QESCHUNKER: A SINGLE OBJECTIVE UNIFIES OVERLAPPING AND NON-OVERLAPPING CHUNKING FOR RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

Chunking is a fundamental preprocessing step in retrieval-augmented generation (RAG), yet most existing chunkers rely primarily on heuristics. We introduce the Question–Evidence-Supervised Chunker (QESChunker), to our knowledge the first method for constructing static chunk indices from question–evidence supervision. QESChunker constructs document-specific chunk indices by optimizing over human-annotated or LLM-generated question–evidence pairs, without access to held-out test queries. We formulate non-overlapping and overlapping index construction through a unified question–evidence utility-maximization objective, turning chunking into a supervised combinatorial optimization problem. Under the non-overlap constraint, the objective decomposes over chunks and can be optimized exactly in polynomial time by dynamic programming. When overlap is allowed, the objective becomes monotone submodular; exact optimization is NP-hard, but our greedy achieves a $(1 - 1/\sqrt{e})$-approximation. Across two self-constructed and three public datasets, both variants outperform existing chunking methods in complete-evidence recall in nearly all evaluated settings, under the same overlap regime and a fixed hybrid retriever. These results recast chunking as supervised, budget-aware index construction for evidence retrieval.

1 INTRODUCTION

Retrieval-augmented generation (RAG) conditions a language model on passages retrieved from an external corpus (Lewis et al., 2020; Gao et al., 2023). Before any retrieval can occur, each document must be segmented into *chunks* that are embedded and indexed. Where the chunk boundaries fall largely determines whether the evidence for a question can be retrieved *completely*.

Despite its importance, chunking is overwhelmingly heuristic in practice. The baselines evaluated in this paper span fixed-length and fixed-overlap passages used by retrieval systems (Lewis et al., 2020; Guu et al., 2020; Karpukhin et al., 2020), natural paragraphs, lexical-cohesion segmentation with TextTiling (Hearst, 1997), embedding-based ClusterSemanticChunker (Smith & Troynikov, 2024), LLM-predicted boundaries with LumberChunker (Duarte et al., 2024), sentence-window retrieval (LlamaIndex, 2024b), and hierarchical Parent-Document and Auto-Merging retrieval (LlamaIndex, 2024a). These practical alternatives cover non-overlapping, overlapping, semantic, LLM-based, and hierarchical designs, but none optimizes an explicit question–evidence objective.

We cast chunking as an explicit optimization problem and show that a *single* objective, subjected to two different structural constraints, yields two algorithms with complementary guarantees. We call the resulting framework QESCHUNKER (**Q**uestion–**E**vidence **S**upervised Chunker): questions and their evidence blocks supervise where boundaries should fall, and the same objective is then solved exactly or near-optimally under each constraint. Figure 1 summarizes the framework.

Our contributions are:

1. **A unified objective.** We model a document as a sequence of atomic blocks and define a two-term objective comprising a coverage term and a structural term (§3).
2. **Two algorithms under different constraints.** Under a *non-overlap* constraint—that is, a contiguous partition into multiple chunks—the objective becomes additive over chunks,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

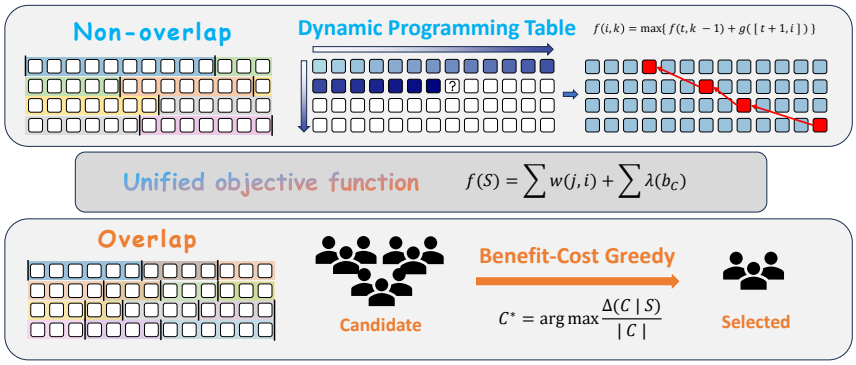

Figure 1: Overview of QESCHUNKER. A shared question–evidence objective supports exact dynamic programming for non-overlapping partitions and benefit–cost greedy selection for overlapping indices.

and an exact dynamic program returns the global optimum in polynomial time (§4). Under an *overlap* constraint, an index budget over a candidate chunk family must be satisfied. We prove that exact optimization of the objective is **NP-hard** by a reduction from Grid Segment Covering (§5), and give a benefit-cost greedy algorithm that attains a $(1 - 1/\sqrt{e}) \approx 0.393$ guarantee for budgeted monotone submodular maximization under the corrected analysis of Tang et al. (2021).

3. **Broad wins on multiple datasets.** Across two self-constructed Chinese policy corpora and three public English corpora (QASPER, PeerQA, and PrivacyQA), under a fixed hybrid retriever, our non-overlap DP ranks first in $9/10$ corpus$\times$tier display cells and our overlap method in $10/10$ (Tables 3 and 4).

2 Related Work

Chunking and adjacent segmentation. Our evaluated baselines are fixed-length, natural-paragraph, TextTiling, ClusterSemanticChunker, LumberChunker, fixed-window, Sentence-Window, Parent-Document, and Auto-Merging. Classical topic and discourse segmenters such as C99 and supervised neural boundary models optimize segmentation labels rather than RAG evidence recall (Choi, 2000; Koshorek et al., 2018; Lukasik et al., 2020). Other adjacent systems change more than chunk boundaries: Dense X replaces passages with generated propositions, Late Chunking changes contextual embedding computation, and RAPTOR introduces recursive summaries and hierarchical retrieval (Chen et al., 2024; Günther et al., 2024; Sarthi et al., 2024). We therefore exclude them from matched comparisons that isolate boundary selection under a fixed retriever and original-atom budget.

Submodular optimization. Our overlap solver draws on classical monotone-submodular maximization (Nemhauser et al., 1978; Krause & Golovin, 2014); the corresponding non-overlapping partition instead admits an exact dynamic program.

