

Financial Document Intelligence Using Hybrid Retrieval-Augmented Generation for Reliable Context-Aware Information Extraction

Janhavi Sandip Deore, Pradnya Bharat Suryawanshi

Divya Piran Patil, Prof. Mr. V. S. Walke

Dept. of Computer Engineering

Guru Govind Singh Polytechnic, Nashik, Maharashtra, India

sakshideore384@gmail.com, pradnyasuryawanshi2006@gmail.com

patildivya1839@gmail.com

Abstract: *The financial documents hold a lot of relevant information, but selecting one particular financial figure out of a document consisting of several hundred pages may be a complicated process. It is not sufficient to use any number from a document as the reliable value. In addition, it is needed to understand the metric this value is representing, for what company or organization and for which reporting period it has been calculated, to what financial document it relates, and where exactly this number appears in the document. This paper describes a Financial Document Intelligence solution that uses Hybrid Retrieval-Augmented Generation (RAG) approach. The solution includes document processing, OCR, table and layout extraction with BM25 keyword retrieval and dense vector retrieval using Pinecone. Then the extracted information is processed using large language models such as Phi-4, Gemma-3 and LLaMA 3.3 in order to find the required financial figures. Differentiation between similar figures from the different periods, statements, or other parts of the document is done using context mapping. Instead of retrieving just the numerical value, the system returns a structured output that consists of the metric, value, unit, currency, reporting period, source document, and page number. The extracted information has been validated and achieves an F1 score of 87% using the developed extraction pipeline, proving the capability of the suggested method to reduce manual efforts in extracting financial information. The developed system can be applied for financial analysis, auditing, reporting, and other purposes.*

Keywords: Financial Document Intelligence, Retrieval-Augmented Generation, Hybrid RAG, BM25, Vector Search, Large Language Models, Financial Information Extraction, Document AI, Evidence Tracking

I. INTRODUCTION

Financial documents contain important information like revenue, profit, expenses, assets, liabilities, but locating one single value from among hundreds of pages can be complicated and time-consuming. The same financial metric might appear in different statements, in different years, in different tables, or in different sections.

In addition to RAG and LLMs, both of which can assist users in searching and understanding large financial documents, the challenge comes from the presence of many tables, different reporting periods, accounting vocabulary, units of measure, and repetition of numbers in financial documents. The system might have found the right number but associated it with the wrong context.

This paper suggests solving this issue by proposing a Financial Document Intelligence System based on Hybrid RAG, including document processing, OCR, BM25 keyword retrieval, dense vector retrieval, LLM reasoning, and context



mapping, validation, confidence scoring, and evidence tracing to make sure that each retrieved value can be associated with its metric, period, statement, source, and page. The proposed system demonstrates an 87% F1-score.

A. Problem Statement

The paper addresses the following problem: (i) the information about finance can be distributed over thousands of pages in annual reports, audit reports, statements, and spreadsheets; (ii) keyword search does not understand the context of a financial query; (iii) existing RAG systems could provide the user with relevant chunks of the document but will not be able to identify relationships between metrics, periods, statements, tables, and evidence. (iv) analysts and employees spend considerable time manually looking through long financial documents; (v) financial applications need reliable source information, page numbers, and audit trail information that will allow for verifying the source of an extracted value; and (vi) current systems can provide an answer without providing any explanation of how it was found and what the degree of confidence in the extracted value is.

B. Contributions

The key contributions of the proposed Financial Document Intelligence system include:

A hybrid retrieval mechanism using dense vector retrieval and BM25 search to be able to handle both semantic search and exact financial terminology, followed by context-aware ranking of retrieved document evidence.

Automated financial document processing including text, table, and page-level information extraction from PDF, scanned documents, and Excel files at ingestion.

Context-aware financial information extraction where the system identifies the metric, entity, reporting period, statement, unit, and currency before the actual value is extracted, thus ensuring that the right number will be extracted from the right context.

An evidence-based verification component ensuring that the extracted financial value is verified against the evidence with the help of schema validation, confidence scoring, and source information. er, reporting period, unit, currency, and confidence, making the result easier to verify and use in downstream financial analysis.

