Entity Linking in Invoices: Reducing Large Language Model Calls by Using Knowledge-Guided Heuristic Filtering

Pedro H. Azevedo, Renato Fileto, André W. Zibetti, Simone S. Werner

[1]Dep. de Informática e Estatística – Univ. Federal de Santa Catarina (UFSC)
Campus Trindade, Caixa Postal 476 – 88.040-370 – Florianópolis – SC – Brazil

pedro.henrique.azevedo@posgrad.ufsc.br,

r.fileto@ufsc.br, andre.zibetti@ufsc.br, simone.werner@ufsc.br

Abstract. *Accurately identifying invoice items is crucial for procurement auditing, but challenging due to noisy, non-standardized descriptions. Large Language Models (LLMs) alone solve this problem with limited performance, and entail high computational costs and privacy concerns. Thus, we propose a hybrid entity linking framework that combines index-based retrieval to match entity surface names in a knowledge base, knowledge-guided heuristic filtering of EL candidates, and an on-premise LLM to solve only persistent ambiguities. Evaluated on Brazilian medicine invoices, our filters reduced LLM inferences by 49.60% and tokens per call by 42.4%. Using* `qwen2.5:7b`*, our approach achieved 84.48% accuracy, outperforming a filterless baseline by over 21%.*

1. Introduction

Identifiers such as the GTIN (Global Trade Item Number)[1] are often missing or incorrect in public procurement datasets [Heisler et al. 2025]. Consequently, identification efforts must rely on raw textual descriptions of invoice items. Information extraction from these descriptions is a significant challenge due to the massive volume of daily records, the low-context nature of the text, and a lack of standardization. Accurately resolving the entities within these descriptions provides the information required to audit public spending, detect overpricing schemes, and identify fraudulent practices, among other applications.

Entity Linking (EL) maps textual references to their corresponding entries in a knowledge base (KB) [Shen et al. 2015]. Typically, EL pipelines are composed of three primary modules [Balog 2018]: *mention detection*, which identifies text spans that potentially refer to named entities; *candidate selection*, which retrieves a relevant subset of entities for each detected mention [Tedeschi et al. 2021]; and *disambiguation*, which selects a single matching entity, or none, from the set of candidate entities [Balog 2018].

Supervised EL methods often face challenges that stem from their reliance on labeled datasets. It restricts their adaptability to new domains and their performance on entities of specific kinds [Xin et al. 2025]. These approaches can struggle with cases involving syntax errors, surface form variations, and lexical ambiguity, or scenarios that require external knowledge for disambiguation [Xin et al. 2025]. Moreover, disambiguation modules frequently fail to handle the None of the Candidates (NoC) problem—identifying when the correct entity is missing from the retrieved set—often forcing

[1]`https://www.gs1.org/standards/id-keys/gtin`

incorrect links [Zhou et al. 2024]. These limitations are particularly severe in specialized domains. In the biomedical field, for example, which is characterized by vast specific knowledge [Ye and Mitchell 2025], general-purpose models often fail due to lack of training in specific terminology [Ye and Mitchell 2025, Romero et al. 2025, Liu et al. 2024].

Recently, Large Language Models (LLMs) have redefined the state of the art in several Natural Language Processing (NLP) tasks, with their powerful semantic modeling and reasoning capabilities acquired through training on large-scale datasets [Wang et al. 2025]. Using prompt-based reasoning to generalize from limited inputs, these models bypass the reliance on extensive annotated datasets [Xin et al. 2025]. Nevertheless, LLMs face significant reliability challenges, such as hallucinations referring to entities absent from the KB [Vollmers et al. 2025] and difficulties in maintaining consistent outputs required for automation [Ding et al. 2025]. Furthermore, they often have high latency and costs for repeated calls [Xin et al. 2025]. They also struggle with context window limitations when processing large sets of candidates [Liu et al. 2024] and pose privacy risks inherent to third-party APIs that complicate compliance with regulations such as the Brazilian General Data Protection Law (LGPD).

This work introduces an EL framework for invoice item descriptions, combining domain knowledge, rules, and LLMs. To support the framework, we built a KB on medicines from historical price datasets provided by the Chamber for the Regulation of the Pharmaceutical Market (CMED)[2]. This KB was then enriched with controlled vocabularies from the Brazilian Health Regulatory Agency (ANVISA)[3], which incorporate surface forms, abbreviations, and acronyms. To handle high transaction volumes while navigating the complexities of invoice texts, we combine index-based candidate selection with a hybrid disambiguation strategy that integrates a rule-based heuristic filtering algorithm based on quantitative attributes (e.g. active ingredient concentration, quantity per package) with on-premise LLMs. The model is called only to solve persistent ambiguities. Using a small model on premises enables data privacy and efficient processing.