3 Problem Setup and Unified Objective

3.1 Atomic blocks, evidence, and hits

A document is atomized into a sequence $u_1, \ldots, u_n$ of *atomic blocks* by splitting on minimal punctuation; blocks are the common accounting unit for both indexing cost and returned context. Let $B \subseteq [n]$ be the set of block indices that terminate natural paragraphs. We are given m fitting questions. For question j:

- R_j is the set of gold evidence atomic blocks;
- H_j is the *hit set*: the top-κ atomic blocks most similar to the question embedding, i.e., the blocks that a first-stage retriever would actually surface.

Let S denote a set of chunks (either a partition or an overlapping set). A chunk $C = [a, b]$ covers blocks $a, \ldots, b$.

3.2 HIT-GATED WEIGHTED COVERAGE WITH A TIERED STRUCTURAL PRIOR

Treat each pair "question j, evidence block r" as an element to be covered, with weight

$$w(j, r) = \frac{1}{m\,|R_j|}, \qquad \sum_j \sum_{r \in R_j} w(j, r) = 1. \tag{1}$$

A single chunk C covers the element set

$$\Gamma(C) = \big\{(j, r) :\ r \in C \cap R_j,\ \ C \cap H_j \neq \varnothing\big\}. \tag{2}$$

The *hit gate* $C \cap H_j \neq \varnothing$ encodes retrieval reality: a chunk that contains none of question j's hit blocks will not be surfaced for j, so it cannot contribute coverage for j no matter how much gold text it holds.

The unified objective adds a tiered structural prior over chunk right-endpoints b_C:

$$\boxed{f(S) = \underbrace{\sum_{(j,r) \in \bigcup_{C \in S} \Gamma(C)} w(j, r)}_{\text{coverage term}} + \underbrace{\sum_{C \in S} \lambda(b_C)}_{\text{structural term}}} \tag{3}$$

where a chunk's structural term is determined by the terminal punctuation of its right block:

$$\lambda(b_C) = \lambda \times \begin{cases} \rho_1, & b_C \text{ ends a paragraph,} \\ \rho_2, & b_C \text{ ends a sentence but not a paragraph,} \\ \rho_3, & b_C \text{ ends a clause (semicolon or colon),} \\ \rho_4, & b_C \text{ ends at a comma or another position,} \end{cases} \qquad \rho_1 \geq \rho_2 \geq \rho_3 \geq \rho_4 \geq 0. \tag{4}$$

The parameter λ controls the contribution of the structural term. In this paper, the tier weights $\rho_1, \rho_2, \rho_3, \rho_4$ are set to 0.1, 0.005, 0.001, and 0, respectively.

Why hierarchical boundary rewards. Paragraph endings usually mark the strongest semantic boundaries; when a cut must fall inside a paragraph, sentence endings are preferable to clause endings, and clause endings are preferable to arbitrary positions. The tier weights encode this hierarchy while allowing question–evidence supervision to trade structural preference against evidence coverage.

3.3 PARTITION DECOMPOSITION

When S is a *partition*, its chunks are disjoint, so $\bigcup_{C \in S} \Gamma(C)$ is a disjoint union and f decomposes as a sum of per-chunk values $g(C)$—exactly the condition that makes dynamic programming valid (§4). When overlap is allowed this decomposition breaks and f must be treated as a set function, which is where submodularity earns its keep (§5). Both settings therefore share the same f and, under the evaluation convention of §6.1, directly comparable results.

4 SETTING 1: NON-OVERLAP PARTITION

Given the prior parameters K, $L_{\min}$, and $L_{\max}$, we seek a contiguous partition of $u_1, \ldots, u_n$ into exactly K chunks such that every chunk C satisfies $L_{\min} \leq |C| \leq L_{\max}$ or is admitted by the whole-paragraph exception below:

$$\max_S\ f(S) \quad \text{s.t.} \quad S \text{ is a contiguous } K\text{-partition satisfying the length rule.} \tag{5}$$

Whole-paragraph exception. Because a natural paragraph is often a semantic unit, an entire paragraph longer than $L_{\max}$ may form one chunk; oversized chunks that cross paragraph boundaries remain forbidden.

For every feasible chunk C, define its per-chunk utility as

$$g(C) = \sum_{(j,r)\in\Gamma(C)} w(j,r) + \lambda(b_C). \tag{6}$$

Let $f(i,k)$ be the maximum utility of a feasible partition of the prefix $u_1,\dots,u_i$ into exactly k chunks, with value $-\infty$ when no such partition exists. The sequential structure gives the recurrence

$$f(i,k) = \max_{t\in\mathcal{T}(i)} \Big\{ f(t,k-1) + g\big([t{+}1,i]\big) \Big\}, \qquad f(0,0) = 0, \tag{7}$$

where the predecessor set adjoins the whole-paragraph exception to the usual length window:

$$\mathcal{T}(i) = \underbrace{\{t:\ i - L_{\max} \le t \le i - L_{\min},\ t \ge 0\}}_{\text{length window}} \cup \underbrace{\big\{\mathrm{start}(i) - 1 \,\big|\, i \in B,\ i - \mathrm{start}(i) + 1 > L_{\max}\big\}}_{\text{whole-paragraph exception}}, \tag{8}$$

with $\mathrm{start}(i)$ the first block of the paragraph ending at i. The optimum is $f(n,K)$; cut points are recovered by back-pointers.

Time complexity analysis. Let $W = L_{\max} - L_{\min} + 1$. After interval gains are computed incrementally with block-to-question inverted lists, each DP state examines at most W length-feasible predecessors plus one whole-paragraph predecessor. The running time is therefore $O(nKW)$ and the table with back-pointers uses $O(nK)$ space.

