Beyond Fuzzy Matching: A Dual-Augmentation RAG System for Robust Product Reconciliation in Accounting

Michail Dadopoulos * and Stratos Moschidis

Posted Date: 5 May 2026

doi: 10.20944/preprints202605.0214.v1

Keywords: accounts payable; audit trail; decision support systems; entity resolution; invoice reconciliation; retrieval-augmented generation; large language models

 doi:10.20944/preprints202605.0214.v1

Article

Beyond Fuzzy Matching: A Dual-Augmentation RAG System for Robust Product Reconciliation in Accounting

Michail Dadopoulos [1,*] and Stratos Moschidis [2]

[1] Information Technologies Institute, Centre of Research & Technology Hellas, Thessaloniki, Greece

[2] Department of Accounting and Finance, University of Macedonia, Thessaloniki, Greece

* Correspondence: mdadopoulos@iti.gr

Abstract

Accurate product-to-catalog invoice matching is a foundational internal control critical to financial oversight and audit quality, yet it is often bottlenecked by inconsistent vendor descriptions. Traditional rule-based matching fails to address this "long tail" of supplier heterogeneity, leading to costly manual reconciliation. This study presents an end-to-end system for automated invoice reconciliation. We introduce a novel "augment-both-sides" strategy: catalog entries are proactively enriched with LLM-generated keywords and synonyms before vectorization, while incoming invoice line items undergo query expansion to bridge the semantic gap between vendor terminology and master data. A final LLM-based reranker applies context-aware judgment to produce highly accurate Top-3 match candidates. We evaluate this system using three diverse entity resolution benchmark datasets, Abt-Buy, Amazon-Google and Walmart-Amazon, structured to simulate real-world ERP environments. The system achieves a Top-3 Recall of 93.14% to 97.96% across all domains, effectively narrowing the search space for accounting and auditing professionals from thousands of SKUs to a precise set of candidates. These results demonstrate that the architecture functions as a highly reliable intelligent decision aid, standardizing complex reconciliations, and structuring the reconciliation task for subsequent human verification.

Keywords: accounts payable; audit trail; decision support systems; entity resolution; invoice reconciliation; retrieval-augmented generation; large language models

1. Introduction

Automated invoice processing is a core component of modern enterprise resource planning (ERP) and financial accounting systems, which aim to integrate and streamline end-to-end financial operations (Grabski et al., 2011; O'Leary, 2000). A critical, yet highly challenging, component of this workflow is the accurate reconciliation of product line items from vendor invoices against an internal corporate catalog. This reconciliation is a foundational internal control, forming the basis of the "three-way match" required for rigorous financial oversight and the prevention of fraudulent or erroneous payments. Evaluating these literal discrepancies requires professional judgment, and the consistency of that judgment is of direct interest to internal audit functions that test the reliability of the underlying controls.

This task is notoriously difficult and represents a complex "fuzzy matching" problem. Invoice product descriptions are often noisy due to Optical Character Recognition (OCR) errors, highly abbreviated, and use of heterogeneous, vendor-specific terminology, especially when derived from scanned or semi-structured documents (Cristani et al., 2018; Ha and Horák, 2022). In contrast, internal catalogs may contain structured, but lexically different, descriptions. Traditional fuzzy string-matching algorithms, which rely on character-level similarity (e.g., Levenshtein distance) or simple

 doi:10.20944/preprints202605.0214.v1

keyword-based searches, are brittle and fail in this low-signal, high-variance environment (Cohen et al., 2003).

This failure of traditional automation is a primary source of operational friction. It breaks the "touchless" processing workflow, forcing "literal discrepancies" to be routed for significant manual intervention. Such manual invoice handling (including data entry, visual inspection, reconciliation, and archiving) remains slow and costly in practice and can materially delay downstream payments and procurement operations. Under cognitive fatigue, human evaluators face a complex decision-making task that increases the risk of misclassification, thereby threatening audit quality and financial data integrity.

Recent advancements in Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) offer a new paradigm for tackling this advanced fuzzy matching problem. While Robotic Process Automation (RPA) has successfully automated deterministic, rule-based tasks, it fundamentally struggles when processes require robust interpretation of unstructured and variable text (Huang and Vasarhelyi, 2019; Ng et al., 2021), and real-world deployment in purchasing/procurement highlights both practical potential and implementation barriers (Flechsig et al., 2022). In this broader context of procurement digitization and automation (Bode et al., 2023; Strohmer et al., 2020), we present an end-to-end system that leverages LLMs for robust, scalable invoice-to-catalog matching. Our core contribution is a novel "augment-both-sides" strategy:

1. Catalog Augmentation: We first proactively enrich the internal corporate catalog. An LLM generates additional keywords, synonyms, and potential invoice-variants for each product, and these enhanced entries are stored as embeddings in a vector database.
2. Query Augmentation & Reranking: During live invoice processing, our system leverages an LLM to generate multiple augmented query variants from the raw, extracted invoice line. This "query expansion" retrieves a broad set of potential candidates, which are then evaluated by a specialized LLM-based reranker to produce the final Top-3 matches.

Crucially, this system is designed to function as a high-efficiency decision support system, rather than a fully autonomous "black box." In a production AP workflow, the operational goal is to accelerate human review by presenting the correct match within the operator's immediate field of view. Therefore, we define our primary performance metrics as Top-1 Recall (representing the potential for fully "touchless" automation) and Top-3 Recall (representing rapid, human-verified processing). In this study we focus on matchable line items (where a correct catalog mapping exists). Detecting true non-catalog items and exception-only lines is important in practice but is out of scope for this evaluation.

This system was developed and validated in a real-world enterprise setting, processing complex and heterogeneous Greek invoices. In this production environment, the system was evaluated on approximately 200 Greek invoice line items, each manually verified by an AP analyst against the corporate catalog; the system surfaced the correct catalog entry within the Top-3 in roughly 97% of cases, and the residual failures were dominated by lines whose ground-truth catalog entry shared neither brand nor product name with the invoice description and could not in principle be matched from the line text alone. Due to the commercial sensitivity of this corporate data, we cannot publish the dataset. Therefore, to ensure academic reproducibility and benchmark our method, we evaluate the core of our matching methodology on three well-established public entity resolution datasets: Abt-Buy, Amazon-Google, and Walmart-Amazon. Our results demonstrate that this hybrid architecture achieves robust, state-of-the-art performance, validating its effectiveness as a scalable solution for enterprise accounting environments.

2. Related Work

2.1. E-Invoicing Adoption and RPA Governance

Electronic invoicing (e-invoicing) and Robotic Process Automation (RPA) have been central to the transformation of Accounts Payable (AP) within Accounting Information Systems (AIS). At the

 doi:10.20944/preprints202605.0214.v1

adoption level, empirical work using the Technology–Organization–Environment (TOE) lens shows that compatibility, complexity, relative advantage, trialability, firm size, and competitive/regulatory pressure are significant determinants of e-invoicing uptake, underscoring why many AP functions continue to operate with hybrid (paper/PDF + EDI/XML) flows and why downstream automation must tolerate heterogeneity (Tiwari et al., 2023).