The approach was evaluated in experiments using real textual descriptions of invoice items from public purchases by authorities of the Santa Catarina State, Brazil. The results demonstrate that the cascading strategy mitigates computational overhead by reducing the volume of LLM inference calls by 49.60%. Furthermore, while `gpt-oss:20b` achieved the highest absolute accuracy (86.56%), `qwen2.5:7b` demonstrated an optimal balance between performance (84.48% accuracy) and operational efficiency (2.77s average latency).

In the remainder of this paper, Section 2 discusses related works, Section 3 details the proposed approach, Section 4 reports the experiments, Section 5 presents and discusses the results, and Section 6 concludes the paper.

2. Related Works

Recent studies have exploited hybrid EL pipelines. Some approaches employ LLMs solely as context augmenters to provide enriched information about the entities to traditional EL models. *LLMaEL* [Xin et al. 2025] feeds the generated descriptions into EL

[2]`https://www.gov.br/anvisa/pt-br/assuntos/medicamentos/cmed/precos`
[3]`https://bibliotecadigital.anvisa.gov.br/jspui/handle/anvisa/360`

systems such as ReFinED [Ayoola et al. 2022], while [Vollmers et al. 2025] utilize Llama 3 [Touvron et al. 2023] for context augmentation, followed by a *Seq2Seq* model for the final ranking. On the other hand, the *REAL* framework [Shlyk et al. 2024] proposes a RAG paradigm for the biomedical domain. To avoid a specific training step, a bio-ontology is converted into embeddings and indexed to retrieve the top-k entities based on the mention's definition; then, an LLM identifies the best matching candidate from the retrieved set through multiple-choice selection. The *ARTER* [Li et al. 2025] hybrid architecture leverages the *ReFinED* module to generate an initial pool of candidates, alongside an adaptive routing module. In this module, a lightweight language model generates a confidence score which, combined with embedding-based cosine-similarity signals, is fed into a *Random Forest* classifier to categorize mentions as either "easy" or "hard". Only the "hard" mentions are routed to the LLM, while the "easy" ones are handled by the *ReFinED* model.

Other approaches use LLMs only for disambiguation. [Ye and Mitchell 2025] uses models like *SapBERT* and *ArboEL* for candidate generation and LLMs for disambiguation through in-context learning with a RAG-like system to retrieve relevant training examples. [Albuquerque et al. 2025] proposes the I^2E framework to link medication mentions in invoice descriptions to a domain-specific KB. It retrieves matching entities using a textual indexing mechanism to handle real-world data variations and minimize computational costs. The retrieved candidates are enriched with contextual attributes before invoking the LLM strictly for final zero-shot disambiguation. However, their original proposal does not filter candidate entities for reducing LLM calls and their average number of tokens, as we do in this work to further reduce costs and improve performance.

[Liu et al. 2024] proposes *OneNET*, a framework for fine-tuning-free in three steps: (i) summarizing entity descriptions and filtering out irrelevant candidates, (ii) integrating contextual cues and intrinsic prior knowledge, and (iii) applying a consistency algorithm to reduce hallucinations. In contrast, [Rynkiewicz et al. 2025] adopted a representation-centric approach with *Universal EL*, fine-tuning an LLM to generate structured entity profiles which are subsequently vectorized for retrieval, offering versatility across knowledge bases without requiring retraining for new entities.

The reviewed literature shows that LLMs are increasingly used as complementary components rather than standalone end-to-end solvers, but significant trade-offs persist. Context augmentation strategies effectively bridge semantic gaps but remain susceptible to hallucination and "topic drift" [Vollmers et al. 2025]. Conversely, architectures prioritizing targeted reasoning – such as *OneNet* [Liu et al. 2024] – resolve structural bottlenecks but often incur prohibitive latency and computational costs. While routing systems like *ARTER* [Li et al. 2025] attempt to mitigate these inefficiencies via third-party APIs, they introduce issues regarding data privacy and infrastructure dependence. Thus, despite recent advances, existing approaches remain ill-suited for high-volume, low-context transactional environments, and regulated domains requiring strict data sovereignty.

3. Proposed Approach

Our proposal comprises two main interdependent modules: the construction of a domain-specific KB and an EL pipeline. As illustrated in Figure 1 (top), the KB is built by an Extract, Transform and Load (ETL) workflow that integrates official medicine reg-

istries with CMED pricing data. To address the unstructured nature of the CMED "apresentação" (presentation) field, we leverage the ANVISA controlled vocabulary to identify and normalize abbreviations and accurately extract key attributes: pharmaceutical form, concentration, and quantity per pack. The processed data are stored and grouped into unique therapeutic entities defined by the tuple ⟨active ingredient, pharmaceutical form, concentration⟩. This aggregation condenses the 26,504 medicine registries into 5,078 groups in the KB, which are then indexed using Solr to support the EL pipeline.