Algorithm 1 Exact DP for non-overlapping chunking

```
Input: u_{1:n}, (R_j, H_j)_{j=1}^m, K, L_min, L_max, B
Initialize f(0,0) ← 0 and all other states to −∞.
for i = 1, ..., n do
    Form T(i) and compute g([t+1, i]) incrementally for t ∈ T(i).
    for k = 1, ..., K do
        Update f(i,k) by equation 7 and store its maximizing predecessor.
    end for
end for
return the partition obtained by backtracking from f(n,K).
```

5 Setting 2: Overlap Selection

Given a prescribed index budget B_{index}, which caps the total number of indexed atomic blocks, and prior length bounds $L_{\min}$ and $L_{\max}$, let

$$\mathcal{V} = \{[a,b] : 1 \le a \le b \le n,\ L_{\min} \le b - a + 1 \le L_{\max}\}$$

be the candidate family. Allowing overlap, the index is a set $S \subseteq \mathcal{V}$ rather than a partition, and we solve

$$\max_{S\subseteq\mathcal{V}} f(S) \quad \text{s.t.} \quad \sum_{C\in S} |C| \le B_{\text{index}}. \tag{9}$$

Here each selected chunk incurs cost $|C|$, and the structural prior is instantiated as a paragraph-end reward (the sentence and clause tiers are zero). Unlike the partition setting, the overlapping candidate family uses only the prescribed length bounds and does not invoke the whole-paragraph exception.

5.1 Monotone submodularity

Proposition 1 (Monotone submodularity)**.** *The set function f in equation 3 is monotone and submodular over chunks drawn from any candidate family.*

Proof sketch. The coverage term is a weighted coverage function: each element (j,r) contributes $w(j,r)$ once, the first time it is covered by a selected chunk. Weighted coverage is monotone submodular (Nemhauser et al., 1978). The prior term is non-negative and modular because each summand depends only on one chunk's right endpoint. Their sum is therefore monotone submodular. Appendix A gives the full proof. □

5.2 EXACT OPTIMIZATION IS NP-HARD

We set all structural-prior weights to zero, so the result applies directly to the coverage term of equation 3 and implies hardness of the full objective when the prior weights are allowed to be zero.

Theorem 1 (NP-hardness for contiguous chunks)**.** *Let the maximum chunk length L be part of the input and let*

$$\mathcal{V}_L = \{[a, b] : 1 \le a \le b \le n,\ b - a + 1 \le L\}.$$

Deciding whether there exists $S \subseteq \mathcal{V}_L$ with $\sum_{C \in S} |C| \le B_{\text{index}}$ and $f_{\text{cov}}(S) = 1$ is NP-complete, even when $|R_j| = 1$ and $|H_j| = 2$ for every question, the weights are $w(j, r) = 1/m$, and all structural-prior weights are zero. Consequently, exact maximization of the coverage-only objective over contiguous chunks is NP-hard. The same conclusion holds when $\mathcal{V}_L$ consists of all intervals of length exactly *L.*

Proof sketch. We reduce from GRID SEGMENT COVERING, which is NP-complete by Bergren et al. (2020, Theorem 7). Its input is a set of m vertical unit segments $e_j = [(x_j, y_j), (x_j, y_j + 1)]$ on an integer grid of side $q = O(m)$ and an integer k; a horizontal length-2 segment covers e_j if it contains either endpoint. We may assume $1 \le k \le m$.

Set $D = q+2m+3$ and encode (x, y) by $Dy+x+3$. The endpoints of e_j then become $\{p_j, p_j+D\}$, where $p_j = Dy_j+x_j+3$. Encoded points on distinct rows are more than 2 apart, so the grid instance is coverable by k horizontal length-2 segments exactly when k length-2 intervals on the line contain at least one point from every pair $\{p_j, p_j + D\}$. Such line intervals may be assumed to have integer left endpoints.

Let $p_{\max} = \max_j p_j$ and construct $n = p_{\max} + 2D + 2$ atomic blocks. For each j, define

$$R_j = \{p_j + D\}, \qquad H_j = \{p_j, p_j + 2D\}. \tag{10}$$

Give its sole evidence pair weight $1/m$, set all prior weights to zero, and choose $L = D + 3$ and $B_{\text{index}} = kL$. For a length-L chunk $C_t = [t, t + D + 2]$,

$$\begin{aligned}(j, p_j + D) \in \Gamma(C_t) &\iff t \in [p_j - 2, p_j] \cup [p_j + D - 2, p_j + D] \\ &\iff [t, t + 2] \cap \{p_j, p_j + D\} \neq \varnothing.\end{aligned}$$

Thus C_t covers question j exactly when the corresponding length-2 interval covers an encoded endpoint of e_j.

This correspondence maps every grid cover of size at most k to a feasible chunk family with $f_{\text{cov}} = 1$. Conversely, every chunk with nonempty coverage spans an evidence block and a hit block D positions apart, and hence costs at least $D + 1$. Since $(k + 1)(D + 1) - kL = D + 1 - 2k > 0$, a feasible solution contains at most k such chunks. Extending each to length L preserves coverage, after which the same correspondence yields a grid cover of size at most k. Appendix B gives both directions in full, including the normalization and boundary arguments.

The construction is polynomial because $D, L = O(m)$, $n = O(m^2)$, and $|\mathcal{V}_L| = O(nL) = O(m^3)$; feasibility and coverage are also polynomially verifiable. This proves NP-completeness. If only length-L chunks are allowed, the same construction applies and the budget directly limits the solution to k chunks, proving the final claim. □

5.3 A $(1 - 1/\sqrt{e})$ COST-BENEFIT GREEDY

By Proposition 1, f is monotone submodular, so equation 9 is budgeted monotone submodular maximization, for which the *cost-benefit greedy with a singleton safeguard* (MGREEDY) attains the following conservative guarantee under the corrected analysis of Tang et al. (2021).

