Beyond Semantic Similarity: A Two-Phase Non-Parametric Retrieval Workflow for Corporate Credit Underwriting

Linus Ng Junjia [1] **Ezekiel Tee Kongquan** [1 2] **Kelvin Heng** [1] **Kenneth Zhu Ke** [1] **Zhao Jing Yuan** [1]

[1]*OCBC, Singapore* [2]*Georgia Institute of Technology*. linus.ng@ocbc.com, ezekieltee@ocbc.com, kelvinheng@ocbc.com, kennethzhu@ocbc.com, jingyuanzhao@ocbc.com.

1. Introduction

Corporate credit underwriting relies heavily on the analysis of long-form financial documents such as annual reports and industry reports [1]. Analysts must extract relevant financial indicators, assess risk disclosures, and synthesize insights from documents that can span hundreds of pages and multiple languages [1].

Retrieval-Augmented Generation (RAG) systems have emerged as a promising approach to assist document-intensive workflows [2, 3]. By retrieving supporting passages from external corpora, RAG systems can improve factual grounding and reduce hallucination in language model outputs [2, 3]. However, standard RAG pipelines typically prioritize semantic similarity between queries and document passages [2]. In financial analysis tasks, this objective often fails to align with the needs of analysts [1].

Financial documents often contain narrative descriptions, regulatory disclosures, and repetitive boilerplate language [1]. As a result, similarity-based retrieval systems may surface passages that share terminology with the query but lack actionable analytical value [4]. We refer to this issue as the *similarity–utility gap* [4].

In the following, we introduce a two-phase retrieval and re-ranking architecture designed to close the gap between semantic similarity and decision utility while operating entirely within an on-premise environment.

2. Related Work

Retrieval-Augmented Generation has become a widely adopted approach for grounding large language models in external knowledge sources [2, 3]. Early work introduced dense retrieval methods using neural embeddings to retrieve semantically relevant passages from large corpora [2]. Hybrid retrieval approaches combining lexical search with dense embeddings have further improved recall in heterogeneous document collections [5].

Recent research has explored the use of language models as evaluators for ranking retrieved content [6]. These *LLM-as-a-Judge* approaches leverage the reasoning capabilities of language models to assess the quality and relevance of candidate passages [6]. Such methods have been applied in question answering, information retrieval evaluation, and ranking tasks.

3. Methodology

3.1 Problem Setting

We formalize the task as retrieving and ranking a set of document segments that maximize decision utility rather than semantic similarity. Given a user query q and a corpus of long financial documents $D = \{d_1, d_2, ..., d_n\}$, the objective is to identify a set of passages $P^* \subset D$ that contain verifiable evidence relevant to credit underwriting decisions.

3.2 System Overview

The proposed architecture consists of five main components: document ingestion, hybrid candidate retrieval, adaptive retrieval controller, utility-grounded re-ranking, and context-aware evidence extraction.

The overall pipeline is illustrated conceptually as:

$$q_{\text{statement}} \rightarrow R_{\text{hybrid}}(D) \rightarrow C_{\text{adaptive}} \rightarrow J_{\text{utility}} \rightarrow E_{\text{context}}$$

where $q_{\text{statement}}$ a query supplemented with its corresponding financial statement, R_{hybrid} retrieves an initial candidate pool, C_{adaptive} filters candidates using query-aware reasoning, J_{utility} ranks candidates by decision usefulness, E_{context} extracts the final evidence

3.3 Document Ingestion

Corporate financial documents are segmented into structured sections using document layout cues. Each segment is indexed with metadata including document source, section title, and page references. This preprocessing step ensures that downstream components can leverage structural information during retrieval and extraction.

3.4 Phase 1: Hybrid Candidate Retrieval

The first phase performs broad retrieval to maximize recall across multilingual and heterogeneous financial documents.

Given a query q, we retrieve an initial candidate set C_0 using a hybrid retrieval strategy that combines keyword retrieval and dense semantic retrieval using multilingual embeddings:

$$C_0 = \text{TopK}_{\text{kw}}(q, D) \cup \text{TopK}_{\text{embed}}(q, D)$$

In Hybrid retrieval, keyword retrieval preserves precision for financial terminology, while semantic retrieval captures paraphrased or contextual references.

The choice of the TopK value represents a critical hyperparameter in this stage: setting it too low risks

restricting the candidate pool passed to subsequent phase, while setting it too large diminishes the value of initial retrieval.

3.5 Phase 2: Adaptive Retrieval Controller & Utility-Grounded Re-ranking

Not all retrieved passages are useful for credit analysis. Financial reports contain boilerplate disclosures, legal notes, and narrative sections that are semantically related to a query but irrelevant for underwriting decisions.

To address this issue, we introduce an adaptive retrieval controller that evaluates candidate passages using query intent and document structure.