The subsequent EL pipeline (Figure 1, bottom) filters medicine invoice items using NCM (*Nomenclatura Comum do Mercosul*) codes – a regional standard for Mercosul goods classification – and normalizes item descriptions. Afterwards, the `Candidate Retrieval` task adapts the approach of the I^2E framework [Albuquerque et al. 2025] to query the indexed KB and retrieve candidate entities. For the subsequent `Disambiguation` phase, we utilize a heuristic filter to reduce candidate-entity sets, reserving the reasoning capabilities of on-premise LLMs exclusively to solve persistent ambiguities. The KB construction and indexing is done just once, and it is periodically updated with new medicine records made available in the data sources, while only the EL portion of our pipeline is repeatedly used at runtime, ensuring high-throughput.

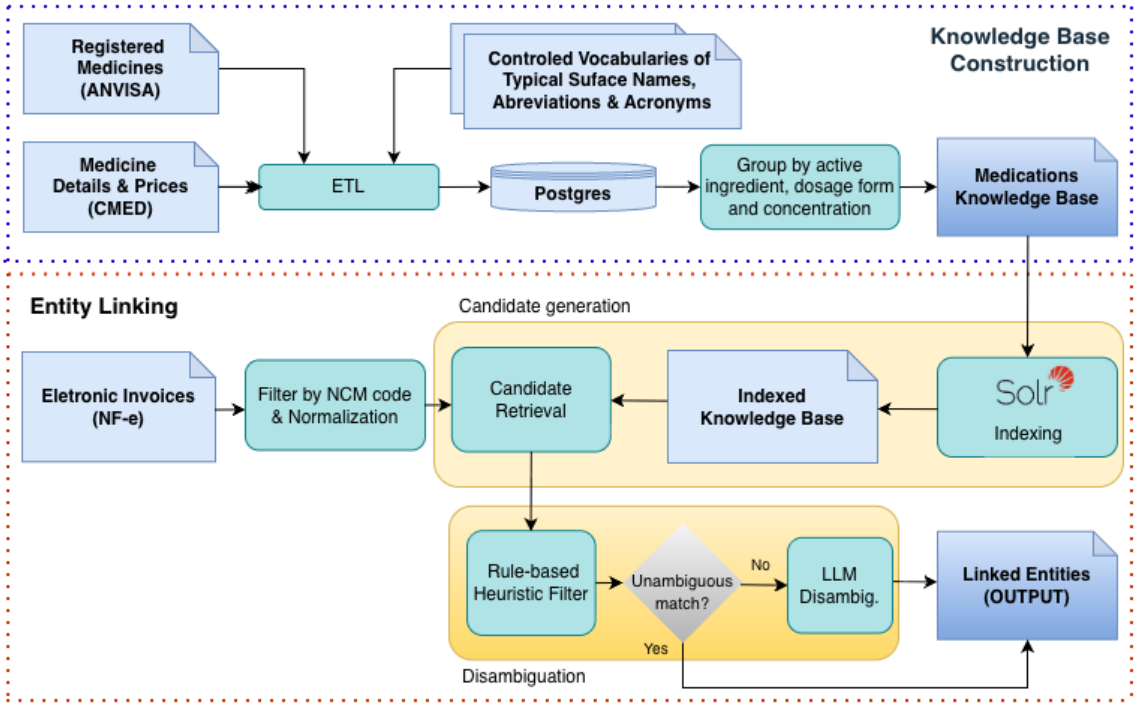

Figure 1. Overview of the proposal.

3.1. Knowledge Base Construction

This component transforms raw regulatory data into structured resources to help with the accurate disambiguation of invoice descriptions. The primary data source is the CMED price list, comprising 26,513 records across 26,504 distinct medicine presentations. Critical differentiation attributes – specifically dosage form, concentration, quantity, and packaging – are embedded within unstructured text strings in the presentation field. To address

this structural deficiency, the ETL workflow leverages custom parsing dictionaries constructed directly from the official ANVISA Controlled Vocabulary. Rather than employing simple keyword matching, the algorithm parses these unstructured strings by identifying attributes based on their spatial relationships. For instance, given the presentation string *"250 MG COM CT FR VD AMB X 25"*, the system isolates "250 MG" as the *concentration* because it precedes the first identified dictionary entity. Then, it maps the embedded abbreviations using the ANVISA dictionaries: "COM" to *Comprimido* (intake form), "CT" to *Cartucho* (secondary packaging), and "FR VD AMB" to *Frasco de Vidro Âmbar* (primary packaging). Finally, by locating the specific multiplication marker ("X"), the algorithm extracts "25" as the *quantity*, successfully structuring auxiliary information to support subsequent tasks.

