# Intent-Driven Dynamic Chunking: Segmenting Documents to Reflect Predicted Information Needs

Christos Koutsiaris
Department of Computer Science and Information Systems
University of Limerick, Ireland
24220094@studentmail.ul.ie

**Abstract**

Breaking long documents into smaller segments is a fundamental challenge in information retrieval. Whether for search engines, question-answering systems, or retrieval-augmented generation (RAG), effective segmentation determines how well systems can locate and return relevant information. However, traditional methods, such as fixed-length or coherence-based segmentation, ignore user intent, leading to chunks that split answers or contain irrelevant noise. We introduce Intent-Driven Dynamic Chunking (IDC), a novel approach that uses predicted user queries to guide document segmentation. IDC leverages a Large Language Model to generate likely user intents for a document and then employs a dynamic programming algorithm to find the globally optimal chunk boundaries. This represents a novel application of DP to intent-aware segmentation that avoids greedy pitfalls. We evaluated IDC on six diverse question-answering datasets, including news articles, Wikipedia, academic papers, and technical documentation. IDC outperformed traditional chunking strategies on five datasets, improving top-1 retrieval accuracy by 5% to 67%, and matched the best baseline on the sixth. Additionally, IDC produced 40–60% fewer chunks than baseline methods while achieving 93–100% answer coverage. These results demonstrate that aligning document structure with anticipated information needs significantly boosts retrieval performance, particularly for long and heterogeneous documents.

**Keywords:** Document Segmentation, Information Retrieval, User Intent, Question Answering, RAG, Dynamic Programming

**Code:** `https://github.com/unseen1980/IDC`

## 1 Introduction

Breaking long documents into well-chosen smaller segments is a fundamental preprocessing step in information retrieval systems. From search engines and question-answering applications to retrieval-augmented generation (RAG), documents must be split so that each segment can be efficiently indexed, retrieved, and presented to users or downstream models. In practice, nearly every modern retrieval system performs some form of document chunking. However, *how* these chunks are defined can greatly influence performance; even small changes in chunking strategy can noticeably affect retrieval recall and precision. Despite this impact, many implementations

treat chunking as a simplistic, ad-hoc procedure rather than as a core algorithmic component informed by end-user needs.

The most common approach, fixed-length chunking, divides text into uniform blocks (e.g., every 200 tokens). While simple, this method is arbitrary: it often cuts through sentences or logical topics, separating context from content. If the window is too small, a single answer can be fragmented across multiple chunks; if too large, each chunk may contain extraneous text that dilutes relevant information. Fixed segmentation is also highly sensitive to the chosen segment length; if not tuned carefully, retrieval quality drops markedly.

Coherence-based methods (e.g., TextTiling [Hearst, 1997], C99 [Choi, 2000]) improve on this by respecting discourse boundaries, keeping related ideas together. However, they remain **query-agnostic**: they optimize for internal document structure rather than the user's information need. A coherent section might still be too broad for a specific query, or an answer might span two coherent sections. This misalignment between document segments and user queries leads to suboptimal retrieval: relevant information may be buried in irrelevant text or fragmented across multiple chunks.

Existing solutions like document expansion (e.g., docT5query [Nogueira and Lin, 2019]) address vocabulary mismatch by adding predicted queries to text, but they do not alter the underlying segmentation. A retrieval system might still return chunks that contain answers mixed with unrelated content, simply because the document was segmented without regard to specific questions.

We propose **Intent-Driven Dynamic Chunking (IDC)**, a method that realigns document segmentation with user intent. IDC first predicts a set of likely user queries (intents) for a document using a generative model. It then employs a dynamic programming algorithm to segment the text such that each chunk optimally answers one of these predicted questions. By making segmentation intent-aware, IDC ensures that chunks are "answer-sized" and focused, containing complete, relevant information without excessive noise.

