# WattBot 2025: Retrieval-Augmented Generation for ESG Intelligence Term Project of Natural Language Processing

**Shao-Hua Wu[‡], Pei-Ju Hsieh[‡], Bo-Hao Chen[‡], Yi-Chen Hsiao[*], and Yi-Yang Xue[†]**

[*] Department of Finance, National Chengchi University,
[†] Department of Computer Science, National Chengchi University,
[‡] Department of Money and Banking, National Chengchi University

## Abstract

We introduce HERO (Hybrid Evidence Retrieval Optimization), a retrieval-augmented generation (RAG) system developed for the WattBot 2025 shared task on extracting verifiable ESG and energy-use information from technical papers. Starting from a BM25 baseline, we identify a "table paradox": the very tables that contain crucial numerical evidence systematically degrade sparse retrieval because of structural noise. HERO addresses this with visual-aware document ingestion (Docling + OCR), dual sparse–dense indexing with weighted reciprocal-rank fusion that boosts table and OCR chunks, CrossEncoder reranking, and type-specific two-stage generation. Experiments on the competition corpus demonstrate that this structure- and modality-aware design yields consistent gains in both retrieval and answer quality across multiple LLM backends. The results highlight that for ESG-focused information extraction, retrieval architecture plays a comparably critical role to model selection.

## 1 Introduction

Large language models (LLMs) such as GPTs and LLaMA (Touvron et al., 2023) have demonstrated impressive capabilities in open-domain question answering, reasoning, and text generation. However, these models inherently rely on the static knowledge encoded during pretraining, making it challenging to access facts or developments emerging after their training cutoff. While post-hoc methods such as supervised fine-tuning and instruction tuning can inject task-specific knowledge or improve generalization, they are fundamentally constrained by the availability, cost, and timeliness of curated training data.

To overcome these limitations, the RAG paradigm has emerged as a compelling solution(Lewis et al., 2020). By equipping LLMs with the ability to retrieve and condition on external corpora (Guu et al., 2020; Karpukhin et al., 2020), RAG systems allow models to generate responses grounded in external evidence, extending beyond the information encoded during pretraining. However, two persistent challenges remain: retrieval modules often miss fine-grained aspects of complex queries, and generated answers may outpace their supporting evidence, especially under tight output constraints.

In this technical report, we present HERO, which is a RAG system developed for the WattBot 2025 shared task on extracting verifiable energy and emissions data from technical papers. The task requires answering numeric, categorical, and unanswerable sustainability queries from a small but heterogeneous corpus where key evidence often appears in tables and figures.

Starting from a strong BM25 baseline(Robertson and Zaragoza, 2009), we identify a table paradox: tables contain most gold answers yet consistently degrade sparse-retrieval quality due to structural noise(Wei et al., 2006). HERO resolves this with a visual-aware ingestion layer (Docling + EasyOCR), dual sparse–dense indexing (BM25 + BGE-M3), weighted reciprocal-rank fusion that emphasizes table/OCR chunks, CrossEncoder reranking, and type-specific, two-stage generation. On the competition data, HERO improves retrieval metrics over the baseline (e.g., Recall@10 from 92.31% to 95.12%) and raises the leaderboard score from 0.807 to 0.821 with Gemini 2.5 Pro, and up to 0.886 with Gemini 3.0 Pro. Our analysis shows that

LLMs remain bottlenecked by retrieval quality, and that structure- and modality-aware retrieval design is crucial for ESG intelligence over technical documents.

### 1.1 Task description

The WattBot 2025 challenge aims to address the lack of reliable data on the environmental impacts of AI systems—such as emissions, energy, and water use—by having participants develop RAG systems that extract credible and citation-supported estimates from technical literature. The competition emphasizes transparent, evidence-based answers to sustainability questions, with a key challenge lying in the balance between retrieval accuracy and generative precision, while accounting for gaps in available evidence.

### 1.2 Corpus Characteristics

The competition provides a curated corpus of over 30 scholarly articles spanning machine learning, energy systems, and environmental science. Each document is catalogued in `metadata.csv` with unique identifiers, full citations, and access URLs. The training set (`train_QA.csv`) contains example queries paired with gold-standard answers, reference IDs, and supporting evidence such as verbatim excerpts, table indices, and figure citations. Test questions (`test_Q.csv`) cover three answer types: numeric values (e.g., $CO_2$ emissions in pounds), categorical terms (e.g., "Water consumption"), and binary True/False assertions. A subset of questions intentionally requires multi-document reasoning, visual interpretation via optical character recognition (OCR), or explicit recognition of unanswerable queries.

### 1.3 Evaluation Protocol

Submissions are evaluated using a custom WattBot Score that weights three components: answer_value (0.75), ref_id (0.15), and is_NA (0.10). Numeric answers must satisfy $\pm 0.1\%$ relative tolerance; categorical values require exact matches after normalization. Citation credit is computed via Jaccard overlap between predicted and ground-truth reference sets. For unanswerable questions, systems must emit is_blank across all fields (answer_value, answer_unit, ref_id, ref_url, supporting_materials). Submissions must follow the structured format of train_QA.csv, including natural-language answers, normalized values, units, citations, supporting materials, and explanations connecting evidence to conclusions.