The normalized records are grouped according to therapeutic equivalence, defined by the tuple ⟨active ingredient, dosage form, concentration⟩. This aggregation condenses the initial 26,504 presentations into 5,078 unique groups, significantly reducing the search space and providing a noise-free foundation. The resulting KB is stored (currently in PostgreSQL) and indexed (currently using Solr) to support the downstream candidate selection and disambiguation processes.

3.2. The Entity Linking Pipeline

The proposed EL pipeline links invoice item descriptions to KB entries through a pipeline designed for high-data-volume, low-context environments. The workflow integrates domain-specific constraints to address issues like noise and abbreviations. The major stages are: (i) *Candidate Generation* employing textual indexing with a cascading strategy to balance recall with precision; and (ii) a hybrid *Disambiguation* module initiated by an heuristic filtering, that applies KB driven rules to reduce candidates, and delegates to the LLM only remaining ambiguities. By reserving complex generative inference to persistent ambiguities, the architecture optimizes computational efficiency and performance, while maintaining on-premise data privacy.

Initially, we select medicine invoices with NCM prefix code 3004. However, as invoice item description texts frequently contain noise – such as lot numbers, dates, or internal codes – that can mislead strict matching algorithms, the raw text passes through a normalization process prior to token extraction. Thus, Numerals, units, special characters, and stopwords are removed. Date patterns and metadata prefixes like "l:" or "Val:" are also removed. This process deliberately strips quantitative attributes to construct a search query focused exclusively on the commercial name and active ingredient, reducing the noise for the index lookup.

The `Candidate Retrieval` module retrieves potential entities from the KB [Tedeschi et al. 2021] by querying an Apache Solr index. To minimize noise, the index targets the fields *commercial name* and *active ingredient*, deliberately omitting secondary attributes during the search phase. The retrieval employs a cascading strategy implemented via standard Lucene query syntax, prioritizing exact string matching to ensure high precision before addressing recall.

First, a strict intersection query (Boolean `AND`) requiring every remaining target token to match exactly is applied. If this strict constraint yields no results, the system relaxes to a union fallback (Boolean `OR`) to accommodate descriptions containing extra-

neous terms. Finally, to resolve persistent non-matches, the system resorts to a weighted hybrid expansion for the retained tokens. This final layer applies Levenshtein distance to capture mentions distorted by lexical variations, and typographical errors endemic to manual invoice data entry. Once identified, the candidate records are enriched with metadata (e.g., concentration, dosage form, quantity, manufacturer) to support the subsequent disambiguation phases.

The `Disambiguation` module aims to find the correct candidate entity for each item in the invoice. While general-purpose LLMs have advanced the state of the art in handling complex ambiguities [Xin et al. 2025], utilizing them for every inference is computationally expensive. To balance precision with operational efficiency, an heuristic filter is applied first to all candidate sets. It resolves instances for which quantitative constraints are sufficient, ensuring that the LLM is invoked strictly when ambiguity persists, and operates solely on the remaining candidates.

The heuristic filtering stage applies a quantitative filter targeting the alignment of numerical tokens (e.g., concentration, dosage quantity) between the description and the candidates' structured attributes. To maximize recall, we implement a recursive strategy (Algorithm 1). The filter initially imposes a strict constraint, requiring all extracted numbers to match the candidate's attributes. If this criterion eliminates all candidates – suggesting the presence of noise values in the description – the constraint is progressively relaxed (requiring $N - 1$ matches). This ensures that valid entities are not discarded due to irrelevant numerical tokens or missing information, allowing the system to efficiently resolve unambiguous instances where quantitative evidence is conclusive, while deferring complex cases to the subsequent stage.

Algorithm 1: Candidate Filtering

Input: D (description), C (candidates)
Output: C_{final}

```
N ← ExtractNumbers(D);
if N = ∅ then
    return C;
n ← |N|;
Function Filter(C, n):
    C_temp ← ∅;
    foreach c ∈ C do
        m ← CountMatches(c.attr, N);
        if m ≥ n then
            C_temp ← C_temp ∪ {c};
    if |C_temp| > 0 then
        return C_temp;
    else
        if n > 1 then
            return Filter(C, n − 1);
        else
            return ∅;
C_result ← Filter(C, n);
if C_result ≠ ∅ then
    return C_result;
else
    return C;                    // Fallback: retain original set if filtering fails
```

Following the heuristic filtering, the second disambiguation stage applies exclu-

sively to the remaining ambiguous cases. By employing on-premise LLMs solely for these instances that require implicit context interpretation, we minimize overall system latency and ensure data privacy. To facilitate efficient reasoning, we structure candidate lists into a nested JSON schema. At the root level, candidates are aggregated by core pharmaceutical properties: active ingredient, concentration, and dosage form. Consequently, commercial variations representing the same entity—including brand, manufacturer, and packaging—are nested as sub-items. This schema optimizes context window usage and provides a scaffold for the model to distinguish between descriptions that lack complete clinical information by using supplementary commercial details.

