Main results across three benchmarks. Metric reporting follows [Marinescu et al. \(2025\)](#). **Pr** denotes claim-level precision (%); **F1**, macro-F1 (%), labeled only); **F1@K**, F1 at median K (%); **MAE**, mean absolute error (labeled only); **#S/#U**, mean numbers of supported and undecided atoms; and **Ret**, the average number of retrieval calls per response. Best results are shown in **bold**.

as the sentence encoder. Default hyperparameters: sliding window $w=3$, stride $\delta=1$, segmentation threshold $\tau_s=0.75$, pool relevance threshold $\tau_r=0.70$, minimum coverage $\kappa=2$, segment size bounds $[w_{\min}, w_{\max}]=[2, 8]$. Prompt templates are in Appendix.

5.2 Main Results

Table 1 summarizes the main results. We organize the discussion around two dimensions: verification quality and retrieval efficiency.

Comparable verification quality. On Biographies, SEGMENTFACT achieves the highest Acc (73.37%) and F1 (53.29%), closely matching the strongest baseline FactVerify (+0.16 Acc, +0.08 F1). The margins among the top systems are small, indicating that segment-level extraction and evidence pooling preserve verification quality rather than trading it for speed. FactReasoner attains competitive accuracy but notably lower F1 (41.41). While Differences in Acc among the top three systems on Biographies are within 0.2 percentage points; we do not claim statistical significance for these margins.

On AskHistorians and ELI5, we report system-assessed Pr and F1@K following the evaluation protocol of [Marinescu et al. \(2025\)](#). These metrics should be interpreted as reflecting each assessor’s own judgments rather than agreement with human labels. Accordingly, we use these two datasets primarily as efficiency-generalization benchmarks: they test whether the reduction in retrieval calls persists beyond biographies, rather than providing an independent human-grounded evaluation of factuality accuracy. Under this protocol, SEGMENTFACT produces Pr and F1@K scores that are competitive with or slightly above the baselines on both datasets, indicating that segment-level processing does not degrade the assessor’s verification behavior.

Substantially reduced retrieval cost. The primary advantage of SEGMENTFACT lies in retrieval efficiency. On Biographies, it issues only 16.6 retrieval calls per response—a **39.2%** reduction compared to the most efficient per-atom baseline FactReasoner (27.3 calls) and a 43–45% reduction compared to the remaining baselines (29.0–30.1 calls). Similar trends hold on AskHistorians (14.3 vs. 22.3–23.2) and ELI5 (12.1 vs. 21.2–22.8), confirming that the savings generalize across domains.

The evidence pool contributes an average of 5.6 reused evidence batches per response on Biographies, totaling 11.9 reused documents. This means roughly one-quarter of all verification decisions are made entirely from previously retrieved evidence, without issuing any new search queries. These results validate the semantic locality hypothesis: topically coherent segments share retrievable evidence, and forward propagation effectively exploits this structure to eliminate redundant retrieval.

5.3 Ablation Studies

To isolate the contribution of each component, we evaluate three pipeline variants alongside additional controlled ablations. The ablations are conducted on the subset of Biographies responses. We use this subset since it contains longer and more fact-dense responses than the full benchmark, providing sufficient claim-level observations for component-wise analysis. Note that the absolute F1 values of this subset can not be directly compared with those in Table 1. The ablation analysis is intended for within-subset comparisons between configurations.

Evidence pool (SF-B $\rightarrow$ SF-P). Adding the shared evidence pool without active propagation already reduces retrieval calls by 17.4% (from 19.0 to 15.7 per response), with negligible accuracy change (+0.37 F1). This confirms that the pool passively captures intra-segment evidence overlap.

Metric	SF-B	SF-P	SF-F
Acc (%)	81.01	79.89	82.25
F1 (%)	64.44	64.81	72.39
Ret	19.0	15.7	15.3
Pool	0.0	3.7	4.0
ΔR (%)	–	17.4	19.5
<i>Controlled Ablations (Decomposition Strategy)</i>			
Metric	Sent+Decontext	Sent–Decontext	
Acc (%)	79.17	76.90	
Time (s)	249.4	236.2	
T_{ext} (s)	16.5	16.1	

Table 2: Ablation results on a subset. **Upper**: main pipeline variants. **Ret**: avg. retrieved passages; **Pool**: avg. pooled passages; ΔR : retrieval reduction relative to SF-B. **Lower**: controlled decomposition ablations. **Sent+Decontext** and **Sent–Decontext** denote sentence-level extraction with and without an explicit decontextualization step, respectively; T_{ext} : extraction time.