## 2 Related Works

To guide our WattBot 2025 strategy, we analyzed winning Kaggle solutions—especially KohakuRAG, which stood out for its innovative architecture. KohakuRAG challenged several standard RAG assumptions and provided concrete design patterns for handling heterogeneous technical documents. Its emphasis on structured parsing and hybrid retrieval offered key insights that directly informed the refinement of our own approach.

### 2.1 KohakuRAG: Hierarchical Indexing Architecture

The core innovation of KohakuRAG lies in its hierarchical document representation. Rather than treating documents as flat sequences of fixed-size chunks, the system parses each document into a semantic tree structure: Document $\rightarrow$ Section $\rightarrow$ Paragraph $\rightarrow$ Sentence. This hierarchy preserves the logical structure of academic papers and addresses the context fragmentation problem inherent in traditional chunking approaches.

The embedding strategy reflects this structure. Leaf nodes (sentences) are embedded using jina-embeddings-v3 with 1024-dimensional vectors for precise matching. Parent nodes (paragraphs and sections) receive averaged embeddings of their children, creating semantic centroids that capture broader context. During retrieval, queries match against sentence-level embeddings for precision, but the system automatically expands to retrieve the full parent paragraph, providing the LLM with coherent context rather than isolated fragments.

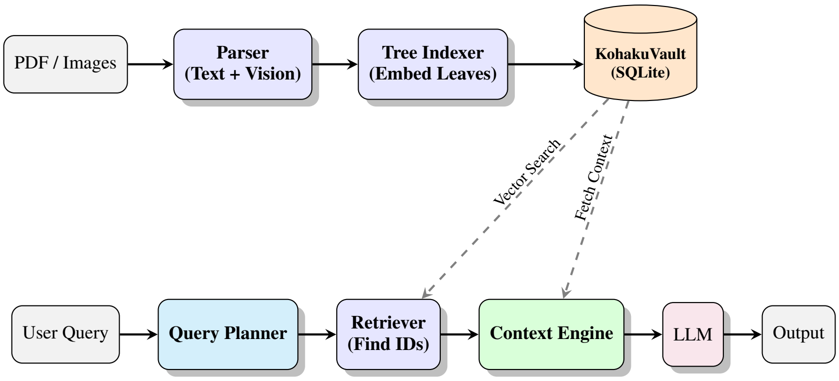

Figure 1: KohakuRAG Pipeline Architecture: Offline indexing creates hierarchical embeddings stored in SQLite; online processing uses query planning, vector retrieval, and context expansion.

### 2.2 Hybrid Retrieval Pipeline

KohakuRAG implements a sophisticated three-stage retrieval process. The first stage employs dense retrieval using Faiss with IVF-PQ indexing for semantic matching. The second stage applies BM25 sparse retrieval to capture exact term matches critical for technical queries containing specific model names, numerical values, or domain acronyms. The third stage uses a CrossEncoder model (bge-reranker-large) for reranking. Unlike the bi-encoders in stages one and two, the CrossEncoder processes query-document pairs jointly through cross-attention mechanisms, enabling deeper relevance assessment at the cost of higher computational complexity.

Figure 1 illustrates the complete KohakuRAG architecture, showing the separation between offline indexing and online query processing.

### 2.3 Engineering Design Choices

KohakuRAG builds on SQLite rather than specialized vector databases like Pinecone or Milvus. For the competition's 32-paper corpus, exact nearest neighbor search completes in milliseconds, making approximate methods unnecessary. The SQLite-based architecture (via KohakuVault) stores the entire knowledge base—text, embeddings, metadata, and image captions—in a single .db file, ensuring reproducibility and eliminating deployment complexity.

The system also demonstrates production-grade engineering through extensive use of Python's asyncio for parallelizing API calls, exponential backoff for rate limit handling, and comprehensive caching of embeddings and LLM responses. These practices significantly reduce development iteration time and API costs.

### 2.4 Multimodal Processing

Academic papers frequently present critical information exclusively in charts and figures. KohakuRAG addresses this through layout-aware parsing that inserts placeholder tags (e.g., [Image page=5 idx=2]) at image locations, then uses the Qwen-VL vision-language model to generate captions(Bai et al., 2023). These captions are indexed alongside text, enabling retrieval of visual evidence.

### 2.5 Key Insights for Our Approach

Our analysis of KohakuRAG shaped our HERO pipeline design. First, hierarchical parsing consistently outperforms flat chunking for structure preservation(Jin et al., 2025). Second, hybrid retrieval is mandatory to handle both precise technical terms (lexical) and conceptual queries (semantic). Third, minimizing external dependencies improves reliability and reproducibility. Fourth, multimodal processing is essential, as text-only systems cannot retrieve data from figures.