To navigate this structure, we employ a zero-shot prompting strategy, detailed in Figure 2, that conditions the model via a hierarchical reasoning framework [Zhou et al. 2023] mirroring clinical validation logic. The prompt instructs the LLM to execute a two-stage decision process: first, a therapeutic validation based on the root-level attributes to ensure clinical equivalence; and second, a presentation disambiguation utilizing the nested commercial details—derived from the augmentation in Section 3.1—strictly as tie-breakers. This approach prevents the prioritization of superficial lexical matches over pharmaceutical consistency, while a constrained decoding mechanism ensures the deterministic parsing of the final prediction.

```
# Role
You are an expert pharmacist. Task: Link the Description to the best Candidate group_id.

# Data Spec
- group_attributes: Shared definition (same ingredient, concentration, and form).
- presentations: Individual variations (brand, packaging, quantity).

# Logic
- Primary match: Filter candidates using group_attributes. Ensure form and unit consistency (e.g., solid vs liquid).
- Tie-breaker: If the description lacks core attributes, use presentations to identify the most specific match based on brand, packaging, or quantity.
Apply the rules carefully and select the best candidate.

# Output Format
<answer>group_id</answer>

# Input
Description: {description}
Candidates: {candidates}
```

Figure 2. Zero-shot prompt for the final disambiguation stage.

4. Experiments

Our experiments evaluate the system's efficacy across two primary dimensions: overall linking accuracy – defined as the ratio of correctly resolved medicines to the total number of descriptions – and operational efficiency. To isolate the impact of the proposed hybrid design, we conduct an ablation study comparing the framework performance with and without the rule-based heuristic filtering. This comparative analysis rigorously assesses the system across multiple metrics, including variations in linking accuracy, average token consumption, inference latency, and the effective reduction of complex LLM calls, ultimately validating the viability of deploying resource-efficient, on-premise models in high-volume administrative domains.

4.1. Datasets

This study relies on three primary data sources: (i) a ground truth dataset comprising 1,250 real-world textual descriptions extracted from public purchase invoices (NFe) in Santa Catarina, Brazil; (ii) the CMED drug price list, which serves as the authoritative target KB. This regulatory dataset contains 26,513 records detailing 26,504 distinct medicine presentations. It provides comprehensive pharmaceutical and commercial information for all medicines marketed in Brazil, including active ingredients, brand names, manufacturing laboratories, therapeutic classes, and pricing data; and (iii) the official ANVISA controlled vocabulary, used to establish the unique therapeutic entities required for fiscal auditing. As detailed in Section 3.1, the unstructured attributes within the CMED data are normalized using this vocabulary.

To establish a reliable ground truth, stratified sampling was applied to the NFe records. To facilitate the evaluation and annotation process, the selection was restricted to invoices containing a valid GTIN. From this filtered set, a subset of 1,250 distinct descriptions was selected based on character length distribution, ensuring coverage of the entire spectrum of textual complexity—from concise abbreviations to verbose descriptions. These records were manually annotated by a team of annotators, with subsequent review to resolve ambiguities and ensure inter-annotator consensus, to identify their correct CMED entity group, providing a statistically representative evaluation corpus while maintaining computational feasibility for extensive experimental runs.

4.2. Experimental Setup

To ensure data privacy and optimize processing efficiency, all experiments were conducted on a localized server at the Céos Lab. The hardware infrastructure is equipped with an AMD EPYC 9654 96-core CPU, 1.5 TB of system RAM, and operates on Linux (Kernel 6.8). All LLM inferences were orchestrated using the `Ollama` framework running on an NVIDIA RTX A6000 GPU (48 GB VRAM) with CUDA 12.8.

To evaluate the *Disambiguation* component, we selected a diverse cohort of seven open-weights models, suitable for on-premise deployment. The selection criteria focused on models optimized for instruction following and reasoning, with parameter counts strictly capped at 32 billion to ensure compatibility with standard enterprise hardware.

The experimental set spans state-of-the-art model families, including *Qwen*, *Gemma*, *Llama*, and *Phi*. To analyze the trade-off between reasoning capability and operational latency, we selected models across a diverse range of parameter sizes. This variety allows for a granular assessment of how models of different scales behave on this specific task, particularly evaluating whether smaller, faster models can achieve satisfactory disambiguation performance when aided by the proposed heuristic filtering compared to the baseline without heuristics, or if larger reasoning engines remain indispensable.