Within auditing and controllership, RPA has been framed as a way to offload "well-defined, low-judgment" tasks so professionals can focus on higher-judgment work (Huang and Vasarhelyi, 2019). At the same time, newer risk work warns that RPA initiatives carry operational/controllability risks that must be explicitly rated and governed, for example via impact–uncontrollability matrices (Schlegel et al., 2024). Together, these streams explain why, despite high RPA adoption, a significant volume of residual manual reconciliation persists, necessitating more intelligent, cognitive automation approaches.

Earlier AIS/operations research on indirect procurement further clarifies why AP matching is hard: organizations use diverse B2B e-procurement processes and platforms, and "fit" varies by process archetype, driving the messy long-tail of documents and item descriptions that AP must reconcile (Kim and Shunk, 2004).

2.2. Automated Invoice Processing & Information Extraction (IE)

Invoice IE has progressed from rule/layout heuristics to multimodal deep models and industrial IDP services. Early template-free systems (Palm et al., 2017) demonstrated generalization beyond fixed forms; subsequent Visual Document Understanding (VDU) architectures fuse text, layout, and vision to improve robustness across varied layouts. Representative systems include OCRMiner (text and layout features) (Ha & Horák, 2022), semantic graph-based methods (Luo and Yu, 2024), entity-relevancy models like MatchVIE (Tang et al., 2021), and pretrain-then-finetune families such as LayoutLM/LayoutLMv3 (Huang et al., 2022; Xu et al., 2020). OCR-free transformers like Document understanding transformer (DONUT) (Kim et al., 2022) further show that strong layout with vision priors can recover structured fields without external OCR. Most recently, instruction-tuned document LLMs (Luo et al., 2024) and general-purpose VLMs (e.g., GPT-4V, LLaVA) exhibit compelling zero/few-shot capabilities and can be prompted for structured outputs (e.g., JSON) directly from page images (Liu et al., 2023; OpenAI et al., 2024).

Public evaluations such as DocILE highlight remaining pain points, abbreviations, noisy OCR, vendor-specific phrasing, and layout edge cases, especially at the line-item level (Šimsa et al., 2023). Field reports echo this: in production, the "long-tail" supplier problem (rare formats, inconsistent layout, infrequent vendors) limits pure template/RPA approaches, while transformer-based models generalize better yet still leave residual errors that must be handled downstream (Krieger et al., 2023). A complementary stream leverages structured e-invoices for downstream accounting automation (e.g., VAT/account-code classification) to reduce manual bookkeeping effort (Bardelli et al., 2020).

Commercial IDP services (Azure Document Intelligence, Amazon Textract, Google Document AI) and open-source toolchains (e.g., LlamaParse, Docling) now expose pretrained invoice parsers via APIs, offering high extraction quality on headers and line items. As both DocILE results and industry practice suggest, outputs often remain abbreviated or partially erroneous, motivating post-extraction repair. For example, retrieval-augmented pipelines that pair DONUT with RAG have been used to correct systematically mis-extracted addresses in parcel invoices, illustrating how external knowledge can reliably fix recurrent IE errors (Jeong et al., 2025).

2.3. Entity Resolution (ER) & Product Matching for Accounts Payable (AP)

The core of our pipeline addresses the entity resolution (ER) problem in accounts payable (AP), specifically matching invoice line items to items in the purchasing catalog. In procure-to-pay (P2P) workflows, this step is a well-known bottleneck in two- and three-way matching, because human-readable descriptions on supplier invoices rarely match catalog records exactly. The issue is particularly pronounced in indirect procurement, where organizations rely on multiple,

 doi:10.20944/preprints202605.0214.v1

heterogeneous B2B e-procurement systems. Historically, baseline approaches to this matching problem have relied on lexical and bag-of-words similarity methods, ranging from edit-distance-based string metrics (Cohen et al., 2003; Wagner and Fischer, 1974) to ranking models such as TF–IDF (Salton et al., 1975) and BM25 (Robertson et al., 1995). While foundational, these techniques are brittle in the presence of vendor-specific abbreviations, missing or mis-ordered attributes, multilingual text, and the OCR noise that is typical of scanned invoices in AP.

In response, supervised deep-learning approaches to entity resolution (ER) have emerged. Early neural models such as DeepMatcher and Ditto demonstrated substantial gains over lexical baselines by learning contextual representations directly from record pairs (Li et al., 2020; Mudgal et al., 2018). Subsequent work has refined these approaches by analyzing the impact of different pre-trained embeddings for both blocking and matching (Zeakis et al., 2023), investigating alternative training strategies such as supervised contrastive learning (Peeters and Bizer, 2022), and proposing new neural architectures for product matching (Mistiawan and Suhartono, 2024). A complementary pillar is candidate generation at catalog scale: semantic search and dense retrieval are increasingly used to reduce the match space before classification or re-ranking and are now widely adopted in large e-commerce catalogs (Nigam et al., 2019).

Procurement and accounts payable (AP) contexts introduce domain-specific constraints, unit and packaging variants, legacy item descriptions, multilingual text, and supplier-specific shorthand, as well as organizational requirements such as auditability and exception handling. Prior B2B tendering research on "semantic product matching" addresses product heterogeneity at scale under different governance and data-sharing arrangements and informs our treatment of cross-source variation (Mehrbod et al., 2018). Within AP specifically, two recurring themes in recent work are (1) the long tail of suppliers and invoice formats, and (2) interactive learning from users: systems that learn online from practitioner feedback to improve invoice line-item matching (Maurya et al., 2020) and multi-stage AI architectures that combine robust candidate retrieval with human-in-the-loop disambiguation and traceability for invoice exceptions (Tater et al., 2022).

These strands motivate our candidate-generation plus large language model (LLM) re-ranking design. We combine scalable semantic retrieval to cope with catalog size, with an instruction-tuned LLM re-ranker that acts as a policy enforcement layer. This layer applies AP-specific logic, such as normalizing units of measure and resolving packaging equivalences, ensuring that technical matches also make sense from a procurement and accounting perspective. In addition, we expose feedback mechanisms so that user corrections on long-tail cases are captured and can be exploited for continual model improvement.

2.4. Applying Large Language Models (LLMs) for Contextual Matching and Judgment

The emergence of Large Language Models (LLMs) offers a new paradigm for solving the complex entity resolution (ER) challenges outlined in 2.3. While some research explores using LLMs to perform zero-shot matching directly (Peeters et al., 2024; Wang et al., 2024), such "single-step" approaches can be slow, costly, and difficult to audit when applied at the scale required for enterprise AP.

Instead, our system integrates LLMs into a multi-stage process, often called a Retrieval-Augmented Generation (RAG) pipeline (Lewis et al., 2020), that is more scalable, controllable, and traceable. This design moves LLMs from being a single "black box" oracle to a specialized component that enhances traditional retrieval methods. Our "augment-both-sides" strategy uses LLMs to mimic specific forms of expert AP judgment at three distinct stages:

1. Proactive Catalog Enrichment: First, we use an LLM to "read" our internal product catalog and proactively generate realistic synonyms, common abbreviations, and alternative descriptions for each item. This is an automated form of master data enhancement. For example, "M6 Stainless Steel Hex Bolt, 10mm, 100-pack" might be enriched with terms like "SS M6 bolt," "hex 10mm," or "box of 100 bolts." This enriched data is stored in a high-speed vector database, allowing our system to anticipate the messy, inconsistent language suppliers use before their invoices even

 doi:10.20944/preprints202605.0214.v1

arrive. This concept builds on RAG research into document-side augmentation (Raina and Gales, 2024) but applies it as a practical control for data quality in a procurement context.