Given candidate passages $C_0 = \{p_1, ..., p_k\}$, the controller predicts relevance and support for a given passage:

$$Rel_i = f(q_{\text{statement}}, p_i, m_i)$$

$$S_i = f(q_{\text{statement}}, p_i, m_i)$$

$$U_i = f(q_{\text{statement}}, p_i, m_i)$$

where $q_{\text{statement}}$ is a query supplemented with its corresponding financial statement, p_i is the candidate passage, m_i represents structural metadata, Rel_i represents relevancy (boolean) of candidate passage, S_i represents evidence support (boolean) of candidate passage, and represents utility score (numerical score) of candidate passage.

The controller is implemented using a lightweight language model that evaluates whether the passage is likely to contain relevant and supportive information.

This stage produces a refined candidate set:

$$C_1 = \{p_i \in C_0 \mid (S_i) \cdot [Rel_i]\}$$

where $S_i \cdot [Rel_i]$ represents the logical condition that S_i is effectively nullified unless Rel_i succeeds.

This mechanism is conceptually related to adaptive retrieval strategies that condition retrieval on model reasoning rather than fixed pipelines [7].

The remaining candidates are ranked according to decision utility using an LLM-as-a-Judge framework. Rather than measuring similarity to the query, the judge evaluates each passage based on its usefulness for underwriting decisions. Given a passage p_i, the judge produces a utility score:

$$J_1 = \{p_i \in C_1 \mid U_i \geq U_{\text{threshold}}\}$$

where U_i is the utility score and $U_{\text{threshold}}$ is the tunable utility score threshold. $U_{\text{threshold}}$ serves as a critical hyperparameter in the architecture, balancing precision and recall in the final evidence set. A lower $U_{\text{threshold}}$ increases recall by admitting a larger set of passages, which is advantageous for exploratory queries or scenarios where analysts require broad and comprehensive coverage. Conversely, a higher $U_{\text{threshold}}$ prioritizes precision, returning only passages with the highest decision utility which is ideal for targeted questions where concise evidence is needed.

The utility-grounded re-ranking mechanism allows the system to prioritize passages containing financial indicators and industry signals.

3.6 Context-Aware Evidence Extraction

Financial documents frequently contain complex tables, footnotes, and structured subsections. Simple chunk extraction can distort meaning, break structural relationships, or cause loss of attribution. We therefore introduce a context-aware extraction module that dynamically selects the appropriate extraction strategy based on document structure.

Two extraction modes are employed depending on the structure of the source content. For narrative sections, relevant text spans are extracted using markdown-aware segmentation, which preserves structural elements such as section headers, bullet lists, and paragraph boundaries.

When information appears in tables or structured financial statements, the system distinguishes between complex and non-complex tables. Non-complex tables (tables with single-level headers and regular grid structures) are parsed to extract the relevant rows or cells. In contrast, complex tables containing multi-level headers, hierarchical indices, merged cells, or irregular layouts are preserved along with source metadata, including the document name and page reference, to support manual verification and accurate source attribution.

This approach ensures that extracted financial metrics remain interpretable and can be reliably traced back to their original document context, while accommodating the diversity and structural complexity of tables across different reports.

4. Results

We evaluated the system on a multilingual corpus of financial documents with relevance labels curated by credit analysts. Compared to traditional naive retrieval systems, the proposed system significantly improves retrieval performance.

Due to the proprietary nature of the financial documents used in this study, evaluation was conducted in a restricted enterprise environment. The dataset consists of internal corporate credit documents containing confidential financial information; therefore, the annotated dataset and detailed case studies cannot be publicly released.

To support meaningful evaluation, financial statements and contextual materials were curated by senior credit analysts based on real underwriting tasks. Queries reflect typical credit assessment questions, such as identifying balance sheet components, income statement items, and industry outlook indicators.

In real-world deployment across more than 800 analysts, the system reduced document review time from several hours to approximately three minutes.

Acknowledgments

This work was conducted at OCBC AI Lab. The authors would like to express their sincere gratitude to Kok Ker Ern Kovan, Lee Sheng Kiat, and Germaine Goh Yanshan for their valuable contributions and support throughout the project. Their insights and domain expertise were instrumental in the successful outcomes presented in this paper. The authors would also like to specifically thank Ren Xuezhe, Lisa Tan, Weiyang Song, Qishuai Zhong, and Kenneth Loh Zhen Xiang for their significant contributions to the successful delivery and shipment of the product. Their dedication and collaboration greatly enhanced the impact and practical application of this work.

References

[1] Bhaskarjit Sarmah, Benika Hall, Rohan Rao, Sunil Patel, Stefano Pasquali, and Dhagash Mehta. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction, 2024.

[2] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Pavel Kuksa, and et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] 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, 2024.

[4] Hengran Zhang, Minghao Tang, Keping Bi, Jiafeng Guo, Shihao Liu, Daiting Shi, Dawei Yin, and Xueqi Cheng. Utility-focused llm annotation for retrieval and retrieval-augmented generation, 2025.

[5] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction, 2022.

[6] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[7] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.