Theorem 2 (Approximation guarantee (Tang et al., 2021))**.** *Let f be monotone submodular over $\mathcal{V}$ with $f(\varnothing) \ge 0$ and* $\mathrm{OPT} = \max\{f(S) : S \subseteq \mathcal{V},\ \sum_{C \in S} |C| \le B_{index}\}$*. Algorithm 2 returns S with*

$$f(S) \ \ge\ \left(1 - \tfrac{1}{\sqrt{e}}\right) \mathrm{OPT} \ \approx\ 0.393\, \mathrm{OPT}. \tag{11}$$

Algorithm 2 Lazy cost-benefit greedy with singleton safeguard

Input: $\mathcal{V}$, B_{index}, and objective f
$S \leftarrow \varnothing$; heapify $\mathcal{V}$ by $\Delta(C \mid S)/|C|$.
while the heap is non-empty and $\sum_{C \in S} |C| < B_{\text{index}}$ **do**
    Pop C and recompute $\Delta(C \mid S)/|C|$.
    if C is feasible, has positive gain, and remains heap-maximal **then**
        $S \leftarrow S \cup \{C\}$.
    else if C is feasible and $\Delta(C \mid S) > 0$ **then**
        Reinsert C with its updated density.
    else
        Discard C.
    end if
end while
$s^\star \leftarrow \arg\max_{C \in \mathcal{V}: |C| \le B_{\text{index}}} f(\{C\})$.
return $\arg\max_{T \in \{S, \{s^\star\}\}} f(T)$.

Tang et al. (2021) close a gap in the earlier analysis of Khuller et al. (1999) and prove the stronger 0.405 approximation factor for the same modified greedy algorithm; we retain the weaker $(1-1/\sqrt{e})$ bound here. Each round selects the interval with maximum marginal gain per unit cost,

$$C^\star = \arg\max_{C \in \mathcal{V}} \frac{\Delta(C \mid S)}{|C|}, \qquad \Delta(C \mid S) = f(S \cup \{C\}) - f(S), \tag{12}$$

and the *singleton safeguard* compares the greedy's result with the best single feasible chunk, returning the better. Reaching the $(1 - 1/e) \approx 0.632$ guarantee instead requires enumerating all initial subsets of size ≤ 3 (Sviridenko, 2004), which is $|\mathcal{V}|^3 \approx 10^{14}$ at our scale and infeasible; and for coverage objectives under cardinality constraints no polynomial algorithm can beat $(1 - 1/e)$ unless $\mathrm{P} = \mathrm{NP}$ (Feige, 1998), so the residual $\sqrt{e}$ gap is the price of tractability. *Lazy evaluation* makes the greedy practical: by submodularity a cached marginal gain is an upper bound on the current one, so a max-heap of stale densities needs only tens of exact re-evaluations per accepted chunk, and it returns exactly the same set as the eager cost-benefit greedy. Algorithm 2 summarizes the procedure.

In practice, the empirical gap between the algorithm's value and the optimum is generally much smaller than $1 - \dfrac{1}{\sqrt{e}}$.

6 Experiments

6.1 Experimental Setup

Datasets and supervision. Table 1 summarizes the corpora and supervision protocols.

Table 1: Dataset statistics and supervision.

Dataset	Type	Docs	Blocks	Paragraphs	Questions
Party Congress report	Self-built	1	2,842	147	642
Public-health white paper	Self-built	1	2,884	232	683
QASPER (Dasigi et al., 2021)	Public	185	103,024	12,548	532
PeerQA (Baumgärtner et al., 2025)	Public	70	43,656	6,222	136
PrivacyQA (Ravichander et al., 2019)	Public	34	4,931	4,931	1,276

Supervision protocol. Both self-built corpora use eight-fold cross-validation, with boundary fitting and testing separated in every fold. For each public document containing n atomic blocks, the `qwen3.8-max` API generates $n/5$ synthetic questions exclusively for boundary fitting; contiguous-evidence and cross-paragraph-evidence questions are generated in a 1:1 ratio. Evaluation then uses only the datasets' native public questions and evidence annotations. We compare against the non-overlap and overlap baseline families described in §2, using identical retrieval and returned-context budgets.

Automatic checks validate the generated fitting data. PrivacyQA treats every atom as a paragraph, so the natural-paragraph baseline is not applicable.

Evaluation convention. @k denotes complete-evidence recall when the deduplicated returned context contains k atomic blocks. @$k\%$ denotes the same metric when the context budget is $k\%$ of the document's atomic blocks.

All methods use the same embedding–BM25 retriever with section-title prefixes. Coverage is aggregated per question; self-built results are averaged over the eight folds. The parameter study below fixes operating points before the cross-corpus comparison.

6.2 Parameter Sweeps

We vary one factor at a time on the public-health white paper while holding the retriever, returned-atom budget, and eight folds fixed.

6.2.1 Structural Weight

Figure 2 shows complementary trends. For DP, coverage rises as small positive structural weights are introduced and then reaches a broad plateau around $\lambda = 10^{-2}$; both display budgets follow the same pattern. Greedy is stable near zero through small bonuses but declines once λ becomes large, sharply so at $\lambda \geq 10^{-2}$, indicating that an over-weighted paragraph prior displaces evidence-bearing chunks.

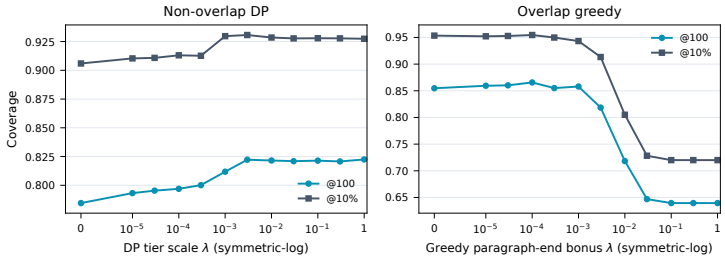

Figure 2: Coverage over the λ sweep at @100 and @10%.

6.2.2 Length Geometry

Figure 3 shows that both variants change smoothly as the feasible interval contracts rather than exhibiting a narrow optimum. DP is nearly flat over broad ranges and degrades only near the fixed-length limit; greedy is similarly stable at @10%, with a modest decline at @100. At the rightmost point $L_{\min} = L_{\max}$, so the admissible chunks are exactly those of equal-length splitting and the two settings are fully equivalent.