Addressing the high overhead and API costs of previous approaches, HERO employs local OCR (EasyOCR + Docling) and weighted fusion. This explicitly boosts table and visual chunks rather than relying solely on semantic

Table 1: Comprehensive performance comparison (39 queries). The Text-Only pipeline achieves superior results across coverage (Recall) and ranking quality (nDCG).

| Metric | Hybrid | Text-Only | Gain |
|---|---|---|---|
| *Recall@K (Coverage)* | | | |
| Recall@1 | 79.49% | **87.18%** | **+7.69%** |
| Recall@3 | 89.74% | **92.31%** | +2.57% |
| Recall@5 | 92.31% | **92.31%** | 0.00% |
| Recall@10 | 92.31% | **94.87%** | +2.56% |
| *nDCG@K (Ranking Quality)* | | | |
| nDCG@1 | 0.7949 | **0.8718** | **+0.0769** |
| nDCG@3 | 0.8497 | **0.8859** | +0.0362 |
| nDCG@5 | 0.8564 | **0.8859** | +0.0295 |
| nDCG@10 | 0.8617 | **0.8950** | +0.0333 |
| *Overall Accuracy* | | | |
| MRR | 0.8526 | **0.8974** | **+0.0448** |

similarity.

## 3 Methods

Our methodology evolved through two major iterations: a baseline BM25 system that established fundamental capabilities and revealed critical limitations in processing complex document structures, followed by the HERO (Hybrid Evidence Retrieval Optimization) pipeline that addressed these limitations through multimodal parsing, weighted hybrid retrieval, and adaptive generation.

### 3.1 Baseline BM25 System

To assess RAG and document trade-offs, we used an improved BM25 baseline. Figure 2 shows the architecture, featuring dual indexing and adaptive query routing for hybrid retrieval.

**Table Paradox.** To investigate table impact, we compared Hybrid and Text-Only datasets. Text-Only outperformed Hybrid across all cases (Table 1), challenging the assumption that more data yields better results(Feng et al., 2025).

Root cause: Markdown delimiters introduce structural noise. Figure 3 shows how delimiters dilute semantic keywords, breaking BM25's term-rarity scoring.

Our findings revealed that simply including tables to capture numerical data reduced retrieval quality(Kim et al., 2024). Improving this meant either avoiding BM25 for tables or boosting table chunks with alternative strategies—key insights that drove the development of HERO's weighted fusion.

**Enhanced Tokenization.** Conventional tokenizers often split numeric expressions — for example, "85.5%" is segmented into [’85’, ’.’, ’5’, ’%’] — which disrupts exact value matching. We address this by introducing a regex-based tokenizer that retains numeric units as atomic entities. As shown in Table 2, this refinement is critical for meeting the ±0.1% accuracy tolerance of the task.

Table 2: Comparison of tokenization strategies. Our regex-based approach preserves numerical integrity and cleans structural noise.

| Type | Tokenization Comparison |
|---|---|
| Percentage | Input: "Usage 85.5%"<br>✗ [’usage’, ’85’, ’.’, ’5’, ’%’]<br>✓ [’usage’, ’85.5%’] |
| Floating Point | Input: "Cost: 12.5"<br>✗ [’cost’, ’:’, ’12’, ’.’, ’5’]<br>✓ [’cost’, ’:’, ’12.5’] |
| Technical Term | Input: "tCO2e emission"<br>✗ [’tco’, ’2’, ’e’, ’emission’]<br>✓ [’tco2e’, ’emission’] |

Table 3: Adaptive retrieval strategy. Note the specific weight penalty applied to table chunks to mitigate structural noise.

| Query Type | Retrieval Composition | Weight |
|---|---|---|
| Text Query | Top-10 from Text-Only Index | 1.0 |
| Table Query | Top-7 from Text-Only Index | 1.0 |
| | Top-3 from Hybrid Index | **0.8 (Penalty)** |

**Adaptive Retrieval.** Query routing applies distinct strategies based on detected type. Table 3 details the composition and weighting logic.

**Two-Stage Generation.** Generation uses different LLM models with a two-stage fallback mechanism designed to balance precision and recall. The first stage prioritizes strict evidence