2. Interpreting Noisy Invoice Queries: When a noisy line item like "SS hexblt 10mm" is extracted from an invoice, it often fails to match the catalog directly. Our system uses an LLM to rewrite this ambiguous query into multiple, clearer variants (e.g., "stainless steel hex bolt 10mm," "M10 hex bolt stainless"). This step mimics an AP clerk's "best guess" at what the supplier meant to say, effectively translating vendor-specific shorthand into our internal terminology. This "query expansion" (Ma et al., 2023) is critical for handling OCR errors and vendor-specific phrasing, ensuring good candidates are found even when the initial data is poor.
3. Applying Contextual Judgment (Reranking): The first two steps retrieve a list of potential matches from the catalog. This list is then passed to a final LLM-based reranker, which acts like a senior AP professional performing a final check. It compares the original invoice line to the top candidates and re-orders them based on deep contextual understanding. This stage is crucial for resolving ambiguities that lexical methods miss, such as a "box" versus an "each" unit of measure or packaging equivalences (e.g., "10-pack" vs. "10 units"). This LLM-based reranking (Adeyemi et al., 2023) applies nuanced business logic, significantly improving the quality of the final Top-3 matches presented to the user.

Together, this multi-stage pipeline provides a scalable and robust solution. It addresses the core AP challenges of noisy data and vendor heterogeneity by embedding contextual judgment at key points, all while maintaining the high throughput and traceability required for modern AIS.

3. Materials and Methods

3.1. Research Design

Our research proposes a novel "Augment-Both-Sides" Retrieval-Augmented Generation (RAG) architecture. Unlike traditional rule-based matching engines used in legacy ERP systems, this approach leverages Large Language Models (LLMs) to bridge the semantic gap between unstructured invoice data and structured master data.

The system design consists of two distinct phases: (1) Offline Catalog Indexing, where the corporate master data is prepared and enriched; and (2) Online Runtime Processing, where incoming invoice line items are resolved against the catalog. Figure 1 summarizes the proposed augment-both-sides architecture, highlighting the offline catalog indexing phase and the online runtime invoice processing phase.

3.2. System Architecture and Implementation

3.2.1. Phase 1: Catalog Augmentation and Vector Indexing

To address the vocabulary mismatch problem, we implemented a proactive enrichment strategy during the indexing phase. As illustrated in our implementation logic, we do not simply index the raw product title. Instead, we construct a composite "chunk text" for every catalog item. This composite text concatenates the product description with an Enriched Metadata field, a set of synthetic keywords, synonyms, common abbreviations, and "invoice-style" variations generated by an LLM (gpt-5-mini). This ensures that the vector representation of the product anticipates the variable language likely to appear on vendor invoices.

The foundation of our system is a robust vector database representing the corporate product catalog. Unlike traditional relational databases used in ERPs, which rely on exact keyword matching, our system also uses Vector Space Models to capture semantic similarities, allowing relevant results even when wording differs.

We implemented a "Hybrid Search" architecture using Qdrant, an open-source vector database. This architecture combines two distinct retrieval mechanisms to maximize recall:

1. Dense Retrieval: Each enriched catalog entry is encoded with OpenAI's text-embedding-3-large model, with the output dimensionality reduced from the default 3,072 to 1,024 for storage and

 doi:10.20944/preprints202605.0214.v1

latency efficiency. Retrieval is performed by cosine similarity in this 1,024-dimensional space. This captures semantic context (e.g., understanding that "portable PC" and "laptop" are related).

2. Sparse Retrieval (BM25): In parallel, each catalog entry is indexed with a BM25 representation, produced by FastEmbed's Qdrant/bm25 model, which preserves exact-keyword signal (e.g., SKU codes such as "X1-Carbon" or "M6×10") that embeddings tend to smooth over.

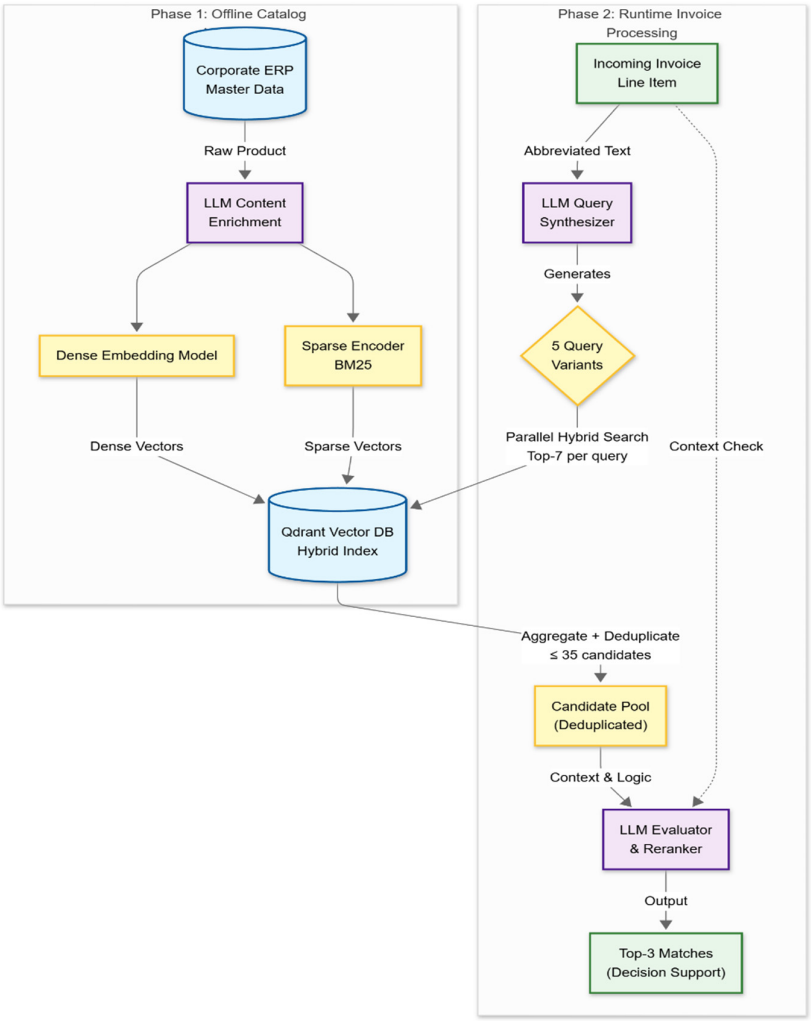

Figure 1. Two-phase "augment-both-sides" architecture for invoice-to-catalog reconciliation: (Phase 1) offline catalog enrichment and hybrid (dense+sparse) indexing. (Phase 2) runtime query expansion, hybrid retrieval, deduplication, and LLM reranking for Top-3 matches.