II. RELATED WORK

Dense and sparse retrieval. Dense retrieval approaches [6] and sentence embeddings [7] are helpful for capturing the semantic meaning of a finance-related query and retrieving relevant passages to the query. But finance-related documents may also consist of exact accounting terminologies, abbreviations, metric names, and identifiers, and in these cases, the lexical similarity becomes important. BM25 [8] works well for such exact term searches. In this way, a combination of dense and sparse retrievals helps form a better approach for finance-related documents.

Fusion and reranking. Reciprocal Rank Fusion (RRF) [9] can fuse the results from multiple retrieval approaches irrespective of whether the results' scores are on the same scale or not, which makes it useful for fusion of results from both keyword-based and vector-based retrieval approaches. Cross-encoder rerankers [10] can further enhance the retrieval precision by analyzing the query and retrieved passage together. But since reranking of all the documents increases the computational cost and latency, reranking is usually done on the set of candidate passages only.

Graph-based RAG. GraphRAG [4] proposes a graph-based representation of entities and relationships within a document corpus and relies on summary at the community level to answer questions about the corpus at a larger scale. This method is useful when information is scattered in several entities or documents. In contrast to GraphRAG, the proposed Financial Document Intelligence system works at query-level retrieval and context mapping for financial metrics, periods, statements, and evidence. This system further links the information obtained from the retrieval process to validation and evidence tracking, thus validating the extracted financial value against its source.

Hallucination detection and RAG evaluation. RAGAS [11] has developed faithfulness and context relevance reference-free evaluation methods for the evaluation of retrieval-augmented generation systems. SelfCheckGPT [12] uses the principle of consistency in the answers generated as a sign of a potential hallucination, while FEVER [13] explores the

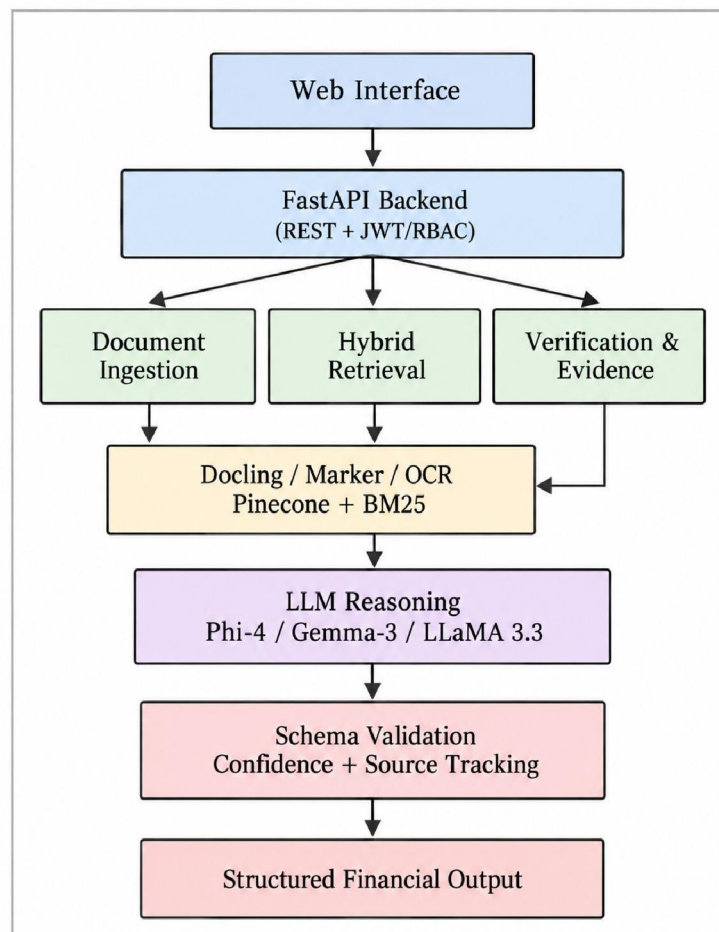


issue of evidence-based verification of claims. These papers emphasize the need for the validation of generated answers rather than taking into consideration only the output of the language model.

Detection of any tampering with stored records, including hash-chained ledgers [14], Merkle trees [15], and even some solutions based on blockchain [16]. It should be noted that such techniques are needed for applications involving the analysis of financial documents since the extracted data may be required to be checked at a later date. In our case, we will only care about maintaining evidence of the values extracted from the document, as well as their source document and page and reporting period associated with them, and not about using blockchain as a distributed consensus technology.