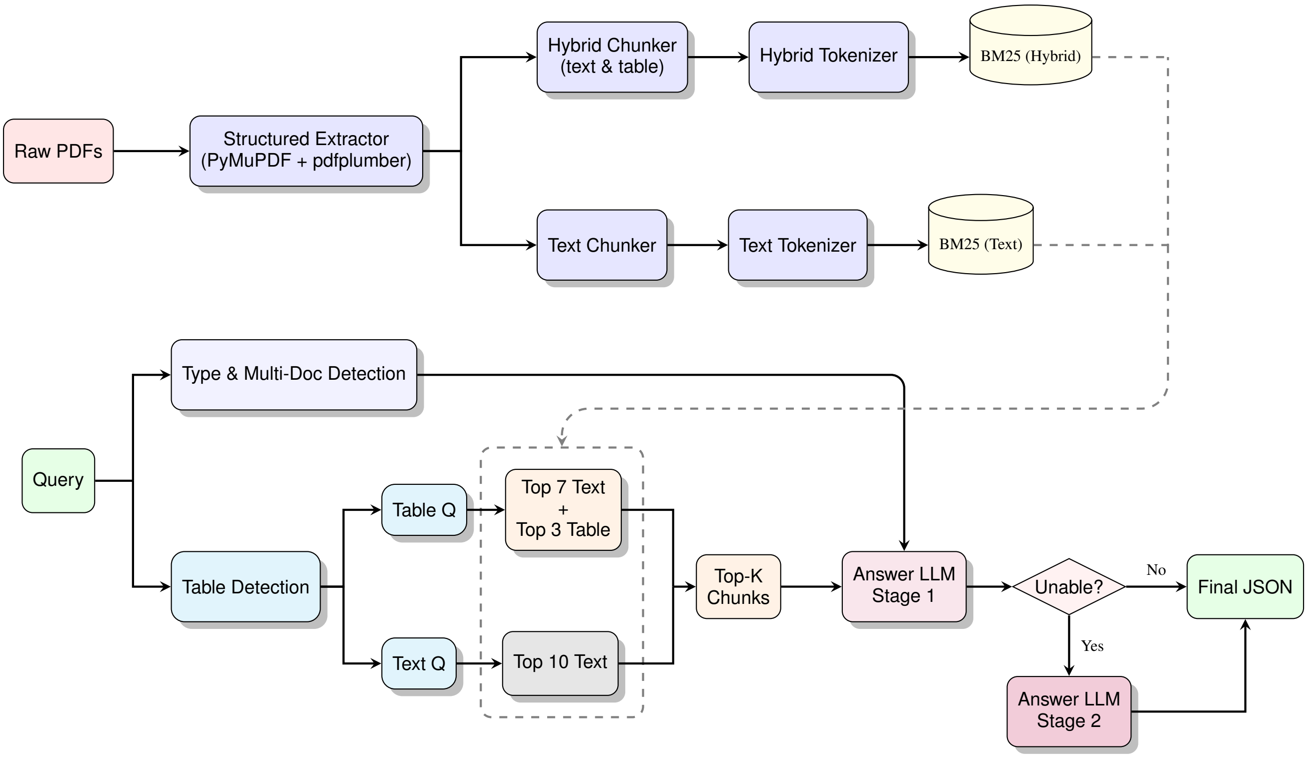

Figure 2: Baseline BM25 Pipeline: Dual indexing (Hybrid vs Text-Only) with adaptive routing and two-stage generation.

**A. Text-Only Chunk (Clean)**

"GPT-3 uses 3640 GPU hours."

→ Tokens: [gpt, 3, uses, 3640, gpu, hours]

*Result:* High term frequency for "GPU".

**B. Hybrid Chunk (Noisy)**

"| Model | GPU | Hours | \n | GPT-3 | ..."

→ Tokens: [|, model, |, gpu, |, hours, |, ...]

Result: Token Dilution. The keyword "GPU" is buried by structural tokens (|), lowering its retrieval score.

Figure 3: Visualizing the "Token Dilution" effect. Frequent markdown delimiters in tables artificially inflate document length and reduce keyword relevance.

Table 4: Two-stage generation logic. The system prioritizes precision in Stage 1 and falls back to a broader context in Stage 2 only upon refusal.

| Stage | Context | Temp | Behavior Strategy |
|---|---|---|---|
| 1 | Top-3 | 0.1 | **Strict**: is_blank if evidence is not enough. |
| 2 | Top-10 | 0.3 | **Relaxed**: inference from partial evidence. |

alignment, while the second stage relaxes constraints when additional context is required. The parameters for each stage are summarized in Table 4.

**Performance and Limitations.** The evaluation revealed a tight clustering of performance across baseline models (**0.75–0.81**), indicating that retrieval quality acts as the primary bottleneck. Table 5 reports the cross-model comparison, and Table 6 analyzes the three fundamental limitations hindering further improvement.

Table 5: Model Performance Comparison (Baseline Configuration).

| Model | Score | Role |
|---|---|---|
| GPT-4o-mini | 0.753 | Benchmark |
| GPT-4o | 0.768 | Benchmark |
| GPT-5.1 | 0.778 | Benchmark |
| **Gemini 2.5 Pro** | **0.807** | **Baseline Winner** |
| **Gemini 3 Pro Preview** | **0.853** | **SOTA Reference** |

## 3.2 HERO Pipeline

HERO augments the baseline system with targeted architectural improvements while retain-

Table 6: Root cause analysis of fundamental limitations in the BM25 baseline.

| Limitation | Mechanism of Failure | Impact |
|---|---|---|
| Semantic Gap | BM25 misses synonyms (e.g., "carbon footprint" $\neq$ "greenhouse gas emissions"). | Recall Drop |
| Table Dilemma | Including tables introduces structural noise (Markdown symbols); excluding them loses data. | No-Win Scenario |
| Precision Limit | Without a semantic reranker, the Top-K context contains irrelevant chunks. | Hallucination |

ing components that already perform well. The pipeline operates in two phases: offline indexing and online query processing. Figure 4 illustrates the complete architecture, emphasizing the multimodal ingestion layer and weighted fusion(Peng et al., 2025).