3.2.2. Phase 2: Real-Time Query Augmentation and Reranking

As shown in Figure 1, at runtime when an invoice line item is extracted, the system does not query the database immediately; instead, it employs a multi-step reconciliation process designed to mimic and augment the reasoning of a human accounts payable clerk. The process begins with an "Augment-Both-Sides" strategy, where an LLM acting as a "Query Synthesizer" analyzes the raw invoice text.

This step is particularly vital given the system's operation context involving hybrid Greek and English invoices, where raw descriptions are frequently noisy, abbreviated, or linguistically mixed. Consequently, the model (gpt-5-mini) generates five diverse search query variants rather than relying

 doi:10.20944/preprints202605.0214.v1

on a single query string that is susceptible to OCR errors. This approach effectively expands the search scope to account for cross-lingual phrasings, spelling errors, or formatting conventions unique to the vendor, ensuring that the semantic intent is preserved even when the syntax varies.

Following this augmentation, the system executes the five generated queries in parallel against the Qdrant vector store. For each query variant, the system retrieves the top seven matches using hybrid search. These results are then aggregated and deduplicated to eliminate redundancy, producing a consolidated "shortlist" of unique candidate matches (up to a theoretical maximum of 35 items). Achieving high recall at this intermediate stage is crucial to ensuring the correct item remains available for the final validation phase. To conclude the process, this unique shortlist is passed to a second LLM module, the "Evaluator," which acts as an automated internal control. This model functions as a logical reranker, analyzing the candidates against the original invoice line based on strict business logic, such as verifying brand consistency, matching product codes, and checking package quantities, to filter out noise and return the definitive Top-3 matches.

3.3. Data Preparation and Experimental Setup

To evaluate the robustness of our system across different domains and degrees of data noise, we utilized three standard Entity Resolution benchmark datasets[1]: Abt-Buy, Amazon-Google, and Walmart-Amazon (Köpcke et al., 2010; Mudgal et al., 2018).

We structured the data to simulate a realistic corporate procurement scenario:

- **The Corporate Catalog (Master Data):** The "right-side" dataset serves as the authorized product master file found in an ERP system.
- **The Invoice Stream (Query Set):** The "left-side" dataset was filtered to create a stream of incoming "invoice line items" that require reconciliation against the catalog.

A critical contribution of our methodology is the domain-specific preprocessing applied to these datasets to simulate the information available in a real-world ERP catalog. Since raw product data is often incomplete, we implemented logic to construct informative "Product Descriptions" for the catalog side, before the phase 1 of our system. Table 1 summarizes the dataset characteristics and the preprocessing logic used to construct the catalog-side descriptions for each benchmark. In all experiments, the full right-side dataset was treated as the corporate catalog ('Catalog Size'), while the left-side dataset was filtered to the linked/matched records used as the in-coming invoice stream ('Evaluated Queries'). Thus, each evaluated query is matched against the entire catalog.

Table 1. Dataset Specifications and Preprocessing Logic.

Dataset	Domain	Catalog Size (Rows)	Evaluated Queries	Catalog Construction Strategy (Preprocessing)
Amazon-Google	Software & Tech	3,226	1,167	**Composite Field Construction:** We concatenated the *Title* and *Manufacturer* fields. The logic explicitly checks if the manufacturer is already present in the title; if not, it is appended to ensure the embedding

[1] The three benchmarks are obtained from the official DeepMatcher dataset index at https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md (Mudgal et al., 2018). Specifically, we use the Structured/Walmart-Amazon, Structured/Amazon-Google and Textual/Abt-Buy entries from that page.

 doi:10.20944/preprints202605.0214.v1

				captures the brand identity, which is critical for matching software licenses.
Abt-Buy	Consumer Electronics	1,092	1,028	**Single Field Indexing:** This dataset contained high-quality, descriptive *Names*. We indexed the *Name* field directly as it contained sufficient signal (Brand + Model + Spec) to distinguish SKUs without additional concatenation.
Walmart-Amazon	General Retail	22,074	962	**Conditional Attribute Injection:** This was the most heterogeneous dataset. We implemented a conditional logic that analyzed the *Title*. If key attributes like *Brand* or *Model Number* were missing from the title string, they were injected from their respective columns. This mirrors the "data cleansing" phase often required in ERP migrations.

3.4. Evaluation Metrics

Consistent with the design of Decision Support Systems (DSS) in accounting, we focus on Top-k Recall for positive matches only. In a production AP workflow, correctly classifying negative pairs (non-matches) adds no value; the operational goal is to retrieve the correct GL code or SKU.

- **Top-1 Recall:** Proxies for "Touchless Automation". This measures the percentage of invoices where the system's first choice is correct, allowing for automatic posting without human review.
- **Top-3 Recall:** Proxies for "Decision Support Efficiency." This measures how often the correct match appears in the top three suggestions. If the correct code is visible immediately, the AP clerk can validate it with a single click (taking seconds) rather than searching the catalog manually (taking minutes).

4. Results

4.1. Accuracy and Robustness

The experimental results, summarized in Table 2, demonstrate the system's effectiveness across varying degrees of data noise and domain complexity. To isolate the contribution of the retrieval layer versus the full generative pipeline, we first evaluate candidate generation without query expansion or LLM reranking. Using each invoice line as a single search query, we compare sparse retrieval (BM25), dense retrieval (embeddings), and hybrid retrieval (dense+sparse fusion) on both

 doi:10.20944/preprints202605.0214.v1

the raw and the LLM-enriched catalog. We report Recall@k on positive pairs, where a hit occurs if the correct catalog item appears in the top-k retrieved candidates.

Table 2. Recall@1 and Recall@3 of retrieval baselines (raw vs. LLM-enriched catalog) and the proposed system across the three benchmarks.

Dataset	Method	Metric	Raw Catalog	Enriched Catalog	Proposed System
Amazon-Google	Dense	R@1	68.21%	70.01%	
	Sparse	R@1	65.98%	63.58%	
	Hybrid	R@1	68.04%	68.04%	**72.24%**
	Dense	R@3	90.75%	92.63%	
	Sparse	R@3	86.38%	86.29%	
	Hybrid	R@3	89.55%	92.12%	**93.14%**
Abt-Buy	Dense	R@1	86.38%	86.22%	
	Sparse	R@1	68.68%	70.14%	
	Hybrid	R@1	74.32%	79.38%	**94.07%**
	Dense	R@3	96.01%	96.38%	
	Sparse	R@3	83.95%	86.58%	
	Hybrid	R@3	91.25%	95.33%	**97.96%**
Walmart-Amazon	Dense	R@1	77.23%	75.47%	
	Sparse	R@1	77.13%	74.84%	
	Hybrid	R@1	78.38%	75.68%	**84.30%**
	Dense	R@3	94.39%	93.76%	
	Sparse	R@3	94.39%	93.76%	
	Hybrid	R@3	95.74%	95.74%	**97.30%**