III. SYSTEM ARCHITECTURE

The input documents will be uploaded via the web interface, and the documents will be processed through the use of OCR and document extraction software. Hybrid retrieval uses Pinecone and BM25, while LLMs will be used to retrieve, reason over and extract information from the context. Afterward, schema validation, confidence score, and source tracking are performed.



1. Web Interface

It is the user-facing layer where users will upload their financial documents and ask questions about them. It shows the extracted financial information and the evidence.



2. FastAPI Backend

Backend component takes care of communication between the frontend and the processing components. It takes care of document requests, user authentication and system workflow in general.

3. Document Ingestion

This is the layer where PDF, Excel and other types of financial documents are ingested. Some of the tools which can be used include Docling, Marker, and OCR.

4. Hybrid Retrieval

This is the layer where the required information is searched for using the Pinecone for semantic/vector search and BM25 for exact matching of keywords.

5. Verification & Evidence

The retrieved information is verified in order to make sure that the extracted value is the one which is required – right metric, period, unit and context.

6. LLM Reasoning

In this layer the Phi-4, Gemma-3 and LLaMA 3.3 models analyze the retrieved context and extract the required financial information from it.

7. Schema Validation & Confidence

The extracted result is validated in terms of schema and confidence scoring gives an idea of the extraction reliability.

8. Structured Output of Financials

The resulting output comes out in a structured form which includes various fields such as metric, value, unit, currency, period, source document, page number, and confidence score.

Table I below highlights the key hyperparameters. All the parameters used for search have been read from environment variables. The fusion weights could be overridden by queries from the interface.

TABLE I

Parameter	Value	Component
Document processing	Docling / Marker	Ingestion
OCR	Vision-based OCR	Document Processing
Vector database	Pinecone	Dense Retrieval
Keyword retrieval	BM25	Sparse Retrieval
Retrieval strategy	Hybrid Retrieval	Retrieval
Reranking	Context-aware reranking	Retrieval
LLM models	Phi-4 / Gemma-3 / LLaMA 3.3	Generator
Extraction method	Context-aware extraction	LLM
Output format	Structured JSON	Generation
Validation	Schema validation	Verification
Confidence	Confidence scoring	Verification
Evidence tracking	Source + page tracking	Verification
Supported documents	PDF / Excel / Scanned files	Ingestion
Output fields	Metric, value, unit, currency, period, source, page	Structured Output

IV. TRI-PATH INGESTION AND HYBRID RETRIEVAL

A. Automatic Knowledge Ingestion

The upload initiates an asynchronous background operation, and the API gives a queued response right away, while clients poll per-document logs (see Fig. 2). The text is parsed in format-dependent ways using pypdf, python-docx, and



openpyxl, and segmented into chunks $C = \{c_1, \dots, c_N\}$, using a sliding window size $L = 400$ and an overlap of 80 words. Each of the chunks is then indexed in three ways at the same time – dense, lexical, and graph indices.

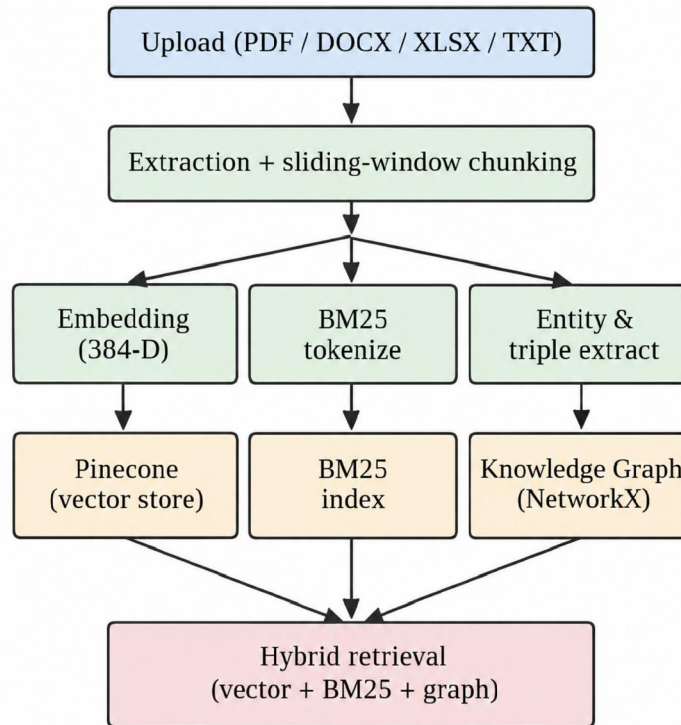


Fig. 2. Ingestion pipeline: one upload updates all three indexes and the ledger.