4.3. Evaluation Criteria and Metrics

The evaluation strategy focuses on three complementary dimensions: the accuracy of the linking predictions, the inference latency, and the efficacy of the pruning mechanism. Linking accuracy is defined as the proportion of test samples where the system predicts the exact Entity Group defined in the ground truth. To isolate the contributions of the deterministic and neural components, we report this metric in three scenarios: (i) *EL*

Table 1. Models selected for disambiguation.

Model	Params	Context Window	Model	Params	Context Window
Qwen 2.5	7B	128k	GPT-OSS	20B	128k
Qwen 2.5	32B	128k	Llama 3.2	3B	128k
Gemma 3	4B	128k	Phi-4	14B	16k
Gemma 3	12B	128k			

(w/o H): Represents the ablation study where the LLM processes the raw retrieval output directly, without heuristic pruning; (ii) *EL (w/ H)*: Measures the composite accuracy of the complete hybrid framework. This metric aggregates the high precision of the deterministic layers (unambiguous instances resolved by the heuristic filter) with the model's predictions on the remaining ambiguous cases; *LLM Disambiguation*: Calculates accuracy exclusively on the subset of "hard" cases (persistent ambiguities) forwarded to the model, effectively isolating the LLM's reasoning performance from the instances resolved by rules.

Regarding computational efficiency, we assess the operational viability for high-volume processing through indicators at both the per-call and dataset levels. *Average Latency* measures the mean processing time per description in seconds, while *Average Tokens* quantifies the mean token consumption per transaction. To evaluate the cumulative resource savings achieved by the heuristic filter compared to the baseline, we also report the *Total Latency* (in hours) and *Total Tokens* (in millions) processed across the entire evaluation dataset.

Finally, filtering effectiveness analyzes the reduction in context complexity by comparing the statistical distribution (Average, Median, and Maximum) of the candidate set size before and after the heuristic stage. We also report candidate recall to quantify the pipeline's stability. This metric is calculated as the ratio of instances where the correct ground truth entity is successfully retained in the candidate set to the total number of evaluated invoices ($N = 1,250$). It effectively measures the penalty incurred when the heuristic filter incorrectly excludes the true target entity, which leads to an inevitable disambiguation failure.

5. Results

This section presents the experimental data obtained from the proposed framework, followed by a critical analysis of the architectural trade-offs and system viability.

5.1. Candidate Retrieval Performance

The efficacy of the cascading retrieval strategy is evaluated based on its ability to generate a valid initial candidate pool containing the correct ground truth entity. As detailed in Table 2, the *Found* metric indicates the total number of input descriptions routed to each specific strategy, while *Correct* represents the absolute number of cases where the true target was successfully included in the search results.

The Solr index successfully returned at least one candidate for 1,248 out of 1,250 test instances (99.84%), with only 2 queries failing entirely. The distribution of these queries highlights the noisy nature of administrative texts. The strict intersection query

(AND) processed 544 mentions with the highest precision of 98.35% and an initial recall of 42.80%. However, the majority of the instances (681 cases) required the relaxed union fallback (OR) to bypass extraneous terms, maintaining a robust precision of 96.33% and increasing the cumulative recall to 95.28%. Furthermore, the Levenshtein similarity expansion was triggered for 23 severely distorted mentions that would have otherwise caused immediate pipeline failure, successfully recovering the correct entity in 21 of them (91.30% precision). Overall, the cascaded retrieval achieved an overall recall of 96.96% (1,212 out of 1,250 cases) with 97.12% precision.

Table 2. Performance of the cascading strategies for candidate retrieval.

Retrieval Strategy	Found	Correct	Precision	Recall	Cum. Recall
Strict Intersection (AND)	544	535	98.35%	42.80%	42.80%
Union Fallback (OR)	681	656	96.33%	52.48%	95.28%
Levenshtein Fallback	23	21	91.30%	1.68%	96.96%
Overall	(2 not found) **1,248**	**1,212**	**97.12%**	**96.96%**	**96.96%**

5.2. Impact of the Heuristic Filtering

The impact of the cascaded filtering strategy on the test set ($N = 1,250$) is detailed in Table 3. The deterministic layers resolved a significant portion of the input stream: the Candidate Retrieval phase identified 122 unambiguous cases, and the Heuristic Filtering module resolved an additional 498 cases. Consequently, the volume of data forwarded to the LLM was reduced to 630 instances, representing 50.40% of the total dataset.

Table 3 shows that the heuristic filter compressed the average candidate set size from 16.30 to 2.53 (a reduction of 84.5%), with the median dropping from 5 to 1. However, this aggressive reduction in candidate volume resulted in a minor penalty to the system's coverage. The precision decreased from 96.96% in the raw search to 94.16% after the heuristic filtering stage.

Table 3. Impact of the proposed heuristics on candidate set sizes and precision.