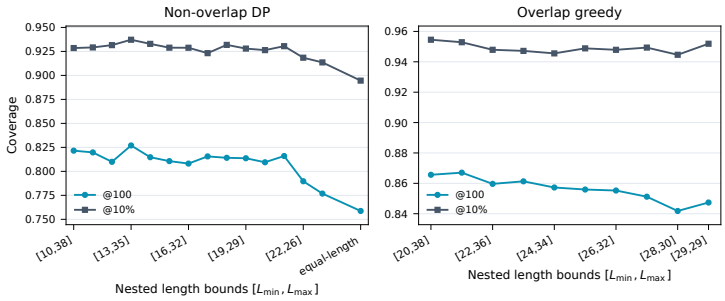

Figure 3: Coverage as candidate-length intervals contract.

6.2.3 Whole-Paragraph Exception

Table 2 shows that the whole-paragraph exception helps at small returned-context budgets but provides no advantage at large budgets. For fairness in the subsequent comparison, we enable the exception in every non-overlap main experiment.

Table 2: White-paper coverage with the whole-paragraph exception disabled or enabled; bold marks the better result.

	Tight budgets (atoms)					Larger budgets (atoms)					
Exception	50	58	100	144	200	288	305	433	577	721	865
Disabled	0.7076	0.7361	0.8142	0.8598	0.8972	**0.9310**	**0.9378**	**0.9595**	**0.9714**	**0.9828**	**0.9863**
Enabled	**0.7177**	**0.7490**	**0.8216**	**0.8631**	**0.8980**	0.9285	0.9340	0.9587	0.9703	0.9798	0.9853
Enabled − disabled (pp)	+1.01	+1.29	+0.73	+0.33	+0.08	−0.25	−0.38	−0.08	−0.12	−0.29	−0.10

6.3 Main Experiments

Based on §6.2, we fix all operating points before the cross-corpus comparison. For non-overlap, we set $\kappa = 5$, $\lambda = 0.01$, take $[L_{\min}, L_{\max}]$ as the 10th–90th percentiles of paragraph length, and set K to the number of paragraphs. For overlap, we set $\lambda = 10^{-4}$, use the same percentile-based length bounds, and set the total budget to three times the document length. We then compare non-overlapping methods under a common partition constraint and overlapping methods under the same returned-context convention.

Residual-budget concentration (overlap only). When the primary coverage objective saturates while index budget remains, we use

$$f_2(S) = \frac{1}{m} \sum_j \max_{C \in S,\, C \cap H_j \neq \varnothing} \frac{|C \cap R_j|}{|R_j|}, \tag{13}$$

and continue selection with $f + \delta f_2$, where $\delta = 0.05$, which favors candidates that concentrate more of a question's evidence in one retrievable chunk.

Table 3: **Non-overlap setting:** complete-evidence coverage at the two display tiers (@100 fixed atoms; @10% of document atoms). On PrivacyQA the natural-paragraph baseline is degenerate (one atomic block per paragraph) and is therefore marked "–".

	Congress report		White paper		QASPER		PeerQA		PrivacyQA	
Method	@100	@10%	@100	@10%	@100	@10%	@100	@10%	@100	@10%
QESChunker (exact DP)	**0.9025**	**0.9781**	**0.8216**	**0.9285**	**0.7588**	**0.6029**	0.7341	**0.6589**	**0.8687**	**0.3166**
Segment-by-paragraph	0.9020	0.9734	0.7818	0.8975	0.7425	0.5837	**0.7399**	0.6483	–	–
Equal-length split	0.8570	0.9669	0.7587	0.8946	0.6565	0.5018	0.6583	0.5424	0.8585	0.2975
TextTiling	0.8054	0.9424	0.7376	0.8801	0.6560	0.4842	0.6395	0.5604	0.8642	0.2956
ClusterSemanticChunker	0.8462	0.9494	0.7499	0.8871	0.6511	0.4540	0.6527	0.5357	0.8607	0.3039
LumberChunker (LLM)	0.8789	0.9694	0.7700	0.9024	0.6915	0.5064	0.6659	0.5406	0.8601	0.3001

Table 4: **Overlap setting:** complete-evidence coverage at the two display tiers, same corpora and convention as Table 3.

	Congress report		White paper		QASPER		PeerQA		PrivacyQA	
Method	@100	@10%	@100	@10%	@100	@10%	@100	@10%	@100	@10%
QESChunker (greedy)	**0.9165**	**0.9781**	**0.8656**	**0.9545**	**0.7412**	**0.5786**	**0.7399**	**0.6483**	**0.8747**	**0.3302**
Fixed window $L{=}bl$, ov. 20%	0.8708	0.9626	0.7266	0.8846	0.6807	0.5296	0.6939	0.5456	0.8581	0.3138
Fixed window $L{=}2bl$, ov. 20%	0.8778	0.9609	0.7889	0.9148	0.6554	0.4721	0.6901	0.5672	0.8686	0.2947
Sentence-Window ± 1	0.8312	0.9538	0.7469	0.8937	0.7086	0.5206	0.6862	0.5555	0.8515	0.2739
Sentence-Window ± 2	0.8464	0.9672	0.7733	0.9079	0.6830	0.5033	0.6600	0.5436	0.8535	0.2639
Parent-Document	0.8615	0.9579	0.7713	0.8909	0.7095	0.5238	0.6830	0.5595	0.8461	0.2945
Auto-Merging (4 sent.)	0.8232	0.9561	0.7559	0.8947	0.6801	0.4881	0.6590	0.5538	0.8509	0.2650

Non-overlap. The exact DP leads in $9/10$ displayed cells. Its only loss is PeerQA at @100, where the paragraph baseline is ahead by just 0.58 points and the evidence is often already paragraph-aligned. The largest gains occur on the white paper ($+3.98$ points at @100, $+2.61$ at @10%); nearties on the Congress report and PrivacyQA indicate less room when simple segmentation already matches the evidence geometry.

Overlap. Greedy leads in all $10/10$ cells, by 0.61–8.11 points over the strongest baseline. The largest gaps occur on the white paper at @100 and PeerQA at @10%; gains remain positive on PrivacyQA. Since all rows share the same overlap and returned-context conventions, the pattern reflects evidence-aware selection rather than unequal evaluation budgets.

6.4 Budget Analysis

Figure 4 traces the white-paper budget range; the other four datasets appear in Appendix C. Coverage rises monotonically for every method, but QESChunker separates at the smallest budgets and remains ahead in both settings. The gap is widest before the curves flatten, when only a small document fraction can be returned.