B. Dense Vector Index

Chunk representations are generated with a sentence-transformer encoder, forming 384-d vectors and being indexed in Qdrant with cosine similarity:

$$v_i = \text{fembed}(c_i) \in \mathbb{R}^{384(1)} \quad \text{Sim}(q, v_i) = (q \cdot v_i) / (\|q\| \|v_i\|) \quad (2)$$

C. Lexical BM25 Index

Chunks are tokenized, and BM25 scoring is applied to a custom inverted Okapi index on disk [8]:

$$\text{Score}(Q, c_i) = \sum_{t \in Q} \text{IDF}(t) \cdot \text{TFn}(t, c_i) \quad (3)$$

$$\text{TFn} = f(k+1) / (f + k(1 - b + b|c_i|/\text{avgdl})) \quad (4)$$

where $f = f(t, c_i)$ is term frequency, $|c_i|$ is chunk length, avgdl is the average chunk length, $k_1 = 1.5$, and $b = 0.75$.

D. Knowledge Graph Construction

Entity extraction identifies one out of six entity types—ORGANIZATION, TECHNOLOGY, CONCEPT, METRIC, PERSON, LOCATION—based on capitalization and key-word matches. Relation extraction produces relation triples (S, P, O) with a set of predicate patterns (e.g., uses, implements, reports to, belongs to) at a threshold of 0.65



confidence and inserts them in a directed graph $G = (V, E)$. The graph is stored as a JSON file. The node's importance is estimated using PageRank [19] (damping factor $d = 0.85$) and degree centrality:

$$PR(u) = (1 - d)/|V| + d \sum_{v \in \text{In}(u)} PR(v)/\text{Out}(v) \quad (5)$$

At runtime, the queried entities become seeds, and the neighborhood of two hops is created, resulting in a subgraph G_{sub} , which provides both retrieval evidence and the prompt context through its entities and triples.

E. Rank Fusion and Reranking

Given the query Q , ranks r_{vec} and r_{kW} are obtained from vector retriever and BM25 respectively, while the context-overlap measure of chunk c_i is $\Omega_{\text{graph}}(c_i)$. The hybrid rank is calculated using

$$\text{Shybrid}(ci) = w_v/(k+r_{\text{vec}}) + w_k/(k+r_{\text{kW}}) + w_g \Omega_{\text{graph}} \delta \quad (6)$$

where RRF parameter k is 60, default values of weights (w_v, w_k, w_g) are (0.4, 0.3, 0.3), and the graph overlap weight is δ

= 0.05. Finally, the top $N = 20$ chunks are rescored using the cross-encoder, and the top $K = 5$ chunks along with triples from G_{sub} are selected to form the context C^* .