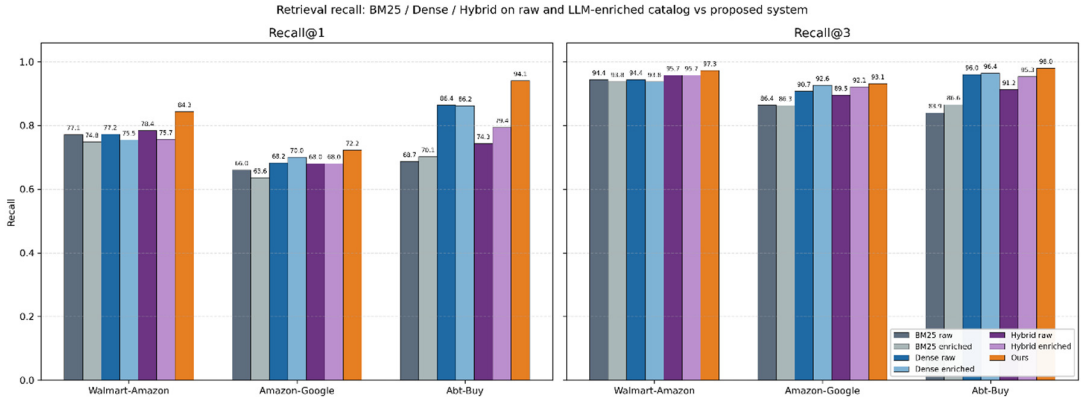

Figure 2. Recall@1 (left) and Recall@3 (right) for the three retrieval baselines under raw and LLM-enriched catalogs, and for the proposed system, across the three benchmarks. Paired colors show the effect of catalog enrichment for each retriever; the orange bar isolates the additional contribution of query expansion and LLM reranking on top of an enriched catalog. Numerical values for all bars are reproduced in Table 2.

 doi:10.20944/preprints202605.0214.v1

4.2. Analysis of Retrieval Baselines

The results in Table 2 provide a nuanced view of how different retrieval strategies handle diverse data qualities.

Across all datasets, dense retrieval (embeddings) consistently outperformed sparse retrieval (BM25) in terms of Recall@1 and Recall@3, confirming that semantic understanding is superior to keyword matching for resolving vendor descriptions. Consistent with literature, Hybrid Retrieval (combining Dense and Sparse) universally outperforms pure Sparse (BM25) retrieval. In most cases, it also outperforms pure Dense retrieval, serving as a robust baseline. A notable exception occurs in the Abt-Buy dataset. Here, pure Dense retrieval achieved an R@1 of 86.38% (Raw), surpassing the Hybrid approach (74.32%). This anomaly can be attributed to the specific nature of the Abt-Buy data, which consists of high-quality, descriptive product names. A possible explanation is that, in such "clean" environments, the semantic signal already carried by the embedding model is sufficient on its own, and the addition of a sparse channel (BM25) shifts ranking weight toward exact-token overlap on common attribute words rather than the discriminative product-name span, an effect we did not observe on the noisier Walmart-Amazon and Amazon-Google catalogs.

The impact of Phase 1 (Catalog Enrichment) varies by domain. For Amazon-Google and Abt-Buy, enrichment primarily aids in widening the net, improving Recall@3. However, for the Walmart-Amazon dataset, enrichment did not improve accuracy; in fact, R@1 dropped from 78.38% to 75.68%. This suggests that for highly heterogeneous, general retail datasets, LLM-generated keywords may introduce "semantic drift" or hallucinations that obscure the correct match, whereas they are highly effective in more specialized domains like electronics or software.

4.3. Performance of the Proposed System

Despite the variations in the baseline and enrichment layers, the full Proposed System (which adds the Phase 2 Query Synthesizer and Reranker) consistently delivers the highest performance.

- **Correction of Retrieval Errors:** Even where the retrieval layer struggled (e.g., the drop in Walmart-Amazon enrichment), the Proposed System recovered significant ground, achieving an R@1 of 84.30%.
- **Top-1 Accuracy:** The system demonstrates its capability for automation, particularly in Abt-Buy, where it achieved a dominant 94.07% R@1, significantly outperforming the best baseline (Dense at 86.38%).

This "generative lift" confirms that while retrieval models are effective at narrowing the search space, the LLM Reranker is essential for the "last mile" of disambiguation. The reranker effectively filters the noise introduced by enrichment, leveraging the generated synonyms for recall while applying strict logic for precision. A qualitative inspection of the residual top-3 misses across all three benchmarks shows that the dominant causes, text-identical catalog rows assigned to different IDs, and near-duplicate SKUs distinguished only by attributes (colour, capacity, licensing tier) absent from the invoice line, are properties of the benchmark catalogs themselves and affect every retrieval method comparably.

5. Discussion

5.1. Synthesis of Findings

This study set out to address a persistent friction in the Procure-to-Pay (P2P) cycle: the reconciliation of unstructured vendor invoice data against structured internal product catalogs. By designing and evaluating a novel "augment-both-sides" system, we demonstrated that Large Language Models (LLMs) can effectively bridge the semantic gap that traditionally causes rule-based systems to fail.

Our results across three diverse datasets indicate that the proposed system functions effectively as a high-reliability Decision Support System (DSS). While "touchless" automation capability (Top-1 recall) varied by domain, ranging from 72.24% in the highly ambiguous Amazon-Google software

 doi:10.20944/preprints202605.0214.v1

dataset to a dominant 94.07% in the clearer Abt-Buy electronics domain, the system consistently achieved a Top-3 recall exceeding 93% across all tests (peaking at 97.96%).

This finding is significant for AIS design: it suggests that even when the system cannot autonomously execute straight-through processing with sufficient confidence, it can successfully narrow the search space to a negligible set of options. This effectively eliminates the high search costs associated with manual reconciliation, transforming a complex retrieval task into a rapid validation task.

5.2. Implications for Accounting Practice and Internal Controls

Supporting Judgment and Decision-Making in Exception Handling: The high Top-3 recall facilitates a shift in AP processing from manual catalog search to a constrained verification task. By consistently presenting the correct SKU within the immediate view of the operator, the system reduces the search component of the task, which prior work in intelligent decision aids (Huang and Vasarhelyi, 2019) argues allows professionals to reallocate attention to non-standard cases. A formal behavioral evaluation of this reallocation, for instance, whether verification accuracy or speed improves when the correct item is surfaced in the Top-3, is outside the scope of the present study and is left to future work.

Enhancing Internal Controls: From an audit perspective, the "augment-both-sides" strategy acts as a robust preventative control. By standardizing the matching process through vector embeddings rather than relying on the subjective keyword searches of individual clerks, the system is intended to reduce the risk of misclassification (e.g., coding a capital asset as an expense). Furthermore, the retrieval-and-rerank architecture exposes natural points at which auditable evidence can be persisted, the five LLM-generated query variants, the deduplicated shortlist of up to 35 candidates retrieved by hybrid search, and the final top-three selections, providing a structured record of how each invoice line was reconciled rather than a single opaque decision. In the present implementation only the top-three selections are logged; persisting the upstream artefacts is a straightforward extension that we discuss as future work.