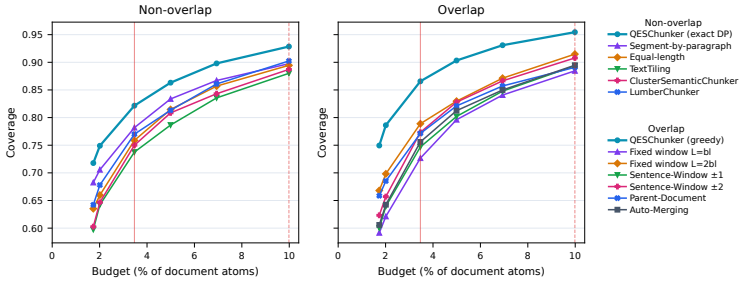

Figure 4: Coverage versus returned-atom budget on the white paper. Vertical lines mark @100 and @10%.

The appendix curves repeat this pattern with different strength: Congress methods converge near saturation; QASPER and PeerQA retain wider overlap-side gaps at low and medium budgets; and PrivacyQA has tightly clustered non-overlap curves but clear overlap separation. Thus more context helps every method while reducing the marginal value of optimized boundaries near the coverage ceiling.

7 Limitations

On one hand, our gains are measured at the retrieval level as complete-evidence recall. Providing more complete and effective evidence can plausibly improve downstream generation, but the rate at which recall gains translate into final answer quality remains unknown. This conversion depends on the generator, prompting, context ordering, evidence redundancy, and answer-level evaluation criteria; we therefore do not claim a fixed or proportional end-to-end improvement.

On the other hand, boundary fitting depends heavily on the quality and coverage of existing question–evidence supervision. It is thus best suited to documents that are queried repeatedly and have accumulated sufficient supervision—for example, policy reports, FAQs, or frequently discussed papers. For one-off, cold-start, or sparsely annotated documents, such supervision may be unreliable or unavailable, and performance may be driven mainly by the structural prior and the retriever rather than by learned boundaries.

8 Conclusion

QESChunker casts chunk construction as supervised, budget-aware optimization rather than heuristic preprocessing. A shared hit-gated coverage objective yields two complementary solvers: exact dynamic programming for non-overlapping partitions and benefit-cost submodular greedy for overlapping indices. Across five corpora, the two variants improve complete-evidence recall in nearly all displayed settings, with the largest gains appearing at tight returned-context budgets and smaller margins near saturation. The parameter sweeps further show that performance remains stable across broad structural-weight and candidate-length ranges. Together, these results indicate that question–evidence supervision can translate recurring retrieval demand into principled, document-specific static indices under either overlap regime.

AI USE STATEMENT

Generative AI was used to generate fitting questions and evidence IDs and to judge auxiliary outputs. Data passed legality, membership, length, hallucination, and duplication checks; public evaluation used only human or expert labels. Judging was anonymized, order-swapped, and conducted per fact. AI was not used for theoretical development or solver implementation. The authors reviewed all outputs and accept responsibility.

ETHICS STATEMENT

This work studies document segmentation for retrieval-augmented generation and uses publicly available scientific QA data (QASPER, PeerQA), a public privacy-policy QA corpus (PrivacyQA), and public government policy documents. It involves no human subjects, no private data, and no personal identifiers. The main dual-use consideration is that better evidence retrieval can improve both beneficial and harmful downstream generation; we mitigate fabrication risk in our own evaluation via hallucination screening, and we report negative and boundary results (scope, oracle ceiling, cost accounting) to avoid overstating capability. LLM-annotated evidence on the self-built corpora may inherit annotator-model biases; we flag this as a limitation and rely on human-annotated QASPER and author-annotated PeerQA for our highest-confidence claims.

REPRODUCIBILITY STATEMENT

The method is fully specified in §3–§5: the objective equation 3 with weights equation 1 and tier ratios equation 4, the DP recurrence and predecessor set equation 7–equation 8 with the whole-paragraph exception (Algorithm 1), the NP-hardness reduction (Theorem 1), and the greedy with its guarantee, lazy evaluation, singleton safeguard (Algorithm 2, Theorem 2). The residual-budget concentration term equation 13, evaluation convention, and sensitivity analyses are given in §6.3, §6.1, and §6.2, respectively. Correctness claims are backed by brute-force enumeration for the DP and by the certificate for the greedy.

REFERENCES

Tim Baumgärtner et al. PeerQA: A scientific question answering dataset from peer reviews. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

Dan Bergren, Eduard Eiben, Robert Ganian, and Iyad Kanj. On covering segments with unit intervals. In *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, volume 154 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 13:1–13:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.STACS.2020.13.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense x retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15159–15177. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.845.

Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000. URL https://aclanthology.org/A00-2004/.

Pradeep Dasigi, Kyle Liu, Ana Marasovic, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

André V. Duarte, João Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L. Oliveira. LumberChunker: Long-form narrative document segmentation. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6473–6486, 2024. doi: 10.18653/v1/2024.findings-emnlp.377.

Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. Late chunking: Contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. REALM: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning (ICML)*, 2020.

Marti A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Samir Khuller, Anna Moss, and Joseph (Seffi) Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. In *Proceedings of NAACL-HLT, Volume 2 (Short Papers)*, pp. 469–473. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-2075.

Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

LlamaIndex. Auto merging retriever. `https://docs.llamaindex.ai/`, 2024a. LlamaIndex documentation.

LlamaIndex. Metadata replacement and node sentence window. `https://docs.llamaindex.ai/`, 2024b. LlamaIndex documentation.

Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. Text segmentation by cross segment attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4707–4716. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.380.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

Abhilasha Ravichander, Alan W. Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. Question answering for privacy policies: Combining computational and legal perspectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4947–4958. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1500.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024.

Brandon Smith and Anton Troynikov. Evaluating chunking strategies for retrieval. Technical report, Chroma, July 2024. URL `https://www.trychroma.com/research/evaluating-chunking`.

Maxim Sviridenko. A note on submodular maximization with knapsack constraints. *Operations Research Letters*, 32(4):309–313, 2004.