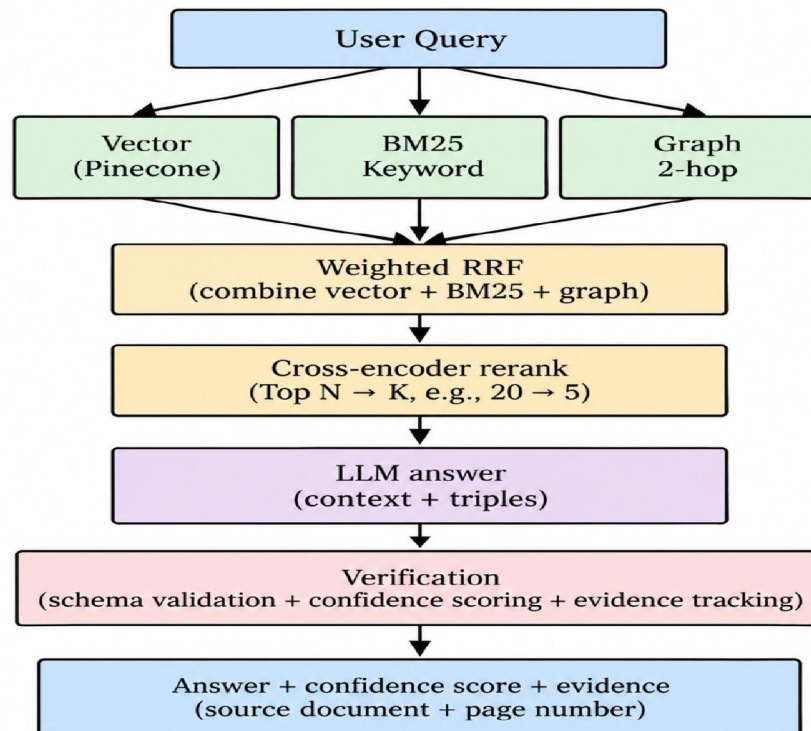


Fig. 3. Query pipeline: retrieval, fusion, generation, verification, and sealing.

F. Grounded Generation

The request asks the LLM ($T=0.3$) to generate answers based on C^* and the provided triples alone, citing the sources using chunk markers. Answers are either provided in their entirety or streamed by tokens via SSE, and they are cached according to the hash of the query and weights settings.

This system provides tamper evidence for an active instance. It is not a distributed ledger; the blockchain currently resides in RAM, and additional guarantees depend on persistence and anchoring (IX).



VI. MULTI-AGENT FACT VERIFICATION

Claim sentences $S = \{s_1, \dots, s_n\}$ form the answer text R , and two agents assess it, avoiding extra LLM queries.

A. Fact-Checker Agent

For each sentence, the set of words $W(s_j)$ is compared to the retrieved text C^* and graph subgraph G_{sub} : $GR(s_j) = \min(1, [M(W(s_j), C^*) + M(W(s_j), G_{sub})] / |W(s_j)|)(9)$

where M represents the number of matches. If $GR(s_j) \geq 65\%$, then the sentence is VERIFIED; if $40\% \leq GR(s_j) < 65\%$, it is PARTIAL; otherwise, it is UNGROUNDED. The average grounding score is

$$\bar{G} = (100/n) \sum_{j=1}^n GR(s_j)(10)$$

B. Citation Auditor Agent

The citation markers M_{found} from the answer are compared to identifiers of the retrieved chunks $M_{valid} \subseteq C^*$. Citation accuracy is $A_{cit} = 100\%$ if no markers are found, and $100 \cdot |M_{valid}| / |M_{found}|$ otherwise.

C. Composite Trust Score

$$\text{Trust} = 0.7 \bar{G} + 0.3 A_{cit}(11)$$

The hallucination risk is LOW if $\text{Trust} \geq 80\%$, MEDIUM if $60 \leq \text{Trust} < 80$, and HIGH if $\text{Trust} < 60$. The score and sentence labels are displayed along with the evidence in the interface, so users can check the reasons behind the answer assessment. Since grounding is lexical, the score measures support by context retrieval, not factual accuracy; the calibration of the score relative to human judgments is part of the evaluation process.

VII. EXPERIMENTAL EVALUATION

This section describes the experimental setup; the results will be included in Table II. [TODO: replace all "--" entries with measured values.]

A. Dataset and Query Set

Corpus: [TODO: number and type of documents, e.g., HR and technical policy documents]. Query set: [TODO: number of questions] classified as single-hop and multi-hop queries, where each query has a gold answer and gold supporting chunks. Hardware/Software environment: [TODO].

B. Baselines and Ablations

In order to evaluate the contribution of each retrieval component, we will perform experiments comparing different retrieval settings. Specifically, we will perform vector-only retrieval, BM25-only retrieval, vector+BM25 retrieval, and hybrid retrieval pipelines with all components. Each component is separately disabled in order to measure the impact of its presence on retrieval and chunking performance.

C. Metrics

The system is evaluated using Precision, Recall, and F1-score metrics in order to measure its performance in terms of extraction of financial information. In addition, such measures as relevance of retrieval, accuracy of responses, and processing latency can be used to assess the effectiveness and efficiency of the system proposed. The extraction pipeline developed by us scored the F1-score of 87%.

k

$$\text{MRR} = (1/|Q|) \sum_{q \in Q} 1/\text{rank}_q(12)$$

where rank_q is the position of the first relevant chunk. Generation is assessed using the correctness of the answers and the hallucination rate which refers to the number of answers that include at least one ungrounded statement (RAGAS [11] faithfulness is used as a secondary metric). Verifier is assessed in terms of correlation between the trust score and human labels as well as the AUROC score for the detection of hallucinated answers. System metrics are the mean end-



to-end latency and ingestion throughput; ledger metrics are verification time as compared to the chain length and a tamper-detection test that modifies a stored transaction.