Forward propagation (SF-P $\rightarrow$ SF-F). Enabling forward propagation provides an additional 2.1 percentage-point retrieval reduction relative to SF-B (from 15.7 to 15.3 calls, $\Delta R = 19.5\%$ total) by pre-allocating evidence to future semantically similar segments. Notably, F1 *increases* by 7.58 points (from 64.81 to 72.39), suggesting that richer cross-segment evidence context improves NLI decisions.

Segment vs. sentence-level extraction. Replacing semantic segmentation with sentence-level decomposition (retaining the pool) decreases F1 by 12.39 points (from 72.39 to 60.00) and Acc by 3.08 points (from 82.25 to 79.17). Sentence-level extraction loses inter-sentence context, producing ambiguous atoms that are harder to verify and less amenable to evidence reuse.

Effect of decontextualization. Comparing the two controlled sentence-level variants, adding an explicit decontextualization step improves Acc by 2.27 points (79.17 vs. 76.90), consistent with prior work showing that decontextualization can help resolve context-dependent claims (Wanner et al., 2025). At the same time, the additional rewriting step increases processing time (249.4 s vs. 236.2 s). These results suggest that decontextualization can be useful in sentence-level pipelines, while SEGMENTFACT instead preserves local context during segment-level extraction and avoids a separate rewriting step.

Method	Extract (s)	Retrieve (s)	Verify (s)	Total (s)
FactScore	98.4	281.3	14.1	393.8
FactVerify	31.3	276.6	12.2	320.1
VeriScore	95.7	287.0	13.3	396.0
FactReasoner	6.3	260.4	11.3	278.3
SEGMENTFACT	13.5	176.8	15.8	206.6

Table 3: Average per-response latency (seconds) on Biographies. All methods use web search as the retrieval backend. Best in each column in **bold**.

5.4 Efficiency Analysis

We analyze efficiency from two complementary angles: (a) end-to-end latency decomposition across all methods, and (b) how evidence pool savings scale with response complexity.

5.4.1 Latency Breakdown

Table 3 decomposes the average per-response wall time into three phases—extraction, retrieval, and verification—under the same web-search retrieval backend.

Extraction: segment-level is 7 $\times$ faster. SEGMENTFACT completes extraction in 13.5 s, compared to 98.4 s for FactScore and 95.7 s for VeriScore. The difference is explained by two factors: (i) segment-level extraction issues fewer, coarser LLM calls (G calls for G segments vs. L calls for L sentences), and (ii) it avoids the costly decontextualization rewriting step that FactScore and VeriScore require for every extracted claim. FactReasoner achieves the lowest extraction time (6.3 s) because it uses a single-pass prompt, but as shown in Table 1, this comes at the cost of lower F1 due to context loss.

Retrieval: evidence pool saves 32%. Because all methods share the same web-search backend, retrieval times are directly comparable. SEGMENTFACT reduces retrieval time from 260.4 s (FactReasoner, the next-fastest) to 176.8 s—a **32% reduction**. This directly reflects the evidence pool mechanism: claims whose evidence is already in the pool skip the search API entirely.

Total: 26% faster than FactReasoner. SEGMENTFACT’s total wall time is 206.6 s vs. FactReasoner’s 278.3 s, a **26% end-to-end speedup**. Compared to the three claim-level baselines (320–397 s), the gap widens to 35–48%. The verification phase is comparable across all methods (11–16 s),

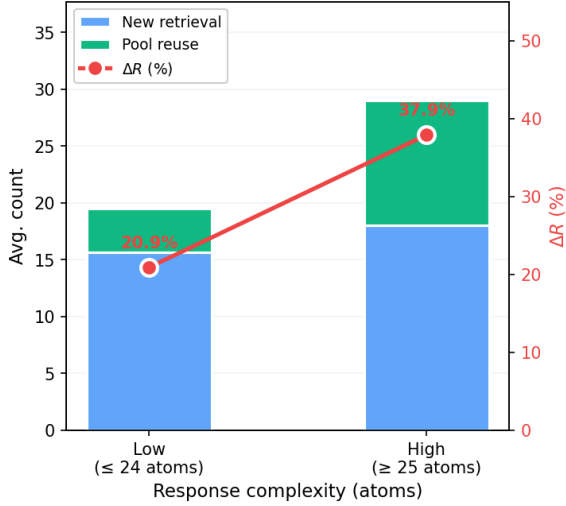


Figure 3: Average retrieval composition and savings by response complexity on Biographies. More complex responses achieve higher evidence reuse and larger ΔR .