5.3. Limitations and Future Research

While the results are promising, this study is subject to limitations that frame our future research directions:

1. Linguistic Scope and Mixed-Script Complexity: A primary limitation of this study is the linguistic homogeneity of the standard benchmarks (Abt-Buy, Amazon-Google, Walmart-Amazon), which are exclusively English. This contrasts with our target operational environment, which is characterized by high linguistic entropy. In our real-world use case, data is not simply "translated"; it involves complex code-switching, where invoices and catalog entries frequently mix Greek and English terms within the same line item (e.g., an English brand name paired with a Greek functional description, or mixed-script abbreviations). While the "Query Synthesizer" demonstrated robust handling of synonyms in the benchmarks, its primary value lies in its ability to normalize this hybrid Greek-English input, a capability not fully quantified by the current English-only datasets.
2. Enrichment Trade-offs: Our experiments revealed that the "Augment-Both-Sides" strategy requires careful tuning. As observed in the Walmart-Amazon dataset, LLM-based catalog enrichment does not universally improve performance and can introduce noise (reducing R@1) in highly heterogeneous retail datasets. Future work should investigate governance mechanisms, such as confidence thresholds or "human-in-the-loop" review stages, to validate generated synonyms before they enter the vector index.
3. A formal evaluation of end-to-end latency and per-line operational cost was out of scope for this study and is left to future work; such measurements would be required before the system could be positioned as a cost-reduction intervention rather than an accuracy/decision-support one.

 doi:10.20944/preprints202605.0214.v1

Multimodal Integration: Finally, our evaluation focused on textual line items. Emerging research in multimodal transformers suggests that incorporating visual layout features (e.g., the spatial coordinates of text on the invoice) could further enhance extraction and matching accuracy, particularly for invoices where the visual structure implies the category.

5.4. Production Deployment Beyond the Benchmarks

Across the three public benchmarks, manual inspection of the residual rank-3 failures shows that the dominant single cause is a labelling artefact rather than a retrieval limitation: the Abt-Buy, Amazon-Google and Walmart-Amazon catalogs contain multiple text-identical rows assigned to different IDs (most strikingly on Amazon-Google, with 103 duplicate text-clusters spanning 300 catalog rows). A second, smaller class of residual failures consists of fine-grained variant queries where the discriminating attribute, colour, storage capacity, licensing tier, is absent from the invoice text and therefore unresolvable from the line item alone. These artefacts penalise every retrieval method in our comparison comparably, including the BM25, Dense and Hybrid baselines, and we left them in place rather than de-duplicate the catalogs so that our numbers remain directly comparable to prior work that uses the same splits. In the operational deployment on a corpus of real Greek-language vendor invoices reconciled against a production corporate catalog, where neither artefact occurs, the system reached approximately 97% top-3 reconciliation accuracy on a manually-verified evaluation of ~200 invoice line items. The English benchmarks should therefore be read as a reproducibility-oriented lower bound on the architecture's behaviour rather than as the primary evidence for production performance.

5.5. Conclusions

The automation of the AP process requires systems that are resilient to the "long tail" of supplier variability. This paper presented a semantic matching architecture that leverages the reasoning capabilities of LLMs to enrich both corporate master data and incoming invoice streams. Our empirical evaluation demonstrates that this approach achieves the high accuracy required for production ERP environments, offering a practical path toward minimizing manual reconciliation effort and modernizing the financial supply chain.

Supplementary Materials: The following supporting information can be downloaded at: Preprints.org.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Three categories of data underlie this study and are released as follows. (i) Public benchmark datasets. The three entity-resolution benchmarks evaluated in §4 (Abt-Buy, Amazon-Google, Walmart-Amazon) are obtained without modification from the official DeepMatcher dataset index at https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md (Mudgal et al., 2018), specifically the Structured/Walmart-Amazon, Structured/Amazon-Google and Textual/Abt-Buy entries. (ii) Derived files used in our pipeline. The retrieval-ready inputs (catalog.csv, catalog_enriched.csv, queries_positive.csv, pairs_positive.csv per dataset) and the per-rank outputs of every retrieval method reported in Table 2 and Figure 2 (BM25, Dense and Hybrid against both raw and LLM-enriched catalogs, plus the proposed system's top-3 predictions and per-row correctness flags in queries_positive_with_top3_eval.csv) are provided as supplementary material accompanying this article. A README.txt in the supplementary archive documents the column schema of every file and maps each cell of Table 2 to its source. These derived files were produced from the public benchmarks above by the procedures described in §3.2 and §3.3. (iii) Production deployment data. The Greek-language vendor invoice corpus and corresponding corporate catalog referenced in §1 and §5.3 cannot be released due to commercial confidentiality. Source code. The implementation of the pipeline (catalog enrichment, query synthesis, hybrid retrieval and LLM reranking) is not redistributed with this submission. The

 doi:10.20944/preprints202605.0214.v1

methodology, prompts and models required to reproduce the system are documented in §3 and Appendix A; the source code is available from the corresponding author on reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (OpenAI), Claude(Anthropic) and Gemini (Google) in order to help improve grammar and clarity, and to assist with structuring sections of the manuscript. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The system uses three LLM prompts, all served by the gpt-5-mini model with reasoning_effort set to "minimal" and verbosity set to "low" for both latency and cost control. The prompts are reproduced verbatim below from the source code; only line wrapping has been adjusted for readability.

A.1. Catalog Enricher (Phase 1 — Offline)

> You are a keyword generator that enriches clean catalog items for better fuzzy retrieval and matching against noisy invoice lines in a RAG system.
>
> Task: For each catalog entry, return only concise keywords (no sentences, no labels, no duplicates or repeat of words). The goal is to expand the catalog entry with realistic variants and terms that could appear in invoices or receipts or other catalogs, improving recall during embedding or vector similarity search.
>
> Guidelines:
>
> - NEVER invent attributes not present in the input. Do not guess colors, sizes, capacities, or brands.
> - Keep the brand EXACT if present; add common brand abbreviations only if widely used (e.g., "hewlett packard" → "hp").
> - Include synonyms for function/category if present (e.g., "headphones", "headset"; "tv", "television").
> - Add realistic invoice-style variations (e.g., abbreviations).
> - Expand common abbreviations (e.g., "Tabl → tablets", "Inj → injection", "Amp → ampoule"), but keep both expanded and abbreviated forms when relevant.
> - Keep all numeric attributes EXACT (capacity, size, version). Also add common unit variants (e.g., "gb" and "gbyte"). Expand or clarify where needed.
> - No sentences, no marketing, no stopwords, no explanations. Keep each keyword short (≤4 words).
> - Focus only on metadata that could plausibly exist for this product in catalog descriptions.
>
> Return at least 5–10 concise, search-oriented keywords that maximize retrieval accuracy between catalog data and real catalog text.

The enricher is invoked in mini-batches of five catalog entries per LLM call to amortise system-prompt overhead. The structured output is constrained to the following Pydantic schema, which requires one keyword list per entry:

Output schema (Pydantic):

```
class EnrichmentMetadata(BaseModel):
   entry1_enriched_metadata_keywords: list[str]
```

 doi:10.20944/preprints202605.0214.v1


```
    entry2_enriched_metadata_keywords: list[str]
    entry3_enriched_metadata_keywords: list[str]
    entry4_enriched_metadata_keywords: list[str]
    entry5_enriched_metadata_keywords: list[str]
```

A.2. Query Synthesizer (Phase 2 — Runtime)

You are a Query Synthesizer for Product Matching in a vector-search pipeline.