**Visual-Aware Ingestion.** HERO replaces basic PDF parsing with Docling, which classifies document elements into semantic categories and preserves spatial structure(Auer et al., 2024; Peng et al., 2025). Table 7 contrasts Docling's capabilities against traditional linear parsers.

Table 7: Docling vs traditional parsers. Docling preserves hierarchical structure and enables type-aware downstream processing.

| Feature | Traditional Parser | Docling |
|---|---|---|
| Element Classification | ✗ | ✓ |
| Layout Preservation | ✗ | ✓ |
| Table Structure (JSON) | ✗ | ✓ |
| Merged Cell Handling | ✗ | ✓ |
| Figure Detection | ✗ | ✓ |
| Output Types | Text only | Text<br>Table<br>Figure<br>Equation |

For visual elements, HERO employs a two-step OCR pipeline to convert pixel data into searchable chunks labeled by content type. Table 8 summarizes the final corpus composition.

**Dual Indexing.** Maximizing retrieval efficacy capitalizes on two complementary paradigms. The corpus is represented in two indexes: a sparse index built on BM25Okapi, optimized for lexical matching, and a dense index using BGE-M3 embeddings(Chen et al., 2024; Li et al., 2023), which excels at semantic similarity. High recall is ensured across technical and abstract queries. Table 9 details the parameter configuration and design rationale.

Table 8: HERO corpus statistics after visual-aware ingestion. Type labels enable downstream weighted fusion.

| Chunk Type | Count | Percentage |
|---|---|---|
| Text | 7,817 | 93.0% |
| Table | 153 | 1.8% |
| OCR (Figure) | 437 | 5.2% |
| **Total** | **8,407** | **100%** |

Table 9: Dual indexing architecture. BM25 handles exact-match queries while BGE-M3 captures semantic similarity.

| Index | Model | Parameters | Strength |
|---|---|---|---|
| Sparse | BM25Okapi | $k_1 = 1.5$<br>$b = 0.75$ | Precise keyword matching (e.g., "GPT-3") |
| Dense | BGE-M3 | 1024-dim<br>Faiss L2 | Semantic understanding (e.g., "carbon $\approx$ greenhouse") |

**Query Enhancement.** Before retrieval, queries undergo expansion and metadata extraction to bridge the semantic gap. Table 10 shows representative examples from our 47-term dictionary. Additionally, author names (e.g., "Strubell") and years (e.g., "2023") are assigned 2.0× weight due to their high reliability as metadata signals.

**Weighted RRF.** HERO combines BM25 and BGE-M3 results using a weighted variant of Reciprocal Rank Fusion(RRF)(Cormack et al., 2009). Unlike standard RRF, our variant applies content-type heuristics:

$$\text{Score}(d) = \sum_{r \in \{\text{BM25,BGE}\}} \frac{w_{\text{type}(d)}}{k + \text{rank}_r(d)} \quad (1)$$

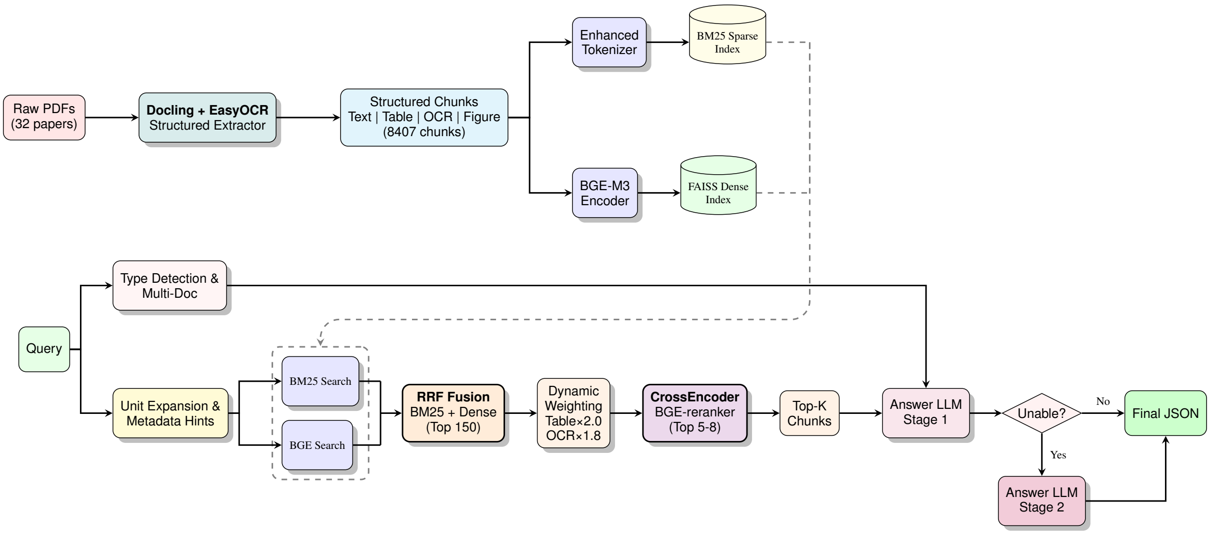
Raw PDFs (32 papers)
Docling + EasyOCR Structured Extractor
Structured Chunks Text | Table | OCR | Figure (8407 chunks)
Enhanced Tokenizer
BM25 Sparse Index
BGE-M3 Encoder
FAISS Dense Index
Type Detection & Multi-Doc
Query
Unit Expansion & Metadata Hints
BM25 Search
BGE Search
RRF Fusion BM25 + Dense (Top 150)
Dynamic Weighting Table×2.0 OCR×1.8
CrossEncoder BGE-reranker (Top 5-8)
Top-K Chunks
Answer LLM Stage 1
Unable?
No
Yes
Final JSON
Answer LLM Stage 2