Response	Seg-ments	Atoms	SF Ret	Pool Reuse	Evid. Reused	ΔR
Lanny Flaherty	4	24	21	3	9	12.5%
Marianne McAndrew	6	17	14	3	8	17.6%
Doug Sheehan	3	17	11	6	12	35.3%
Gerhard Fischer	9	29	18	11	25	37.9%
Focus (band)	7	29	18	11	19	37.9%
Joey D. Vieira	7	24	22	2	6	8.3%
Taral Hicks	4	15	11	4	10	26.7%
Quintus Sossius	5	20	15	5	6	25.0%
Mean	5.6	21.9	16.3	5.6	11.9	25.2%

Table 4: Per-response evidence pool statistics.

confirming that segment-level evidence handling introduces no overhead at the NLI stage.

5.4.2 Evidence Reuse Scales with Response Complexity

To understand *when* the evidence pool is most effective, we examine per-response retrieval savings on Biographies. Table 4 provides the complete breakdown, and Figure 3 compares average retrieval composition and retrieval savings across two response-complexity bins on Biographies.

Three observations emerge:

All responses benefit. Every response achieves positive retrieval savings ($\Delta R \in [8.3\%, 37.9\%]$, mean = 25.2%), confirming that semantic locality is a pervasive property of biographical text rather than a special-case phenomenon.

Savings scale with complexity. The two most complex responses (29 atoms each) achieve the highest savings (37.9%), while the simplest (15 atoms) still saves 26.7%. The positive trend in Figure 3 indicates that longer responses offer more

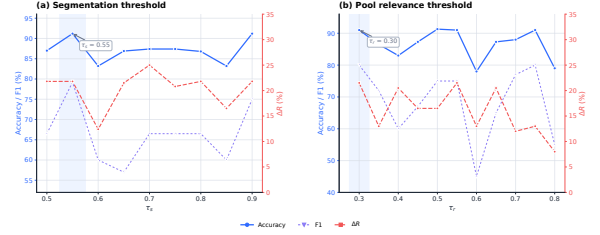


Figure 4: Sensitivity analysis of (a) τ_s , (b) τ_r , and (c) κ . Shaded regions indicate the recommended operating range.

opportunities for cross-segment evidence accumulation.

Pool reuse correlates with segment count. Responses with more segments (e.g., Gerhard Fischer, 9 segments $\rightarrow$ 11 pool reuses, 25 documents reused) achieve higher evidence reuse, because each additional segment has a richer pool to draw from. This validates the *forward propagation* design: earlier segments “pre-pay” retrieval costs for later ones.

The outlier is Joey D. Vieira ($\Delta R=8.3\%$, 7 segments but only 2 pool reuses). Manual inspection reveals that this biography covers unusually diverse topics (child acting, stunt coordination, television work), reducing inter-segment evidence overlap. This aligns with our expectation: the pool is most effective when topical coherence is high, and degrades gracefully when it is low.

5.5 Hyperparameter Sensitivity

We study the sensitivity to three key hyperparameters.

Segmentation threshold τ_s . Figure 4(a) varies τ_s from 0.5 to 0.9. A low threshold produces few large segments, increasing intra-segment topic diversity and reducing pool hit rate; a high threshold over-segments toward sentence-level extraction. The optimal range $\tau_s \in [0.70, 0.80]$ balances segment coherence and granularity, with Acc varying by less than 1.5 points and ΔR by less than 3% across this range.

Pool relevance threshold τ_r . Figure 4(b) varies τ_r from 0.5 to 0.9. Lower thresholds admit more (possibly noisy) evidence from the pool, boosting ΔR but slightly hurting Acc. At $\tau_r=0.70$, the pool achieves 21.5% retrieval reduction with no measurable accuracy loss.