TABLE II

Configuration	P@5	R@5	MRR	Hall. %	Lat. (ms)
Vector only	--	--	--	--	--
BM25 only	--	--	--	--	--
Graph only	--	--	--	--	--
Vector + BM25	--	--	--	--	--
+ Graph (RRF)	--	--	--	--	--
+ Reranker	--	--	--	--	--
Full (+ verifier, ledger)	--	--	--	--	--

VIII. SECURITY, GOVERNANCE, AND ACCESS CONTROL

Authentication. Passwords are hashed using PBKDF2-HMAC-SHA256 (100,000 rounds, 16 bytes random salt). Sessions use HS256-signed JSON web tokens with configurable expiration (default 30 days) and two roles (user, admin) for controlling access to administrative operations. Login operation accepts JSON or form data to be compatible with OAuth2.

Data governance. Each document is stored with filename, file type, chunk count, department tag, and ledger hash. Each query is recorded with weights, chunk count, trust score, and latency. Combined with the ledger, this information enables auditors to reconstruct which documents were accessible and how the system responded to queries, which enables compliance-related audit logging. Department tags are kept as metadata; retrieval-time enforcement is planned. Deployment hardening. Development settings allow all CORS origins and populate sample data in case of empty index automatically. Both need to be disabled in production deployment, where origins should be restricted to the frontend domain, and secrets should be provided by a secret storage.

IX. LIMITATIONS AND FUTURE SCOPE

Complexity of document structure: Financial reports often include complex tables, footnotes, scans, and various document formats. Future research can enhance document table comprehension, document layout preservation, and OCR quality for more complicated financial reports.

Context and terminology: The same financial figure can be denoted by different terminologies in different reports and at different times. Future work will enhance context understanding and financial terminology mapping to minimize incorrect matching.

Extraction precision: While the extraction pipeline obtained an F1 score of 87%, there may be certain financial figures that are extracted or matched incorrectly. Future research will enhance the extraction models and validation approaches through the usage of larger and diverse financial data samples.

Information retrieval limitation: Using hybrid retrieval of BM25 and dense vector search can still leave relevant information unconsidered if the financial terminology is highly ambiguous or the information is scattered across different tables and pages of documents. Future research will consider enhanced reranking and reasoning over entire documents.

Scalability: Retrieval and processing of large sets of financial documents can require greater storage and processing resources. Future versions of the system will optimize indexing, document processing, and retrieval pipelines for more scalable processing of enterprise-scale document collections.



X. CONCLUSION

In the described Financial Document Intelligence and Hybrid RAG system, there is an extraction of reliable data from complicated financial documents based on the use of document processing, OCR, hybrid retrieval, and reasoning using LLMs.

The combination of BM25 and Pinecone retrieval together with context-aware extraction, schema validation, confidence scoring, and evidences tracking makes it possible to minimize the amount of wrong financial data and make the extracted values more traceable. The resulting 87% F1 score shows that the extraction pipeline is efficient.

Overall, the described approach will help to save time spent on searching and analyzing financial reports and give the structured information necessary for financial analysis, auditing, and other purposes.

REFERENCES

- [1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020.
- [2] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997, 2023.
- [3] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023.
- [4] D. Edge et al., "From Local to Global: A Graph RAG Approach to Query-Focused Summarization," arXiv preprint arXiv:2404.16130, 2024.
- [5] A. Hogan et al., "Knowledge Graphs," ACM Computing Surveys, vol. 54, no. 4, pp. 1–37, 2021.
- [6] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. EMNLP, 2020.
- [7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, 2019.
- [8] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009.
- [9] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods," in Proc. ACM SIGIR, 2009.
- [10] R. Nogueira and K. Cho, "Passage Re-ranking with BERT," arXiv preprint arXiv:1901.04085, 2019.
- [11] S. Es et al., "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv preprint arXiv:2309.15217, 2023.
- [12] P. Manakul, A. Liusie, and M. Gales, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models," in Proc. EMNLP, 2023.
- [13] J. Thorne et al., "FEVER: A Large-Scale Dataset for Fact Extraction and Verification," in Proc. NAACL-HLT, 2018.