Figure 4: HERO Pipeline: Visual-aware ingestion, weighted hybrid retrieval, CrossEncoder reranking, and adaptive generation.

Table 10: Unit expansion examples. Abbreviations are expanded to full semantic forms to improve embedding quality.

| Abbreviation | Domain | Expansion |
|---|---|---|
| tCO2e | Emissions | "tons carbon dioxide equivalent emissions" |
| MWh | Energy | "megawatt hour energy consumption power" |
| FLOP | Compute | "floating point operation computation training" |
| kWh | Power | "kilowatt hour electricity energy usage" |
| TPU | Hardware | "tensor processing unit google accelerator" |

Table 11: Weighted RRF configuration. Weights reverse the baseline's penalty logic—tables are boosted rather than suppressed.

| Chunk Type | Weight | Rationale |
|---|---|---|
| Text | 1.0 | Baseline (standard prose content) |
| Table | 2.0 | High information density; numerical answers |
| OCR | 1.8 | Unique visual data unavailable elsewhere |
| Metadata | 2.0 | High precision (author/year matches) |

Table 11 presents the weight configuration derived from error analysis. The system retrieves top-150 from each retriever, fuses via weighted RRF, and passes top-50 to reranking. This design directly addresses the baseline's "Table Dilemma" by leveraging dense embeddings' noise-resilience.

**CrossEncoder Reranking.** BGE-reranker-large processes the top-50 candidates using cross-attention mechanisms(Cormack et al., 2009; Zhou and Devlin, 2021). Table 12 contrasts this approach against bi-encoder architectures.

Scores are sigmoid-normalized to [0,1], and chunks are re-sorted. The final top-K (5-8 based on query complexity) proceed to generation. This reranking proved critical for filtering false positives and preventing LLM hallucination.

Table 12: Bi-encoder vs CrossEncoder architectures. Cross-attention enables fine-grained query-document interaction.

| Architecture | Mechanism | Limitation & Strength |
|---|---|---|
| **Bi-encoder** | Independent encoding → dot product | Fast but shallow: score$(q, d) = q \cdot d$ |
| **CrossEncoder** | Joint-processing → cross-attention | Slow but deep: captures $q \leftrightarrow d$ interactions |

**Dynamic Prompting.** Queries are classified into six types via regex patterns, each trigger-ing specialized system prompts. Table 13 details the classification logic and prompt strategies.

Table 13: Dynamic prompting taxonomy. Type-specific instructions improve answer precision for domain patterns.

| Query Type | Trigger Pattern | Prompt Instruction |
|---|---|---|
| Table_Extraction | "gpu hours", "batch size", "execution time" | "Search row-by-row. Match exact column names." |
| Score_Extraction | "score", "accuracy", "gsm8k" | "Extract numerical metrics. Verify units." |
| Model_Specific | Model names (e.g., "mixtral", "llama") | "Focus on model results. Avoid aggregates." |
| Term_Extraction | "what is the name", "which framework" | "Provide technical definition. Cite source." |
| Numeric_Counting | "how many", "what percentage" | "Count occurrences. Double-check arithmetic." |
| General | Default | "Synthesize from all evidence. Be concise." |

**Adaptive Generation.** HERO maintains the two-stage generation scheme but introduces adaptive context selection. Unlike the baseline's fixed Top-3/Top-10 split, HERO dynamically adjusts based on question type classification (6 categories: table_extraction, model_specific, score_extraction, term_extraction, numeric_extraction, general). Table 14 details the adaptive logic(Taguchi et al., 2025).

Table 14: HERO adaptive generation strategy. Context size adapts to question complexity; both stages use low temperature for consistency.

| Stage | Context | Temp | Strategy |
|---|---|---|---|
| 1 | Top-3 | 0.1 | **Strict or Aggressive** (complex types) |
| 2 | Top-7/10 | 0.1 | **Always Aggressive**: Top-10 for complex, Top-7 for general |

The LLM outputs structured JSON with fields: answer, answer_value, answer_unit, supporting_text, explanation, and source_doc_ids.