Method	Avg.	Median	Max	Precision	Recall
`Search Only`	16.30	5.00	968	97.12%	96.96%
`Search + Heur.`	2.53	1.00	63	94.16%	94.17%
Δ (%)	**-84.5%**	**-80.0%**	**-93.5%**	**-2.96 p.p.**	**-2.79 p.p.**

5.3. EL Performance

Table 4 summarizes the overall EL accuracy, per-call accuracy(LLM Disamb.) and efficiency metrics across all evaluated models. The `gpt-oss:20b` model achieved the highest absolute accuracy (86.56%), closely followed by `qwen2.5:32b` (86.24%) and `qwen2.5:7b` (84.48%). In terms of efficiency, smaller models such as `Gemma3:4b` and `llama3.2:3b` recorded the lowest latencies (2.43s and 3.42s, respectively). The specific accuracy on the ambiguous subset (*LLM Disambig.*) ranged from 52.54% (`llama3.2:3b`) to 78.31% (`gpt-oss:20b`).

To isolate the impact of the heuristic filtering, Table 5 details the ablation study comparing the baseline (*w/o H*) with the hybrid approach (*w/ H*). The inclusion of

Table 4. Overall performance metrics. EL: End-to-end Entity Linking accuracy; LLM Disambig.: Accuracy on the ambiguous subset. Latency and token counts are measured per model call.

	Accuracy		Avg Latency (s)	Avg Tokens
Model	**EL (w/ H)**	**LLM Disambig.**	**(per call)**	**(per call)**
`qwen2.5:7b`	84.48%	73.90%	2.77	**1311.64**
`qwen2.5:32b`	86.24%	77.63%	6.22	1382.43
`Gemma3:4b`	79.68%	63.73%	**2.43**	1330.90
`Gemma3:12b`	85.76%	76.61%	6.16	1427.10
`gpt-oss:20b`	**86.56%**	**78.31%**	8.81	2096.99
`phi4:latest`	85.76%	76.61%	11.02	1700.64
`llama3.2:3b`	74.40%	52.54%	3.42	1602.02

heuristics not only yielded significant accuracy gains—with `qwen2.5:7b` jumping from 63.20% to 84.48%—but also drastically reduced the cumulative computational load. For instance, the total context processed by `qwen2.5:7b` dropped from 1.34 million to 0.77 million tokens, and the total dataset inference latency decreased from 0.60 to 0.45 hours. This resource reduction reflects the fact that 47.04% of the dataset bypassed the LLM entirely, being resolved deterministically in near-zero time.

Table 5. Ablation study metrics. EL: End-to-end Entity Linking accuracy; w/ H vs w/o H.: Comparison between the hybrid pipeline and the retrieval baseline in terms of cumulative computational cost.

	Accuracy		Total Latency (h)		Total Tokens ($\times 10^6$)	
Model	**EL (w/ H)**	**EL (w/o H)**	**w/ H**	**w/o H**	**w/ H**	**w/o H**
`Gemma3:4b`	79.68%	49.36%	**0.40**	0.89	0.78	1.52
`qwen2.5:7b`	84.48%	63.20%	0.45	0.60	**0.77**	1.34
`qwen2.5:32b`	86.24%	84.46%	1.02	1.77	0.81	2.30
`gpt-oss:20b`	**86.56%**	85.44%	1.44	2.22	1.23	2.33

5.4. Discussion

The experimental results validate the proposed modular architecture as a viable solution for high-volume and sensitive data environments, demonstrating that a hybrid approach effectively balances accuracy with operational costs. This architecture can be generalized to other application domains by replacing the domain knowledge and adjusting the heuristic rules accordingly. Crucially, the system highlights that deterministic rules – combining index-based string retrieval with numerical heuristics – can reliably resolve simple and common cases without relying on LLMs. This deterministic resolution of 49.60% of the instances not only doubles throughput capacity but also guarantees higher explainability, confidence, and cost-effectiveness. By acting as a "funnel", the cascaded filtering reserves the computational overhead of LLMs strictly for complex, uncommon ambiguities. Although this aggressive pruning introduces a slight recall penalty, it is outweighed by the substantial gains in pipeline efficiency and precision. Within this targeted role, `qwen2.5:7b` emerges as the optimal on-premise candidate, delivering accuracy within 2.1% of the significantly larger `gpt-oss:20b` but with substantially lower resource demands. This finding underscores that parameter-efficient models, when sup-

ported by structured heuristics and clean context, excel as targeted semantic reasoners for the persistent ambiguities that quantitative rules cannot resolve.