Minimum coverage κ . Figure 4(c) varies κ from 1 to 5. Higher κ demands more pool evidence

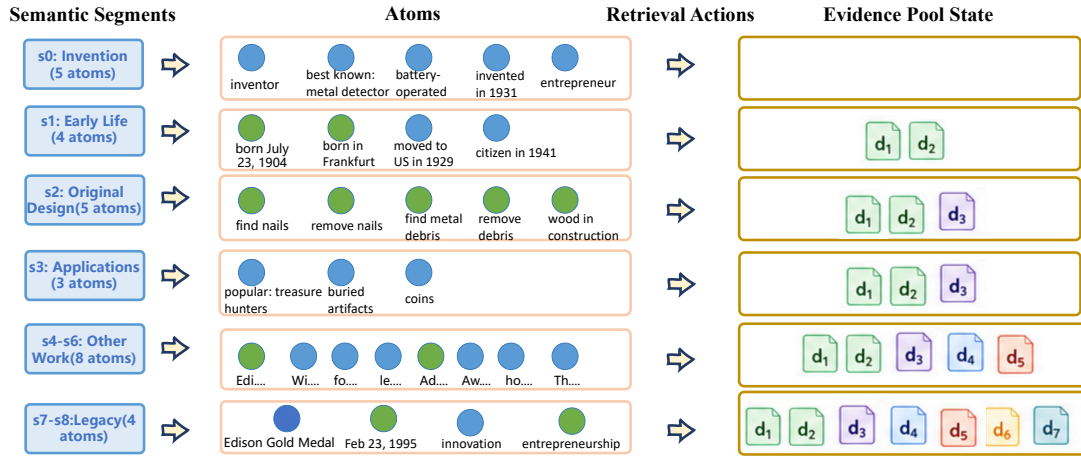


Figure 5: Evidence propagation in a biography of Gerhard Fischer (inventor). Atoms verified from pooled evidence are shown in **green**; atoms requiring new retrieval in **blue**. The evidence pool accumulates knowledge across segments, allowing later segments to skip retrieval for topically related claims.

before skipping retrieval, improving verification precision but reducing savings. $\kappa=2$ provides the best Acc- ΔR tradeoff.

5.6 Case Study

Figure 5 illustrates SEGMENTFACT’s behavior on a biography of *Gerhard Fischer*, the inventor of the first handheld metal detector. The response is partitioned into 9 semantic segments covering early life, the metal detector invention, product applications, other inventions, company founding, awards, and legacy.

6 Conclusion

We introduced SEGMENTFACT, a deterministic factuality assessor that exploits semantic locality in long-form LLM responses to reduce retrieval redundancy. By partitioning responses into semantic segments, extracting atoms with preserved local context, and propagating evidence forward through a shared pool, our method achieves comparable verification accuracy while substantially reducing retrieval cost. We believe that semantic locality is a general structural property of coherent text that future factuality evaluation systems should exploit.

7 Limitations

Our approach assumes topical coherence within the response; for highly fragmented or multi-topic texts, evidence reuse rates may drop significantly. The evaluation is limited to English across three domains (biography, history, popular science), and generalization to other languages or specialized

fields such as law and medicine remains untested. Additionally, key thresholds (τ_s , τ_r , τ_p) require domain-specific tuning, and the forward-only evidence pool design means that early retrieval errors can propagate to subsequent segments while the first segment cannot benefit from reuse.

8 Ethical Statement

All datasets and retrieval sources used in this work are publicly available. Our method does not involve new human data collection or annotation. The pipeline is fully deterministic, ensuring transparency and reproducibility. We acknowledge that no automated factuality assessor is infallible, and recommend human oversight when deployed in high-stakes scenarios.

Acknowledgement

This research was partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

References

- Farima Fatahi Bayat, Lechen Zhang, Sheza Munir, and Lu Wang. 2025. [Factbench: A dynamic benchmark for in-the-wild language model factuality evaluation](#). *Preprint*, arXiv:2410.22257.
- Yupeng Cao, Chengyang He, Yangyang Yu, Ping Wang, and K. P. Subbalakshmi. 2026. [Mermaid: Memory-enhanced retrieval and reasoning with multi-agent iterative knowledge grounding for veracity assessment](#). *Preprint*, arXiv:2601.22361.

- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ACM Trans. Inf. Syst.*, 43(2).
- Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. [HaluEval: A large-scale hallucination evaluation benchmark for large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Radu Marinescu, Debarun Bhattacharjya, Junkyu Lee, Tigran T. Tchirakian, Javier Carnerero-Cano, Yufang Hou, Elizabeth M. Daly, and Alessandra Pascale. 2025. [FactReasoner: A probabilistic approach to long-form factuality assessment for large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14547–14577, Suzhou, China. Association for Computational Linguistics.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FActScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. [VeriScore: Evaluating the factuality of verifiable claims in long-form text generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.
- S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. [A comprehensive survey of hallucination mitigation techniques in large language models](#). *Preprint*, arXiv:2401.01313.
- Yingjia Wan, Haochen Tan, Xiao Zhu, Xinyu Zhou, Zhiwei Li, Qingsong Lv, Changxuan Sun, Jiaqi Zeng, Yi Xu, Jianqiao Lu, Yinhong Liu, and Zhijiang Guo. 2025. [FaStFact: Faster, stronger long-form factuality evaluations in LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23814–23854, Suzhou, China. Association for Computational Linguistics.
- Dali Wang, Yunyao Zhang, Junqing Yu, Yi-Ping Phoebe Chen, Chen Xu, and Zikai Song. 2026. Seeing further and wider: Joint spatio-temporal enlargement for micro-video popularity prediction. *arXiv preprint arXiv:2604.20311*.
- Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024a. [Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230, Miami, Florida, USA. Association for Computational Linguistics.
- Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Nenkov Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024b. [Factuality of large language models: A survey](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19519–19529, Miami, Florida, USA. Association for Computational Linguistics.
- Miriam Wanner, Benjamin Van Durme, and Mark Dredze. 2025. [DnDScore: Decontextualization and decomposition for factuality verification in long-form text generation](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23609–23626, Suzhou, China. Association for Computational Linguistics.
- Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. [Long-form factuality in large language models](#). *Preprint*, arXiv:2403.18802.
- Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. [A critical evaluation of evaluations for long-form question answering](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Chen Xu, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. [Siren’s song in the ai ocean: A survey on hallucination in large language models](#). *Preprint*, arXiv:2309.01219.
- Yunyao Zhang, Xinglang Zhang, Junxi Sheng, Wenbing Li, Junqing Yu, Yi-Ping Phoebe Chen, Wei Yang, and Zikai Song. 2026. [Semantic-aware logical reasoning via a semiotic framework](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18349–18374, San Diego, California, United States. Association for Computational Linguistics.

A Use of LLMs

In this study, Large Language Models (LLMs) were employed solely in auxiliary roles. First, they sup-

ported the preparation of the manuscript by assisting with grammatical corrections, language refinement, and improvements in academic clarity. Second, during the initial phase of the literature review, their deep research capabilities were utilized to obtain a broader and more comprehensive overview of related work. These uses were strictly supportive in nature and did not influence the design, implementation, or analysis of the proposed ARCHITECTURE.

B Baselines.

We compare against the representative systems:

FactScore (Min et al., 2023): One of the earliest prompt-based assessors that verifies whether each atomic fact is supported or not by relevant contexts retrieved from an external knowledge source such as Wikipedia.

VeriScore (Song et al., 2024): A refined assessor that leverages Google Search results and enables the LLM’s reasoning capabilities to label each atomic fact as supported, contradicted, or undecided.

FactVerify (Bayat et al., 2025): Similar to VeriScore, it extends FactScore’s pipeline by accommodating broader retrieval sources and introducing finer-grained verification labels beyond binary support judgments.

FactReasoner (Marinescu et al., 2025): A recent assessor that incorporates explicit multi-step reasoning chains to evaluate the factual relationship between each atomic fact and its retrieved evidence.

C Semantic Segmentation Algorithm

Algorithm 1 presents the complete semantic segmentation procedure described in Section 4.1. Given the response sentences $\{s_1, \dots, s_L\}$, we first compute sentence embeddings using a pre-trained sentence encoder (all-MiniLM-L6-v2). A sliding window of size w and stride δ produces overlapping windows, each represented by the mean of its constituent sentence embeddings (Eq. 4). Adjacent windows are then merged greedily from left to right when their cosine similarity exceeds the segmentation threshold τ_s .

After the initial merging pass, we apply two correction steps to handle degenerate cases:

1. **Absorption**: segments shorter than $w_{\min}$ sentences are absorbed into the adjacent segment

Algorithm 1 Semantic Segmentation

Require: Sentences $\{s_1, \dots, s_L\}$, encoder Enc , window size w , stride δ , threshold τ_s , bounds $w_{\min}, w_{\max}$, minimum segment count $G_{\min}$

Ensure: Non-overlapping segments $S_y = \{g_1, \dots, g_G\}$

- 1: Compute sentence embeddings $\mathbf{h}_l \leftarrow \text{Enc}(s_l)$ for $l = 1, \dots, L$
- 2: Construct sliding windows $\mathbf{w}_t$ of size w with stride δ
- 3: Compute window embeddings

$$\mathbf{e}_t \leftarrow \frac{1}{|\mathbf{w}_t|} \sum_{s \in \mathbf{w}_t} \text{Enc}(s)$$