**Implementation.** Table 15 summarizes the technical stack, computational costs, and performance metrics.

Table 15: HERO implementation specifications. The pipeline achieves production-ready performance without GPU requirements.

| Component | Technology | Specification |
|---|---|---|
| *Core Libraries* | | |
| Embedding Model | BGE-M3 | transformers 4.37, 1024-dim |
| Sparse Retrieval | BM25Okapi | rank$_bm$250.2.2 |
| Dense Index | Faiss | faiss-cpu 1.7.4, IndexFlatL2 |
| PDF Parser | Docling | docling 1.2.0 |
| OCR Engine | EasyOCR | easyocr 1.7, threshold=0.6 |
| LLM API | Gemini 2.5 Pro | google-generativeai 0.3.2 |
| *Performance Metrics* | | |
| Indexing Time | 45 min | 16-core CPU, 32GB RAM |
| Query Latency | 8-12 sec | Dominated by LLM generation |
| Embedding Cost | \$2.50 | BGE-M3 API for 8,407 chunks |
| Storage | 2.1 GB | Complete indexed database |

## 4 Experiments and Results

We evaluated the HERO pipeline on the competition's hold-out test set (282 queries) and an internal validation set. The primary metric is the custom WattBot Score, supplemented by standard IR metrics.

### 4.1 Retrieval Performance Analysis

Table 16 compares our HERO pipeline against the BM25 baseline. The results demonstrate significant improvements across all metrics.

Key observations from the experiment:

- **Visual Data Unlocked:** By integrating EasyOCR(Kittinaradorn et al., 2024), HERO successfully retrieved critical energy metrics trapped in charts, boosting Recall@1 by 1%.
- **Structure-Aware Precision:** The heuristic-weighted RRF (Table ×2.0)

Table 16: Retrieval performance comparison. HERO demonstrates consistent improvements over the Baseline across all coverage and ranking metrics.

| Metric | Baseline | HERO |
|---|---|---|
| *Recall@K (Coverage)* | | |
| Recall@1 | 79.49% | **80.49%** |
| Recall@3 | 89.74% | **89.80%** |
| Recall@5 | 92.31% | **92.68%** |
| Recall@10 | 92.31% | **95.12%** |
| *nDCG@K (Ranking Quality)* | | |
| nDCG@1 | 0.7949 | **0.8049** |
| nDCG@3 | 0.8497 | **0.8502** |
| nDCG@5 | 0.8564 | **0.8584** |
| nDCG@10 | 0.8617 | **0.8665** |
| *Overall Accuracy* | | |
| MRR | 0.8526 | **0.8560** |

effectively surfaced correct numeric answers that were previously buried by structural noise in the baseline.

## 4.2 Final Leaderboard Results

Figure 5 presents our final standing on the public leaderboard. To evaluate the synergy between retrieval architecture and model capability, we benchmarked the Baseline and HERO pipelines across models ranging from GPT-4o-mini to the state-of-the-art Gemini 3.0 Pro.

A critical finding is that HERO consistently amplifies performance across all model sizes. While the Gemini 3.0 Pro Baseline (0.853) establishes a high floor due to its superior reasoning, integrating HERO further elevates the score to 0.886. This demonstrates that even state-of-the-art models are bottlenecked by standard retrieval quality in complex extraction tasks; therefore, pairing high-capacity models with structure-aware retrieval is essential for achieving peak performance.

# 5 Conclusion

Our study shows that structured ESG information cannot be reliably extracted from heterogeneous technical documents using standard RAG architectures alone. The investigation through the WattBot 2025 competition yielded four critical insights that challenge conventional retrieval design assumptions:

**The Table Paradox Necessitates Hybrid Approaches.** Our controlled experiments revealed a fundamental tension: although tables encode 90% of numerical evidence, their structural artifacts reduce BM25 effectiveness by roughly one point in Recall@1. Markdown delimiters introduce structural noise that dilutes term-frequency statistics, breaking BM25's core scoring mechanism(Huang et al., 2022). Addressing this paradox requires hybrid retrieval - where the dense embeddings compensate for sparse methods' structural brittleness - rather than further tokenization refinements. HERO's weighted RRF (Table×2.0, OCR×1.8) transforms this liability into an asset by explicitly boosting high-value content types.

**Visual-Aware Ingestion Unlocks Hidden Evidence.** Academic papers often place essential information in charts and figures. By integrating Docling with EasyOCR, HERO successfully extracted 482 visual chunks (5.7% of corpus) that were completely inaccessible to text-only systems. This multimodal processing directly contributed to the +1% gain in Recall@1, proving essential for comprehensive document understanding in technical domains.