Jing Tang, Xueyan Tang, Andrew Lim, Kai Han, Chongshou Li, and Junsong Yuan. Revisiting modified greedy algorithm for monotone submodular maximization with a knapsack constraint. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(1):1–22, 2021.

A PROOF OF PROPOSITION 1

Proof. Write $f(S) = f_{\text{cov}}(S) + f_{\text{pri}}(S)$ with $f_{\text{cov}}(S) = \sum_{(j,r)\in\bigcup_{C\in S}\Gamma(C)} w(j,r)$ and $f_{\text{pri}}(S) = \sum_{C\in S}\lambda(b_C)$.

f_{cov} is monotone submodular. Each element $e = (j,r)$ with weight $w(e) \geq 0$ is covered by S iff some $C \in S$ has $e \in \Gamma(C)$; thus $f_{\text{cov}}(S) = \sum_e w(e)\,\mathbf{1}[e \text{ covered by } S]$ is a weighted coverage function, which is monotone (adding chunks cannot uncover) and submodular (the marginal contribution of a chunk is the total weight of elements it covers for the first time, which is non-increasing as S grows).

f_{pri} is modular and non-negative. Each summand $\lambda(b_C) \geq 0$ depends only on the single chunk C; adding a chunk C to any S increases f_{pri} by exactly $\lambda(b_C)$, independent of S. A modular function is submodular with equality, and monotone since the increment is non-negative.

Sum. The sum of two monotone submodular functions is monotone submodular, so f is monotone submodular. The concentration term f_2 in equation 13 is a sum over j of a $\max$ over selected chunks of modular-in-C terms gated by $C \cap H_j \neq \varnothing$; a non-negative weighted $\max$ over selected sets is monotone submodular, so $f + \delta f_2$ with $\delta \geq 0$ preserves monotone submodularity. □

On the partition decomposition. If S is a partition, its chunks are pairwise disjoint, so no element (j,r) can be covered by two different chunks (each block r lies in exactly one chunk). Hence $\bigcup_{C\in S}\Gamma(C)$ is a disjoint union and $f_{\text{cov}}(S) = \sum_{C\in S} g_{\text{cov}}(C)$ with $g_{\text{cov}}(C) = \sum_{(j,r)\in\Gamma(C)} w(j,r)$. Together with the already-additive prior, $f(S) = \sum_{C\in S} g(C)$ is additive over chunks, which is precisely the optimal-substructure exploited by equation 7.

B FULL PROOF OF THEOREM 1

We use the construction and notation from the proof sketch.

Proof. **Forward implication.** Suppose the grid instance has a cover with at most k horizontal length-2 segments. After encoding the rows on the line, discard any interval containing no encoded endpoint. Shift every remaining interval rightward until its left endpoint is its smallest contained encoded point. This preserves all covered endpoints and gives an integer left endpoint. We therefore obtain at most k intervals $[t, t+2]$ that contain at least one point from every pair $\{p_j, p_j + D\}$.

Each normalized t is an encoded endpoint, so

$$3 \le t \le p_{\max} + D = n - L + 1.$$

Replace each $[t, t+2]$ by $C_t = [t, t+D+2]$. These are legal chunks, their total cost is at most $kL = B_{\text{index}}$, and the coverage correspondence gives $f_{\text{cov}}(S) = 1$.

Reverse implication. Suppose $S \subseteq \mathcal{V}_L$ satisfies the budget and $f_{\text{cov}}(S) = 1$. Remove chunks with $\Gamma(C) = \varnothing$, preserving coverage. Every remaining chunk covers some question j, so it contains $p_j + D$ and at least one point in $\{p_j, p_j + 2D\}$. These positions differ by D; hence each such chunk costs at least $D + 1$. Since $k \le m$ and $D = q + 2m + 3$,

$$(k+1)(D+1) - kL = D + 1 - 2k > 0.$$

Thus at most k nonempty-coverage chunks remain.

For each remaining chunk $[a, b]$, form $C_a = [a, a+D+2]$. This extension stays inside the document because the original chunk contains an evidence block, so

$$a \le p_j + D \le p_{\max} + D = n - L + 1.$$

Since $b - a + 1 \le L$, C_a contains the original chunk and preserves all its covered pairs. The resulting family has at most k length-L chunks and still covers every question. The coverage correspondence now shows that the intervals $[a, a+2]$ contain at least one point from every pair $\{p_j, p_j + D\}$. As encoded points on distinct rows are more than 2 apart, these intervals map back to at most k horizontal length-2 segments covering the grid instance.

Polynomial size and NP membership. Because $q = O(m)$, we have $D, L = O(m)$ and $n = O(m^2)$. The full candidate family has size $O(nL) = O(m^3)$, so the reduction is polynomial. A selected family is a polynomial-size certificate, and its cost and coverage are polynomially checkable. This proves NP-completeness.

If candidates are restricted to length exactly L, the forward direction is unchanged. In the reverse direction, the budget immediately permits at most k chunks, so the short-chunk argument is unnecessary. □

C Additional Budget Curves

We report the full returned-context budget curves for the four datasets not shown in Figure 4. Each panel follows the same non-overlap/overlap convention as the white-paper analysis.

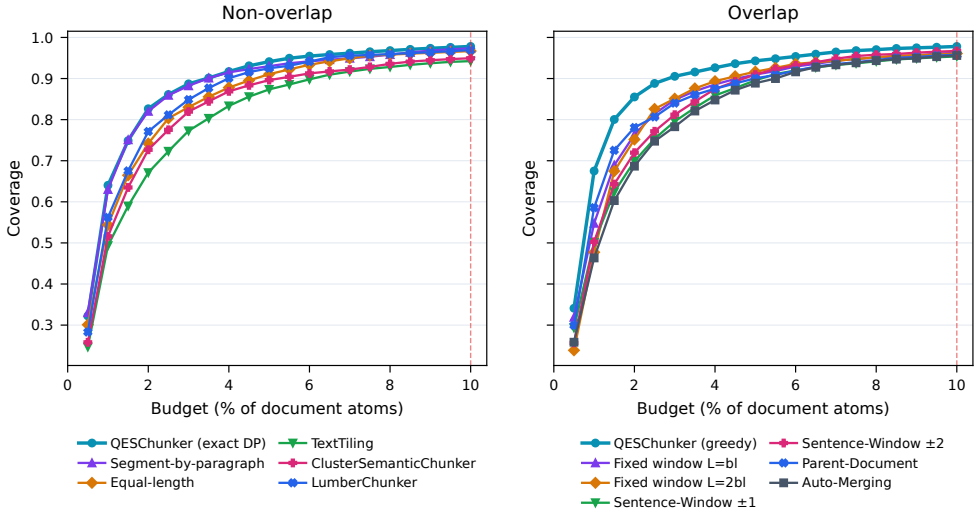

Figure 5: Budget curves on the Party Congress report. Left: non-overlap; right: overlap.