From a single product description, produce FIVE diverse, rich, high-recall queries to retrieve the best-matching product from a vector store of catalog lines.

- Do NOT invent attributes. Use ONLY info present in the input.
- Include every product/model code, sizes/dimensions, versions, and pack/count EXACTLY as they appear (when present). If absent, omit.
- Exclude invoice meta: lots, discounts, prices, VAT, order numbers, dates, addresses, loyalty/offer text, etc.
- Always preserve original script, casing, and diacritics for brand/model tokens; if you add an expansion, KEEP the original too.
- Queries must be DENSE, INFORMATIVE (no minimal queries) and meaningfully different (avoid trivial rephrasings).

DIVERSITY POLICY (pick the most helpful variations based on the product). Across the five queries, ensure you cover several of the following:

- Category and form synonyms.
- Acronym expansions or contracted/long-form variants.
- Units/number/symbol/notation formatting variants found in input (e.g., “500 mg”/”500mg”, “2 x 500 g”/”2x500g”).
- Packaging synonyms (add 1–2 besides the original).

The output is constrained to the following Pydantic schema, which returns a list of exactly five query strings:

Output schema (Pydantic):

```
class SearchQueries(BaseModel):
   search_queries: List[str] = Field(
      ...,
      description="A list of 5 different search queries"
   )
```

A.3. Evaluator / Reranker (Phase 2 — Runtime, Final Stage)

You are a Product Match Evaluator. Your goal is to identify and rank the best product matches for a given product line.

INPUTS:

(1) The original product line.

(2) Up to 35 retrieved catalog lines (duplicates possible).

TASK: Evaluate all candidates and return the top three most relevant catalog lines.

 doi:10.20944/preprints202605.0214.v1

EVALUATION PRIORITY (in order of importance):

1. Exact matches of product type or name.
2. Exact matches of brand name (brand text must match exactly if present).
3. Exact matches of product code(s) (e.g., SKU, EAN, or internal codes).
4. High textual similarity in product type or product name.
5. Exact matches or compatible values for size/dimensions or pack/count.
6. Other contextual or descriptive similarity.

INSTRUCTIONS:

- Always return the three best candidates, ranked from most to least relevant.
- If no exact matches exist, return the three closest ones based on partial or semantic similarity.
- If a candidate matches at least the product type or name, it is valid for ranking.
- Never fabricate or modify product text; use the catalog lines as provided.
- Focus on precision and meaningful relevance rather than sufficiency.

The output is constrained to the following Pydantic schema, which returns exactly three candidate strings; these are then mapped back to (id, title, description) tuples by exact match against the deduplicated shortlist, with a substring fallback for minor reformatting:

Output schema (Pydantic):

```
class Candidates(BaseModel):
   top_3_candidates: List[str] = Field(
      ...,
      description="The top 3 candidates to the OCR-extracted "
                  "product line from the vector store catalog"
   )
```

References

Adeyemi, M., Oladipo, A., Pradeep, R., & Lin, J. (2023, December 26). *Zero-Shot Cross-Lingual Reranking with Large Language Models for Low-Resource Languages*. arXiv.Org. https://arxiv.org/abs/2312.16159v1

Bardelli, C., Rondinelli, A., Vecchio, R., & Figini, S. (2020). Automatic Electronic Invoice Classification Using Machine Learning Models. *Machine Learning and Knowledge Extraction*, *2*(4), 617–629. https://doi.org/10.3390/make2040033

Bode, C., Burkhart, D., Schültken, R., & Vollmer, M. (2023). Future of Procurement. In R. Merkert & K. Hoberg (Eds.), *Global Logistics and Supply Chain Strategies for the 2020s: Vital Skills for the Next Generation* (pp. 261–276). Springer International Publishing. https://doi.org/10.1007/978-3-030-95764-3_15

Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. *IIWeb*, *3*, 73–78. https://pubs.dbs.uni-leipzig.de/dc/files/Cohen2003Acomparisonofstringdistance.pdf

Cristani, M., Bertolaso, A., Scannapieco, S., & Tomazzoli, C. (2018). Future paradigms of automated processing of business documents. *International Journal of Information Management*, *40*, 67–75. https://doi.org/10.1016/j.ijinfomgt.2018.01.010

Flechsig, C., Anslinger, F., & Lasch, R. (2022). Robotic Process Automation in purchasing and supply management: A multiple case study on potentials, barriers, and implementation. *Journal of Purchasing and Supply Management*, *28*(1), 100718. https://doi.org/10.1016/j.pursup.2021.100718

 doi:10.20944/preprints202605.0214.v1

Grabski, S. V., Leech, S. A., & Schmidt, P. J. (2011). A review of ERP research: A future agenda for accounting information systems. *Journal of Information Systems*, *25*(1), 37–78. https://publications.aaahq.org/jis/article-abstract/25/1/37/1563

Ha, H. T., & Horák, A. (2022). Information extraction from scanned invoice images using text analysis and layout features. *Signal Processing: Image Communication*, *102*, 116601. https://doi.org/10.1016/j.image.2021.116601

Huang, F., & Vasarhelyi, M. A. (2019). Applying robotic process automation (RPA) in auditing: A framework. *International Journal of Accounting Information Systems*, *35*, 100433. https://doi.org/10.1016/j.accinf.2019.100433

Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022). *LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking* (arXiv:2204.08387). arXiv. https://doi.org/10.48550/arXiv.2204.08387

Jeong, Y.-B., Seo, H., Kim, Y.-H., & Kim, W.-Y. (2025). Retrieval-augmented visual parcel invoice understanding transformer for address correction. *Engineering Applications of Artificial Intelligence*, *158*, 111542. https://doi.org/10.1016/j.engappai.2025.111542

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., & Park, S. (2022). *OCR-free Document Understanding Transformer* (arXiv:2111.15664). arXiv. https://doi.org/10.48550/arXiv.2111.15664

Kim, J.-I., & Shunk, D. L. (2004). Matching indirect procurement process with different B2B e-procurement systems. *Computers in Industry*, *53*(2), 153–164. https://doi.org/10.1016/j.compind.2003.07.002

Köpcke, H., Thor, A., & Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, *3*(1–2), 484–493. https://doi.org/10.14778/1920841.1920904

Krieger, F., Drews, P., & Funk, B. (2023). Automated invoice processing: Machine learning-based information extraction for long tail suppliers. *Intelligent Systems with Applications*, *20*, 200285. https://doi.org/10.1016/j.iswa.2023.200285

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, 9459–9474.