**Architecture Amplifies Model Capability.** Our cross-model evaluation demonstrates that HERO consistently improves performance across all model sizes—from GPT-4o-mini (+3.6 points) to Gemini 3.0 Pro (+3.3 points). Critically, even state-of-the-art models remain bottlenecked by retrieval quality: the Gemini 3.0 Pro Baseline (0.853) could only reach 0.886 when paired with HERO. This empirical evidence suggests that for structured data extraction, optimizing retrieval architecture yields comparable or greater returns than model upgrades alone.

**Dynamic Adaptation Beats Static Heuristics.** Static penalties (e.g., 0.8× weight on tables) create no-win scenarios where including critical content degrades overall quality. HERO's

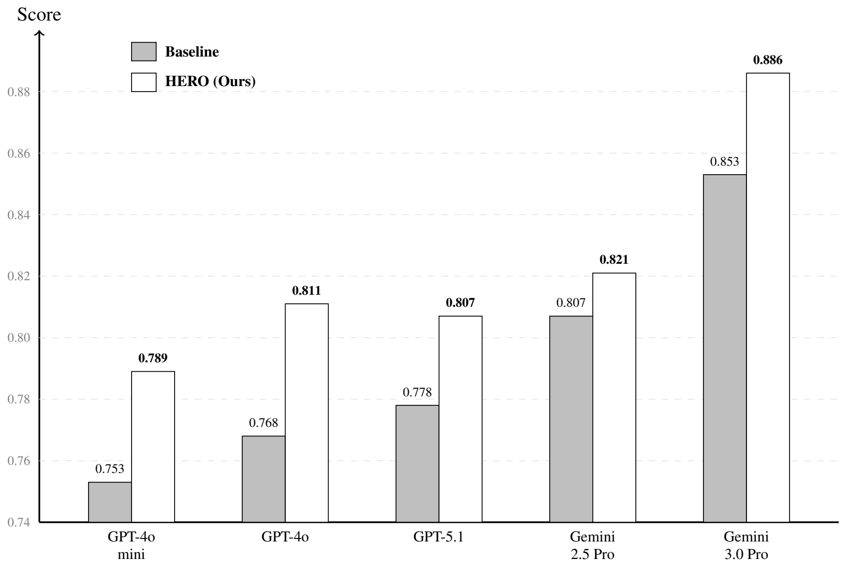

Figure 5: Performance Comparison. HERO consistently improves retrieval scores across all models. Notably, combining HERO with Gemini 3.0 Pro unlocks the highest performance (0.886), surpassing the strong baseline.

dynamic approach—combining content-type weighting, six-category query classification, and adaptive context selection (Top-3 vs Top-7/10)—adjusts to heterogeneous document structures and query complexities. This flexibility proved essential for handling the diverse question types in sustainability intelligence tasks.

By synthesizing visual OCR, weighted hybrid retrieval, CrossEncoder reranking, and adaptive generation, HERO achieves production-grade performance (45-minute indexing, $2.50 embedding cost, 8-12 second query latency) while significantly improving accuracy. Future work should explore learned weighting schemes to replace manual heuristics, extend to multilingual corpora, and investigate hierarchical indexing strategies for even larger document collections.

## References

Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, et al. 2024. Docling technical report. *arXiv preprint arXiv:2408.09869*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Kaiyue Feng, Siyue Zhang, Bingsen Chen, Yilun Zhao, and Chen Zhao. 2025. Sportreason: Evaluating retrieval-augmented reasoning across tables and text for sports question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 649–662.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3929–3938. JMLR.org.

Junjie Huang, Wanjun Zhong, Qian Liu, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Mixed-modality representation learning and pre-training for joint table-and-text retrieval in openqa. *arXiv preprint arXiv:2210.05197*.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. 2025. Hierarchical document refinement for long-context retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of* *the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3502–3520.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Kihun Kim, Mintae Kim, Hokyung Lee, Seongik Park, Youngsub Han, and Byoung-Ki Jeon. 2024. Thorr: Complex table retrieval and refinement for rag. In *Proceedings of the Workshop Information Retrieval's Role in RAG Systems (IR-RAG 2024) co-located with the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, volume 3784, pages 50–55.

Rakpong Kittinaradorn et al. 2024. Easyocr: Ready-to-use ocr with 80+ supported languages. `https://github.com/JaidedAI/EasyOCR`. Accessed: 2025-05-15.

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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *arXiv preprint arXiv:2312.15503*. Foundation for BGE-Reranker.

Xiangyu Peng, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, and Chien-Sheng Wu. 2025. Unidoc-bench: A unified benchmark for document-centric multimodal rag. *arXiv preprint arXiv:2510.03663*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Chihiro Taguchi, Seiji Maekawa, and Nikita Bhutani. 2025. Efficient context selection for long-context qa: No tuning, no iteration, just adaptive-$k$. *arXiv preprint arXiv:2506.08479*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Xing Wei, Bruce Croft, and Andrew McCallum. 2006. Table extraction for answer retrieval. *Information retrieval*, 9(5):589–611.

Giulio Zhou and Jacob Devlin. 2021. Multi-vector attention models for deep re-ranking. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 5452–5456.