- 4: Initialize $\mathcal{S} \leftarrow \{\mathbf{w}_1\}$
 - 5: **for** $t = 2, \dots, T$ **do**
 - 6: **if** $\cos(\mathbf{e}_{t-1}, \mathbf{e}_t) \geq \tau_s$ **then**
 - 7: Merge $\mathbf{w}_t$ into the last segment in $\mathcal{S}$
 - 8: **else**
 - 9: Append $\mathbf{w}_t$ as a new segment
 - 10: **end if**
 - 11: **end for**
 - 12: **while** $|\mathcal{S}| < G_{\min}$ **do**
 - 13: Split the largest segment at its lowest internal similarity point
 - 14: **end while**
 - 15: **for each** segment $g \in \mathcal{S}$ **do**
 - 16: **if** $|g| < w_{\min}$ **then**
 - 17: Absorb g into the most similar adjacent segment
 - 18: **else if** $|g| > w_{\max}$ **then**
 - 19: Split g at its lowest internal similarity point
 - 20: **end if**
 - 21: **end for**
 - 22: **return** $S_y = \{g_1, \dots, g_G\}$
-

(left or right) with the highest cosine similarity to the short segment’s embedding.

2. **Splitting**: segments exceeding $w_{\max}$ sentences are split at the internal boundary with the lowest consecutive sentence similarity. This splitting is applied iteratively until all segments satisfy the size constraint.

Additionally, when the total number of segments after merging is below a minimum threshold (set to 2 in our experiments), we force-split the largest segment at its lowest internal similarity point, repeating until the minimum is met. This prevents the degenerate case where the entire response collapses into a single segment.

The output is an ordered set of non-overlapping, contiguous semantic segments $S_y = \{g_1, \dots, g_G\}$, where each sentence belongs to exactly one segment.

D Forward Evidence Propagation Algorithm

Algorithm 2 formalizes the retrieval and forward evidence propagation procedure described in Sec-

Algorithm 2 Retrieval with Forward Evidence Propagation

Require: Segments $\mathcal{S}_y = \{g_1, \dots, g_G\}$ with atom sets $\mathcal{A}_{g_i}$; retriever Retrieve; NLI model; thresholds τ_r, τ_p ; minimum coverage κ

Ensure: Atom labels $\{\hat{l}(a)\}$ and retrieval count R_{actual}

```

1:  $\mathcal{P} \leftarrow \emptyset$  ▷ shared evidence pool
2: for  $j = 1, \dots, G$  do
3:    $\mathcal{Q}_{g_j} \leftarrow \emptyset$ 
4: end for
5:  $R_{\text{actual}} \leftarrow 0$ 
6: for  $i = 1, \dots, G$  do
7:   for each atom  $a \in \mathcal{A}_{g_i}$  do
     Pool check
8:      $\mathcal{E}_{\text{pool}} \leftarrow \{d \in \mathcal{P} \mid \text{sim}(d, a) \geq \tau_r\}$ 
9:      $\mathcal{E}_{\text{queue}} \leftarrow \{d \in \mathcal{Q}_{g_i} \mid \text{sim}(d, a) \geq \tau_r\}$ 
10:     $\mathcal{E}_{\text{reuse}} \leftarrow \mathcal{E}_{\text{pool}} \cup \mathcal{E}_{\text{queue}}$ 
     Coverage decision
11:    if  $|\mathcal{E}_{\text{reuse}}| \geq \kappa$  then
12:       $\mathcal{C}_a \leftarrow \text{TopK}(\mathcal{E}_{\text{reuse}})$ 
13:    else
14:       $\mathcal{C}_a \leftarrow \text{Retrieve}(a, g_i.\text{context})$ 
15:       $R_{\text{actual}} \leftarrow R_{\text{actual}} + 1$ 
16:       $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{C}_a$ 
     Forward propagation
17:    for  $j = i + 1, \dots, G$  do
18:      for each  $d \in \mathcal{C}_a$  do
19:        if  $\text{sim}(d, g_j) \geq \tau_p$  then
20:           $\mathcal{Q}_{g_j} \leftarrow \mathcal{Q}_{g_j} \cup \{d\}$ 
21:        end if
22:      end for
23:    end for
24:  end if
     NLI verification
25:    Classify each  $d \in \mathcal{C}_a$  against  $a$ 
26:     $\hat{l}(a) \leftarrow \text{AggregateNLI}(\mathcal{C}_a, a)$ 
27:  end for
28: end for
29: return  $\{\hat{l}(a)\}_{a \in \mathcal{A}_y}, R_{\text{actual}}$ 

```

tion 4.3. The pipeline processes segments sequentially from g_1 to g_G , maintaining a shared evidence pool $\mathcal{P}$ and per-segment evidence queues $\{\mathcal{Q}_{g_1}, \dots, \mathcal{Q}_{g_G}\}$.