The motivation for IDC arose from real industrial challenges. In developing a semantic search system for SAP's Fiori technical documentation, we observed that basic chunking strategies held the system back. Engineers seeking specific answers (e.g., "How do I use API X?" or "What does error code Y mean?") often had to sift through multiple irrelevant chunks or piece together fragmented information. This disconnect between how documents were segmented and the questions users asked made search inefficient. IDC addresses this gap by anticipating user questions during segmentation.

The key contributions of this work are:

- We introduce IDC, a novel algorithm that adapts document segmentation to predicted user intents using dynamic programming optimization.
- We evaluate IDC on six QA benchmarks across four domains, showing that it improves Recall@1 on five datasets (with gains from 5% to 67%) and ties the best baseline on the sixth.
- We demonstrate that IDC produces 40–60% fewer chunks than baselines while achieving higher answer coverage (93–100%), making it efficient for indexing.

- We analyze the efficiency and cost of IDC, showing it adds minimal overhead suitable for offline indexing (<$0.01 per long document).

# 2 Related Work

## 2.1 Document Segmentation Methods

Document segmentation research spans several decades. Fixed-length chunking remains common due to simplicity, but early work showed its limitations: Callan [1994] found that fixed windows often divide answers between chunks. Wartena [2013] confirmed that retrieval performance "breaks down" quickly when segment length deviates from optimal values.

Coherence-based approaches emerged to address these issues. TextTiling [Hearst, 1997] detects topic shifts by analyzing lexical cohesion, placing boundaries at "valleys" of low similarity. C99 [Choi, 2000] clusters sentences by semantic similarity to identify topic boundaries. Barzilay and Lapata [2008] introduced entity-based coherence modeling for discourse understanding. More recently, Koshorek et al. [2018] framed segmentation as supervised learning with neural models, and Ghinassi et al. [2024] surveyed transformer-driven segmentation advances that leverage deep contextual embeddings.

While coherence-based methods produce internally consistent segments, they remain query-agnostic, optimizing for document structure without considering what users might ask. This motivates our intent-driven approach.

## 2.2 Query-Aware Document Expansion

Research in query-aware retrieval has largely focused on document expansion. The doc2query method [Nogueira et al., 2019a] predicts likely questions a document can answer and appends them to the text before indexing, bridging vocabulary gaps. Its successor docT5query [Nogueira and Lin, 2019] used the T5 transformer to generate more diverse, fluent questions with improved retrieval gains.

Subsequent work extended this paradigm. InPars [Bonifacio et al., 2022] used GPT-3 to create synthetic query-document pairs as training data for retrievers. Promptagator [Dai et al., 2023] demonstrated that prompting large language models can yield useful query variations with minimal examples.

However, these expansion methods do not alter document segmentation. The added queries become part of the text in each document's index entry, but the underlying splitting remains unchanged. If important information is split across chunks due to suboptimal segmentation, appending questions cannot fix that fragmentation. IDC extends the intuition of query prediction from expansion to *structure*, using predicted queries not just to enrich content, but to drive how the document is segmented.

# 3 Methodology

## 3.1 Overview of IDC

Intent-Driven Dynamic Chunking realigns document segmentation with user information needs through two main offline stages: (1) *Intent Simulation*, where likely user queries are predicted for the document, and (2) *Boundary Optimization*, where the document is segmented to maximize alignment between chunks and these predicted intents.

## 3.2 Intent Simulation

We generate a set of hypothetical user intents $Q = \{q_1, q_2, \ldots, q_M\}$ for document $D$ using Gemini 2.5 Flash. The LLM is prompted to generate questions the document can answer, covering its main topics and key details. To ensure topic coverage, we employ section-wise generation for longer documents and use stochastic decoding (top-$k$ sampling) for diversity.

The number of generated intents adapts to document complexity: short documents (<100 sentences) receive 10–15 questions, while long documents (>400 sentences) receive 35–40 questions. This adaptive strategy ensures adequate coverage without over-segmentation. After generation, we filter redundant questions by computing cosine similarity between their embeddings; if two questions exceed a similarity threshold (0.85), we retain only one.