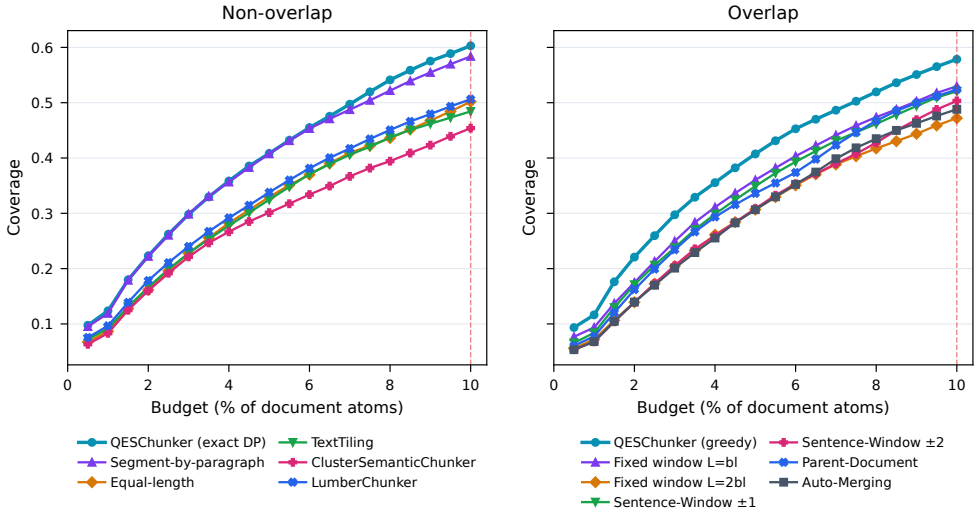

Figure 6: Budget curves on QASPER. Left: non-overlap; right: overlap.

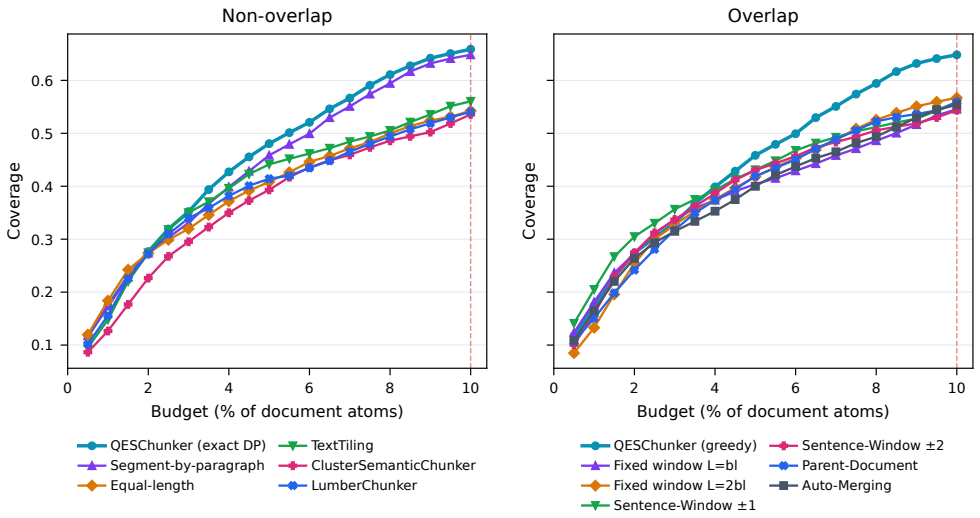

Figure 7: Budget curves on PeerQA. Left: non-overlap; right: overlap.

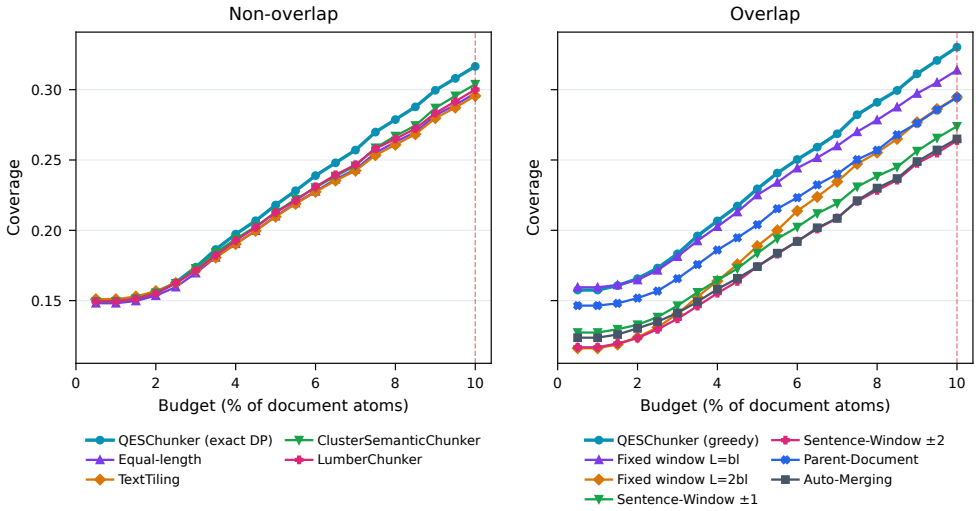

Figure 8: Budget curves on PrivacyQA. Left: non-overlap; right: overlap.