For each atom a in segment g_i , the system first aggregates candidate evidence from two sources: (i) the shared pool $\mathcal{P}$, filtered by cosine similarity $\geq \tau_r$ to the atom embedding; and (ii) the segment’s pre-allocated queue $\mathcal{Q}_{g_i}$, populated by forward propagation from earlier segments. If at least κ relevant documents are found, external retrieval is skipped; otherwise, fresh evidence is retrieved.

After NLI verification (Eq. 9), any newly retrieved documents are added to the shared pool $\mathcal{P}$ and propagated forward: for each future segment g_j ($j > i$), if the cosine similarity between the document embedding and the segment embedding exceeds the propagation threshold τ_p , the document is pre-allocated to $\mathcal{Q}_{g_j}$.

Symbol	Description	Default
<i>Segmentation (Section 4.1)</i>		
w	Sliding window size (sentences)	3
δ	Sliding window stride	1
τ_s	Segmentation similarity threshold	0.75
$w_{\min}$	Minimum segment length (sentences)	2
$w_{\max}$	Maximum segment length (sentences)	8
$G_{\min}$	Minimum number of segments	2
<i>Evidence Pool & Propagation (Section 4.3)</i>		
τ_r	Pool relevance threshold	0.70
τ_p	Forward propagation threshold	0.70
κ	Minimum pool coverage to skip retrieval	2
k	Number of evidence documents per retrieval	5
<i>Retrieval Backend</i>		
–	Search API	Google (Serper)
–	Results per query (q)	5
<i>LLM Configuration</i>		
–	Backbone LLM	GPT-4o-mini
–	Temperature	0

Table 5: Complete list of hyperparameters and default values.

E Hyperparameter Details

Table 5 lists all hyperparameters used in SEGMENT-FACT along with their default values.

Relationship between τ_r and τ_p . The pool relevance threshold τ_r controls whether an existing document in the pool is considered relevant enough to an atom to count toward the coverage requirement κ . The propagation threshold τ_p controls whether a newly retrieved document is forwarded to a future segment’s evidence queue. In principle, these two thresholds serve different purposes: τ_r governs *atom-level* matching (document $\leftrightarrow$ atom), while τ_p governs *segment-level* matching (document $\leftrightarrow$ future segment). In practice, we set $\tau_p = \tau_r = 0.70$ in all experiments; preliminary tuning on a development set showed that decoupling the two thresholds provided no consistent benefit across benchmarks.

Segment size bounds $w_{\min}$ and $w_{\max}$. $w_{\min}$ denotes the minimum allowable segment length in sentences. Segments shorter than $w_{\min}$ after the initial greedy merging pass are absorbed into the adjacent segment (left or right) with the highest cosine similarity, preventing degenerate single-sentence segments that offer insufficient context for atom extraction. $w_{\max}$ denotes the maximum allowable

segment length. Segments exceeding $w_{\max}$ are iteratively split at the internal sentence boundary with the lowest consecutive cosine similarity, ensuring that each segment remains topically coherent. We set $[w_{\min}, w_{\max}] = [2, 8]$ in all experiments, which accommodates the typical paragraph length in biological and expository text.

Notation clarification: $g_i.\text{text}$ and $g_i.\text{context}$. In Eq. 6, $g_i.\text{text}$ denotes the raw text of segment g_i (i.e., the concatenation of its constituent sentences). $g_i.\text{context}$ provides minimal discourse context for coreference resolution during atomization: it consists of the original query x concatenated with the last sentence of the preceding segment g_{i-1} (if $i > 1$). For the first segment (g_1), $g_1.\text{context}$ contains only the query x . This context is included in the atomizer prompt so that the LLM can resolve pronouns and definite descriptions that refer to entities introduced in earlier segments, without requiring a separate decontextualization step.

F Prompt Templates

Below we provide the exact prompts used in each stage of the SEGMENTFACT pipeline. All prompts are issued to GPT-4o-mini with temperature set to 0.