## 3.3 Sentence Embedding and Scoring

The document is split into $N$ sentences $S = \{s_1, s_2, \ldots, s_N\}$. Both sentences and predicted intents are encoded into a shared vector space using a transformer-based sentence embedding model (1536-dimensional embeddings). For a candidate chunk $C_{i,j}$ spanning sentences $i$ to $j$, the chunk embedding is computed as the average of its constituent sentence embeddings. The *intent relevance* score is:

$$R(C_{i,j}) = \max_{q \in Q} \cos(\mathbf{e}(C_{i,j}), \mathbf{e}(q)) \tag{1}$$

where $\mathbf{e}(\cdot)$ denotes the embedding function. $R(C_{i,j})$ quantifies how well the chunk could answer at least one predicted question.

## 3.4 Boundary Optimization

We find segmentation $\mathcal{S} = \{C_1, C_2, \ldots, C_k\}$ that maximizes the utility function:

$$U(\mathcal{S}) = \sum_{m=1}^{k} R(C_m) - \lambda \sum_{m=1}^{k} |C_m|^2 - \beta(k-1) \tag{2}$$

where $\lambda$ is a length penalty (discouraging overly long chunks) and $\beta$ is a boundary penalty (discouraging over-segmentation). Because $|C_m|^2$ grows quickly with chunk size, $\lambda$ is typically very small (e.g., 0.0005 after tuning) to allow context-rich chunks without excessive penalty.

We solve this efficiently using dynamic programming. Let $f(j)$ be the maximum utility for

optimally segmenting sentences 1 through $j$. The recurrence is:

$$f(j) = \max_{0 \le i < j} \{f(i) + R(C_{i+1,j}) - \lambda |C_{i+1,j}|^2 - \beta\} \tag{3}$$

with $f(0) = 0$. We only consider chunks within a maximum length $L$ (e.g., 10–15 sentences), reducing complexity to $O(N \times L)$, which is essentially linear in document length.

After the DP solution, we apply light post-processing: merging very short adjacent chunks with the same intent, and splitting overly long chunks at natural paragraph boundaries if needed.

# 4 Experimental Setup

## 4.1 Datasets

We evaluated IDC on six question-answering datasets spanning four domains (Table 1): news articles (NewsQA), Wikipedia (SQuAD), academic papers (arXiv, Qasper), and technical documentation (Fiori). These datasets vary in length (12–495 sentences) and structure, providing a comprehensive evaluation across document types.

Table 1: Dataset characteristics

| **Dataset** | **Domain** | **Docs** | **QA Pairs** |
|---|---|---|---|
| NewsQA | News | 1 | 15 |
| SQuAD 1-doc | Wikipedia | 1 | 12 |
| SQuAD 2-doc | Wikipedia | 2 | 293 |
| arXiv | Academic | 1 | 15 |
| Qasper | Academic | 10 | 10 |
| Fiori | Technical | 1 | 15 |

## 4.2 Baselines

We compared IDC against four baseline segmentation strategies:

- **Fixed-Length**: Non-overlapping 6-sentence chunks
- **Sliding Window**: 6-sentence chunks with 50% overlap
- **Coherence-Based**: TextTiling-like topic boundary detection
- **Paragraph-Based**: Natural paragraph breaks as boundaries

All methods used identical preprocessing (sentence tokenization), embedding models, and hybrid retrieval (60% dense + 40% BM25).

## 4.3 Evaluation Metrics

We used Recall@1 (R@1), Recall@5 (R@5), and Mean Reciprocal Rank (MRR). R@1 measures the fraction of queries where the top-ranked chunk contains the answer, which is critical for QA systems. We also report chunk counts and answer coverage (percentage of answers fully contained within single chunks).