Li, Y., Li, J., Suhara, Y., Doan, A., & Tan, W.-C. (2020). Deep Entity Matching with Pre-Trained Language Models. *Proceedings of the VLDB Endowment*, *14*(1), 50–60. https://doi.org/10.14778/3421424.3421431

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). *Visual Instruction Tuning* (arXiv:2304.08485). arXiv. https://doi.org/10.48550/arXiv.2304.08485

Luo, C., Shen, Y., Zhu, Z., Zheng, Q., Yu, Z., & Yao, C. (2024). *LayoutLLM: Layout Instruction Tuning with Large Language Models for Document Understanding* (arXiv:2404.05225). arXiv. https://doi.org/10.48550/arXiv.2404.05225

Luo, S., & Yu, J. (2024). SGFNet: A semantic graph-based multimodal network for financial invoice information extraction. *Expert Systems with Applications*, *258*, 125156. https://doi.org/10.1016/j.eswa.2024.125156

Ma, X., Gong, Y., He, P., Zhao, H., & Duan, N. (2023). Query Rewriting in Retrieval-Augmented Large Language Models. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 5303–5315). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.322

Maurya, C. K., Gantayat, N., Dechu, S., & Horvath, T. (2020). *Online Similarity Learning with Feedback for Invoice Line Item Matching* (arXiv:2001.00288). arXiv. https://doi.org/10.48550/arXiv.2001.00288

Mehrbod, A., Zutshi, A., Grilo, A., & Jardim-Gonsalves, R. (2018). Application of a semantic product matching mechanism in open tendering e-marketplaces. *Journal of Public Procurement*, *18*(1), 14–30. https://www.emerald.com/insight/content/doi/10.1108/jopp-03-2018-002/full/html

Mistiawan, A., & Suhartono, D. (2024). Product Matching with Two-Branch Neural Network Embedding. *Journal Européen Des Systèmes Automatisés*, *57*(4). https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=12696935&AN=179548284&h=XUVxwKQFHBME89obu5I7K7Q70spwBavC2gv5K8RxxfrYCNUNy%2FEIpSOfPIXGO37dbB4f0a9%2BHaX94RHACIPDsg%3D%3D&crl=c

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., & Raghavendra, V. (2018). Deep Learning for Entity Matching: A Design Space Exploration. *Proceedings of the 2018 International Conference on Management of Data, SIGMOD '18*, 19–34. https://doi.org/10.1145/3183713.3196926

 doi:10.20944/preprints202605.0214.v1

Ng, K. K. H., Chen, C.-H., Lee, C. K. M., Jiao, J. (Roger), & Yang, Z.-X. (2021). A systematic literature review on intelligent automation: Aligning concepts from theory, practice, and future perspectives. *Advanced Engineering Informatics*, *47*, 101246. https://doi.org/10.1016/j.aei.2021.101246

Nigam, P., Song, Y., Mohan, V., Lakshman, V., Ding, W. (Allen), Shingavi, A., Teo, C. H., Gu, H., & Yin, B. (2019). Semantic Product Search. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2876–2885. https://doi.org/10.1145/3292500.3330759

O'Leary, D. E. (2000). *Enterprise resource planning systems: Systems, life cycle, electronic commerce, and risk*. Cambridge university press. https://books.google.com/books?hl=en&lr=&id=7fzMFG-tCmkC&oi=fnd&pg=PP11&dq=Enterprise+Resource+Planning+Systems+O%27Leary,+D.+E.+(2000)&ots=9a4Vlr0Y9P&sig=dRJS6XswJReudxUtffBskikKTNA

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2024). *GPT-4 Technical Report* (arXiv:2303.08774). arXiv. https://doi.org/10.48550/arXiv.2303.08774

Palm, R. B., Winther, O., & Laws, F. (2017). *CloudScan—A configuration-free invoice analysis system using recurrent neural networks* (arXiv:1708.07403). arXiv. https://doi.org/10.48550/arXiv.1708.07403

Peeters, R., & Bizer, C. (2022). Supervised Contrastive Learning for Product Matching. *Companion Proceedings of the Web Conference 2022, WWW '22*, 248–251. https://doi.org/10.1145/3487553.3524254

Peeters, R., Steiner, A., & Bizer, C. (2024). *Entity Matching using Large Language Models* (arXiv:2310.11244). arXiv. https://doi.org/10.48550/arXiv.2310.11244

Raina, V., & Gales, M. (2024, May 20). *Question-Based Retrieval using Atomic Units for Enterprise RAG*. arXiv.Org. https://arxiv.org/abs/2405.12363v2

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. *Nist Special Publication Sp*, *109*, 109. https://books.google.com/books?hl=en&lr=&id=j-NeLkWNpMoC&oi=fnd&pg=PA109&dq=Okapi+at+TREC-3&ots=YkE6HhAsME&sig=kDgCD0Ysml73EXihKaq8_229ZBQ

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

Schlegel, D., Fundanovic, O., & Kraus, P. (2024). Rating Risks in Robotic Process Automation (RPA) Projects: An Expert Assessment Using an Impact-Uncontrollability Matrix. *Procedia Computer Science, CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2023*, *239*, 185–192. https://doi.org/10.1016/j.procs.2024.06.161

Šimsa, Š., Šulc, M., Uřičář, M., Patel, Y., Hamdi, A., Kocián, M., Skalický, M., Matas, J., Doucet, A., Coustaty, M., & Karatzas, D. (2023). *DocILE Benchmark for Document Information Localization and Extraction* (arXiv:2302.05658). arXiv. https://doi.org/10.48550/arXiv.2302.05658

Strohmer, M. F., Easton, S., Eisenhut, M., Epstein, E., Kromoser, R., Peterson, E. R., & Rizzon, E. (2020). Digital in Procurement. In M. F. Strohmer, S. Easton, M. Eisenhut, E. Epstein, R. Kromoser, E. R. Peterson, & E. Rizzon (Eds.), *Disruptive Procurement: Winning in a Digital World* (pp. 49–76). Springer International Publishing. https://doi.org/10.1007/978-3-030-38950-5_3

Tang, G., Xie, L., Jin, L., Wang, J., Chen, J., Xu, Z., Wang, Q., Wu, Y., & Li, H. (2021). *MatchVIE: Exploiting Match Relevancy between Entities for Visual Information Extraction* (arXiv:2106.12940). arXiv. https://doi.org/10.48550/arXiv.2106.12940

Tater, T., Gantayat, N., Dechu, S., Jagirdar, H., Rawat, H., Guptha, M., Gupta, S., Strak, L., Kiran, S., & Narayanan, S. (2022). AI Driven Accounts Payable Transformation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(11), 12405–12413. https://doi.org/10.1609/aaai.v36i11.21506

Tiwari, A. K., Marak, Z. R., Paul, J., & Deshpande, A. P. (2023). Determinants of electronic invoicing technology adoption: Toward managing business information system transformation. *Journal of Innovation & Knowledge*, *8*(3), 100366. https://doi.org/10.1016/j.jik.2023.100366

Wagner, R. A., & Fischer, M. J. (1974). The String-to-String Correction Problem. *J. ACM*, *21*(1), 168–173. https://doi.org/10.1145/321796.321811

 doi:10.20944/preprints202605.0214.v1

Wang, T., Chen, X., Lin, H., Chen, X., Han, X., Wang, H., Zeng, Z., & Sun, L. (2024). *Match, Compare, or Select? An Investigation of Large Language Models for Entity Matching* (arXiv:2405.16884). arXiv. https://doi.org/10.48550/arXiv.2405.16884

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1192–1200. https://doi.org/10.1145/3394486.3403172

Zeakis, A., Papadakis, G., Skoutas, D., & Koubarakis, M. (2023). Pre-Trained Embeddings for Entity Resolution: An Experimental Analysis. *Proceedings of the VLDB Endowment*, *16*(9), 2225–2238. https://doi.org/10.14778/3598581.3598594