F.1 Atom Extraction Prompt

The following prompt is used to extract atomic claims from each segment (Section 4.2). The `{{paragraph}}` placeholder is replaced with $g_i.\text{text}$.

Decompose the TEXT below into atomic facts. Each atomic fact is one short sentence stating a single verifiable claim. Keep pronouns as-is; do not add outside information.

TEXT:
`{{paragraph}}`

Return a JSON object like `{"id0": "...", "id1": "...", ...}` inside ````json ...```` fences. Only output the JSON, nothing else.

The extracted atoms are post-processed to remove: (i) empty or trivially short outputs (≤ 5 characters); (ii) prompt leakage artifacts (e.g., atoms that echo instruction keywords such as “atomic unit” or “should be extracted”); (iii) duplicate atoms (case-insensitive deduplication within each segment); and (iv) overly long outputs (> 300 characters), which typically indicate extraction of entire paragraphs rather than atomic claims.

F.2 NLI Verification Prompt

The following prompt is used for NLI-based verification (Section 4.3). Each atom–evidence pair is classified independently. The `{evidence}` placeholder is replaced with the evidence document text, and `{claim}` with the atom text together with its segment context.

You are a factual verification assistant. Given a piece of EVIDENCE and a CLAIM, classify the relationship between them as one of exactly three labels:

- "entailment": the evidence clearly supports the claim
- "contradiction": the evidence clearly contradicts the claim
- "neutral": the evidence is insufficient to determine whether the claim is true or false

EVIDENCE:
`{evidence}`

CLAIM:
`{claim}`

Output exactly one of: entailment, contradiction, neutral.

The NLI labels from all evidence documents for a given atom are aggregated using the rule in Eq. 9: the atom is *Supported* if at least one document entails it and none contradicts it; *Not Supported* if at least one contradicts and none entails; and *Uncertain* otherwise.

F.3 Decontextualization Prompt (Ablation Only)

The following prompt is used *only* in the controlled ablation experiment (Section 5.3, “Seg.+Dctx.” variant). It is not part of the default SEGMENTFACT pipeline.

Rewrite the following claim to be completely self-contained. Replace all pronouns, demonstratives, and ellipsis with their full referents from the context. Do NOT add any information not present in the original. Output ONLY the rewritten claim.

Context:
`{context}`

Original claim: `{claim}`

Self-contained claim:

G Embedding Model Details

We use all-MiniLM-L6-v2 from the sentence-transformers library as the sentence encoder for all embedding-based operations:

1. **Segmentation:** computing sliding window embeddings and consecutive sentence similarities (Section 4.1).
2. **Pool relevance check:** computing cosine similarity between atom embeddings and evidence document embeddings (Section 4.3).
3. **Forward propagation:** computing cosine similarity between evidence document embeddings and future segment embeddings (Section 4.3).

but also *improves verification quality* by providing richer cross-segment context.

This model produces 384-dimensional embeddings and has a maximum input length of 256 tokens. For evidence documents that exceed this length, we compute the embedding from the concatenation of the document title and the first 300 characters of the document body. The full document text is still used for NLI verification.

We chose all-MiniLM-L6-v2 for its balance of quality and speed (22M parameters, ~ 4 ms per sentence on GPU). We leave exploration of retrieval-specialized encoders (e.g., E5, BGE) to future work.

H Case study

Evidence accumulates across segments. Evidence retrieved for Segment 0 (the metal detector invention) includes Wikipedia’s article on Gerhard Fischer and a patent database entry. Both documents also contain information about Segments 3 (product applications) and 4 (other inventions). The evidence pool accumulates these documents after Segment 0, allowing 11 of the 29 atoms to be verified directly from pooled evidence without issuing any new retrieval calls.

Retrieval savings are substantial. SEGMENTFACT issues only 18 retrieval calls (vs. 29 atoms), a 37.9% reduction. In contrast, FactReasoner issues all 29 calls independently—including 3 queries that return the same Fischer Wikipedia article already retrieved in Segment 0.

Accuracy is improved, not degraded. SEGMENTFACT achieves 82.1% accuracy on this response (6 TP, 17 TN, 3 FP, 2 FN), compared to FactReasoner’s 75.0% (0 TP, 12 TN, 1 FP, 3 FN). Notably, FactReasoner fails to identify *any* supported claim as correct (0 TP), because its independent per-atom retrieval misses confirmatory evidence that is readily available in the pool. This demonstrates that evidence pooling not only saves cost