# 5 Results

## 5.1 Retrieval Performance

Table 2 presents the main retrieval results. IDC achieved the highest R@1 on five of six datasets and tied on the sixth (Qasper).

Table 2: Retrieval Performance (Recall@1, Recall@5, MRR)

| Dataset / Method | R@1 | R@5 | MRR |
|---|---|---|---|
| *NewsQA* | | | |
| IDC | **0.933** | **1.000** | **0.956** |
| Best Baseline | 0.867 | 0.867 | 0.867 |
| *SQuAD 1-doc* | | | |
| IDC | **0.917** | **1.000** | **0.958** |
| Best Baseline | 0.917 | 0.917 | 0.917 |
| *arXiv (495 sentences)* | | | |
| IDC | **0.667** | **0.933** | **0.789** |
| Best Baseline | 0.400 | 0.800 | 0.530 |
| *Fiori* | | | |
| IDC | **0.533** | **0.933** | **0.686** |
| Best Baseline | 0.333 | 0.733 | 0.502 |
| *SQuAD 2-doc (n=293)* | | | |
| IDC | **0.689** | **0.952** | **0.793** |
| Best Baseline | 0.655 | 0.951 | 0.752 |
| *Qasper* | | | |
| IDC | 0.250 | 0.500 | 0.333 |
| Best Baseline | **0.250** | **0.600** | **0.367** |

IDC's improvements were most pronounced on long, heterogeneous documents. On the 495-sentence arXiv paper, IDC achieved R@1 of 0.667 versus 0.400 for baselines, a **67% relative improvement**. On Fiori technical documentation, IDC reached 0.533 versus 0.333 (+**60%**). On the large SQuAD 2-doc dataset (293 queries), IDC's improvement was statistically significant ($p < 0.05$, Cohen's $d \approx 0.41$).

The Qasper dataset was an exception: IDC tied with the Paragraph baseline on R@1 (0.250) but showed slightly lower R@5 (0.500 vs 0.600) and MRR (0.333 vs 0.367). This suggests that for highly structured academic papers where each section naturally aligns with specific questions, paragraph-based segmentation can be equally effective. The structured nature of research papers, with clear section boundaries corresponding to distinct topics, provides natural "intent alignment" that IDC cannot significantly improve upon.

## 5.2 Segmentation Efficiency

IDC produced significantly fewer chunks than baselines while achieving better retrieval (Figure 3). On arXiv, IDC created 39 chunks versus 83 for Fixed-length (53% reduction). On Fiori, IDC produced 177 chunks versus 304 for Fixed (42% reduction). Fewer chunks means smaller

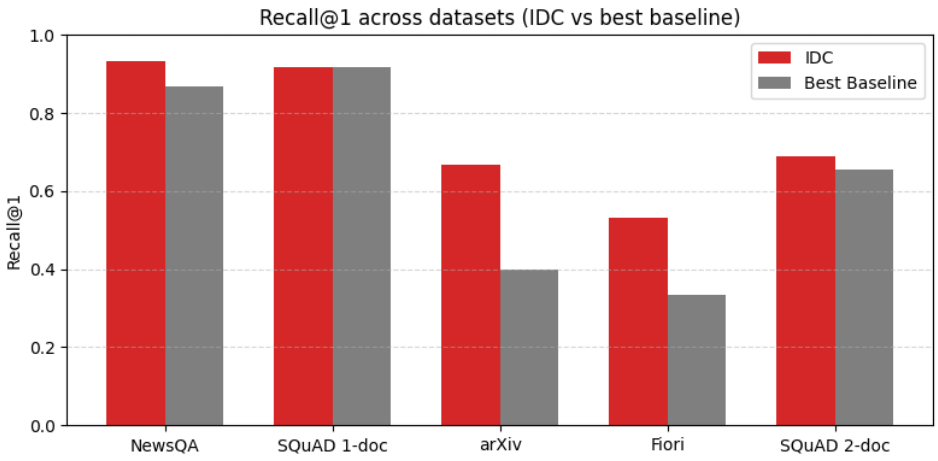


Figure 1: Recall@1 across datasets. IDC (red) consistently matches or exceeds the best baseline (gray), with largest gains on long documents (arXiv +67%, Fiori +60%).

index sizes and faster retrieval.

Despite fewer chunks, IDC achieved higher answer coverage (Figure 4). On arXiv, IDC covered 93.3% of answers within single chunks, compared to 80% for Fixed. On Fiori, IDC achieved 100% coverage versus 86.7% for baselines. This demonstrates that IDC's intent-guided boundaries place cuts more intelligently, keeping complete answers intact.

### 5.3 Efficiency Analysis

**Offline Preprocessing:** IDC takes 1–2 seconds per short document and 10–15 seconds for very long documents (>400 sentences). Intent generation dominates this cost (∼1s via Gemini 2.5 Flash API), while DP segmentation is fast (<200ms).

**Query Latency:** Online retrieval is identical for IDC and baselines (∼500ms, dominated by query embedding and index lookup). IDC's preprocessing is entirely offline.

**Cost Analysis:** Using Gemini 2.5 Flash pricing, costs vary by document length:

- **Short documents** (<100 sentences): ∼$0.0002–0.0005 per document
- **Long documents** (400+ sentences, ∼15k tokens): ∼$0.002–0.005 per document

For a corpus of 1,000 documents, total preprocessing cost ranges from $0.20 (short docs) to $5.00 (long docs). Note that for large-scale processing, API rate limits may become a bottleneck; costs assume parallelization is feasible.

## 6 Discussion

**Why IDC Works:** IDC's improvements stem from aligning chunk boundaries with likely information needs. By predicting questions users might ask, IDC creates "answer-sized" segments that contain complete, focused content. This contrasts with fixed-length chunking (which arbitrarily fragments information) and coherence-based methods (which optimize for topical consistency but not query relevance).

**When IDC Excels:** The largest gains occur on long, heterogeneous documents where static segmentation struggles. Technical manuals (Fiori +60%), academic papers with diverse sections (arXiv +67%), and multi-document collections (SQuAD 2-doc +5%) all benefit substantially. In these cases, IDC's dynamic chunk sizing (larger for broad explanations, smaller for specific facts) outperforms uniform approaches.

**When IDC Ties Baselines:** On well-structured documents like Qasper academic papers, paragraph boundaries naturally align with distinct topics and questions. Here, simple paragraph-based segmentation achieves comparable results. IDC provides no advantage when document structure already reflects likely query boundaries.

**Limitations:** IDC depends on LLM-generated intents; if the model fails to predict relevant questions, segmentation quality suffers. Some datasets had small sample sizes (n=15), limiting statistical power. Additionally, IDC's offline processing adds indexing time, though this is acceptable for most applications.

## 7 Conclusion

We introduced Intent-Driven Dynamic Chunking (IDC), a novel approach that segments documents based on predicted user intents. By generating likely questions via an LLM and optimizing chunk boundaries through dynamic programming, IDC produces segments aligned with actual information needs. Evaluation across six diverse QA datasets showed that IDC outperformed traditional chunking methods on five datasets, with R@1 improvements ranging from 5% to 67%, while producing 40–60% fewer chunks with higher answer coverage.

IDC is particularly effective for long, heterogeneous documents where static segmentation fails to isolate relevant content. The approach adds minimal computational overhead suitable for offline indexing, with no impact on query-time latency.

Future work includes extending IDC to multi-hop queries requiring information synthesis across chunks, incorporating real user feedback for adaptive re-segmentation, and exploring domain-specialized intent generation for technical corpora.

## References


Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. InPars: Data augmentation for information retrieval using large language models. In *Proceedings of SIGIR*, pages 2622–2631, 2022.

James Callan. Passage-level evidence in document retrieval. In *Proceedings of SIGIR*, pages 302–310, 1994.

Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of NAACL*, pages 26–33, 2000.

Zhuyun Dai, Vincent Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *Proceedings of ICLR*, 2023.

Iacopo Ghinassi, Lin Wang, Chris Sherwood Newell, and Matthew Purver. Recent trends in linear text segmentation: A survey. In *Findings of EMNLP*, pages 3084–3095, 2024.

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

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. In *Proceedings of NAACL*, pages 469–473, 2018.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019a.

Rodrigo Nogueira and Jimmy Lin. From doc2query to docT5query. *arXiv preprint arXiv:1910.14424*, 2019.

Christian Wartena. Segmentation strategies for passage retrieval from Internet video using speech transcripts. *Journal of Digital Information Management*, 11(6):399–407, 2013.

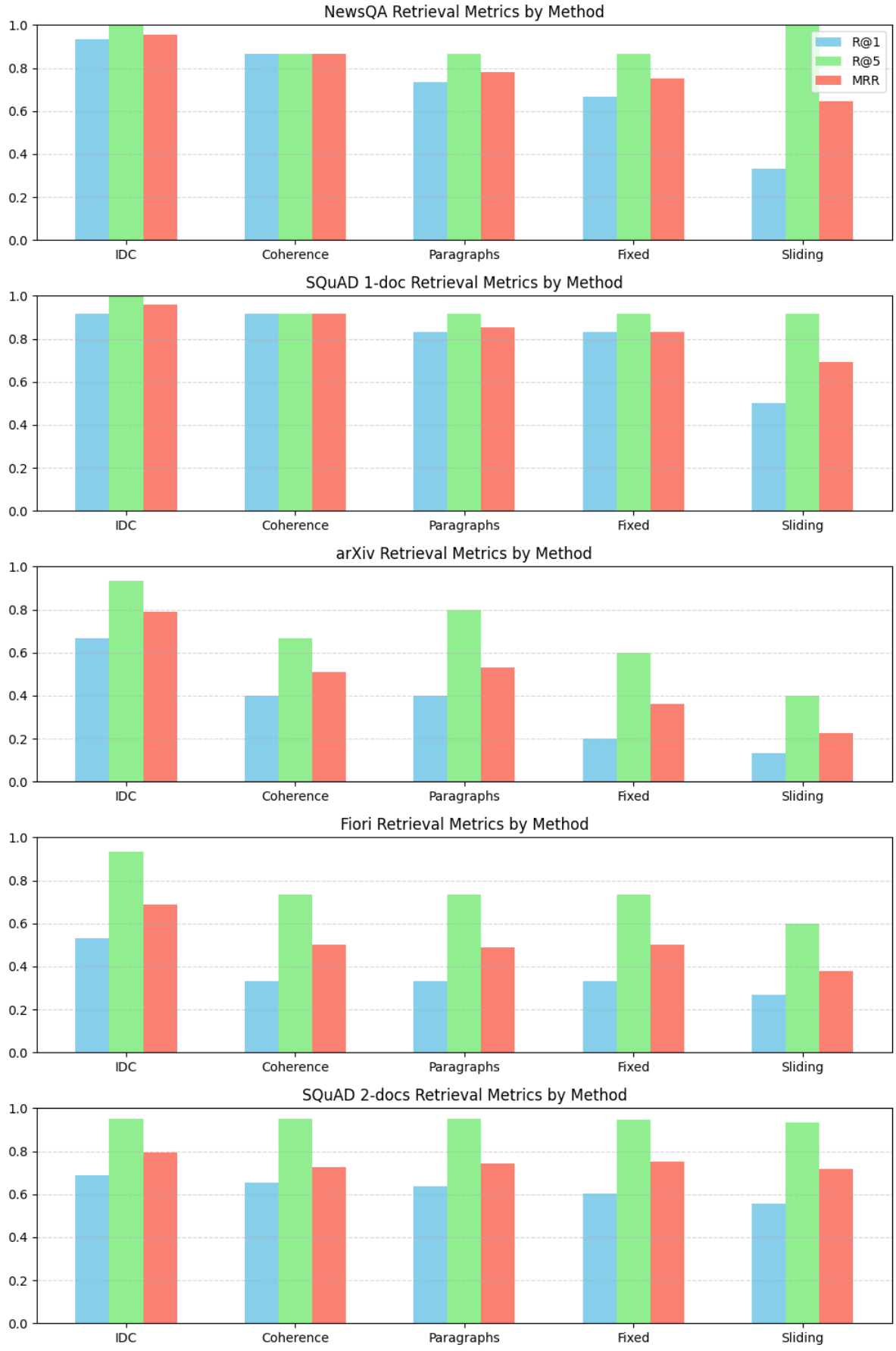


Figure 2: Complete retrieval metrics (R@1, R@5, MRR) across all datasets and methods. IDC achieves the highest or tied-highest scores on 5 of 6 datasets.

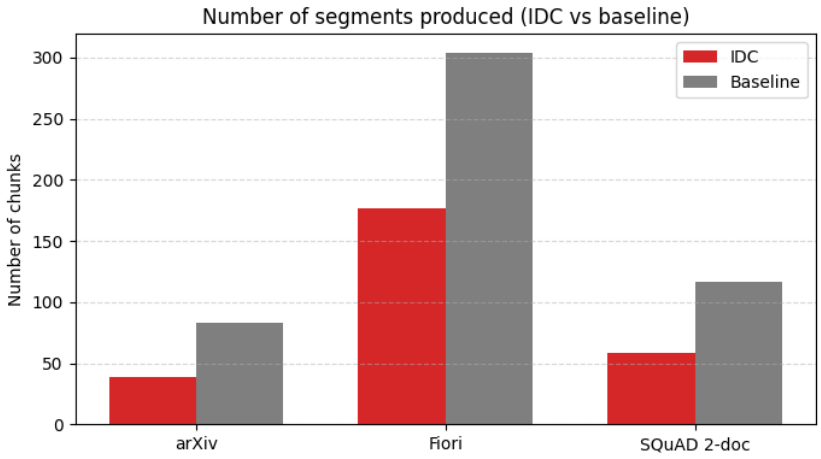


Figure 3: Number of chunks produced by IDC vs baselines. IDC generates 40–60% fewer chunks while achieving higher retrieval performance.

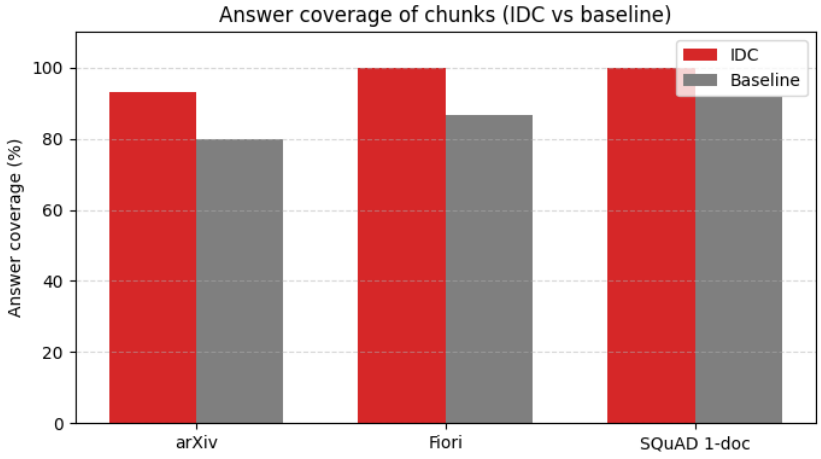


Figure 4: Answer coverage: percentage of questions whose answer is fully contained within a single chunk. IDC achieves 93–100% coverage, compared to 80–87% for baselines.