5.5. Limitations

One primary limitation of this work is its reliance on domain-specific resources, specifically the dependency on controlled vocabularies to generate structured attributes for entity grouping. While this avoids model retraining costs, it constrains transferability to domains and jurisdictions (e.g. other countries) adhering to different standards lacking standardized taxonomies, which requires significant adaptation. Similarly, the deterministic rules that drive the filtering module are highly tailored to specific domains. The construction, maintenance, and refinement of these rule-based components require substantial expert domain knowledge. Furthermore, system performance is bounded by the KB coverage – currently encompassing approximately 26,500 distinct presentations – rendering absent entities inherently unlinkable. This is compounded by the inability to explicitly handle the NoC problem or mitigate error propagation; since the disambiguation module lacks a null-prediction mechanism, it is susceptible to generating incorrect links when the true target is absent from the filtered candidate pool.

6. Conclusion

This paper introduced a hybrid EL architecture tailored for invoices. By integrating deterministic retrieval and heuristic filtering strategy with the reasoning capabilities of on-premise LLMs, the proposed approach addresses the significant challenges of low-context text, high transaction volumes, and data privacy requirements inherent to administrative domains. Our experiments demonstrated that embedding domain knowledge—derived from regulatory standards—into a recursive quantitative filter effectively reduces the search space, pruning candidate sets by 84.5% and halving the volume of required neural inferences. Consequently, the pipeline mitigates the computational bottlenecks typically associated with LLM-driven disambiguation. The evaluation highlights that parameter-efficient models, particularly `qwen2.5:7b`, can achieve highly competitive accuracy (84.48%) while maintaining low latency when supplied with constrained, structured context, proving its viability for real-world auditing workflows.

Future work includes: (i) generalization and evaluation of the proposed architecture for entity linking on descriptive short texts in other application domains; (ii) the development of more robust mechanisms to minimize errors, specially false positives, which are very undesirable in applications like public procurement analytics; (iii) extension of the ground truth dataset to better evaluate issues like `NoC`; and (iv) implement mechnisms to periodically update the KB with new entities or variations inserted in the data sources.

Acknowledgments

We thank the Céos project and the Public Ministry of Santa Catarina State (MPSC) for their financial support to improve our infrastructure and for providing the electronic invoice datasets essential for this research. This work also falls under the goals of the National Institute of Science and Technology in Artificial Intelligence Responsible for Computational Linguistics, Information Processing and Dissemination (INCT-TILD-IAR).

References

Albuquerque, D. L., Santos, V., Nack, P., Fileto, R., and Dorneles, C. (2025). Language models are not a panacea: Combining them with domain knowledge and efficient indexes for entity linking. In *Simp. Brasileiro de Bancos de Dados (SBBD)*, pages 479–492, Porto Alegre, RS, Brasil. SBC.

Ayoola, T., Tyagi, S., Fisher, J., Christodoulopoulos, C., and Pierleoni, A. (2022). ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking. In Loukina, A., Gangadharaiah, R., and Min, B., editors, *Conf. of the North American Chapter of the ACL: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics (ACL).

Balog, K. (2018). *Entity-Oriented Search*, volume 39 of *The Information Retrieval Series*. Springer International Publishing.

Ding, Y., Poudel, A., Zeng, Q., Weninger, T., Veeramani, B., and Bhattacharya, S. (2025). Entgpt: Entity linking with generative large language models. arXiv preprint.

Heisler, G., Beckhauser, W., Santos, V., and Fileto, R. (2025). Detection of vehicle purchases in various invoices using large-scale language models. In *Anais da I Escola Regional de Aprendizado de Máquina e Inteligência Artificial da Região Sul*, pages 164–167, Porto Alegre, RS, Brasil. SBC.

Li, Y., Galimov, A., Ganapaneni, M. D., Thejaswi, P., Meng, D., Kumar, P., and Potdar, S. (2025). Leveraging the power of large language models in entity linking via adaptive routing and targeted reasoning. In Potdar, S., Rojas-Barahona, L., and Montella, S., editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 871–882, Suzhou (China). Association for Computational Linguistics.

Liu, X., Liu, Y., Zhang, K., Wang, K., Liu, Q., and Chen, E. (2024). Onenet: A fine-tuning free framework for few-shot entity linking via large language model prompting. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13634–13651, Miami, Florida, USA. Association for Computational Linguistics.

Romero, P., Han, L., and Nenadic, G. (2025). INSIGHTBUDDY-AI: Medication extraction and entity linking using pre-trained language models and ensemble learning. In Ebrahimi, A., Haider, S., Liu, E., Haider, S., Leonor Pacheco, M., and Wein, S., editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 18–27, Albuquerque, USA. Association for Computational Linguistics.

Rynkiewicz, A. A., Palma, R., and Formanowicz, P. (2025). Universal entity linking. *Engineering Applications of Artificial Intelligence*, 161:112185.

Shen, W., Wang, J., and Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Shlyk, D., Groza, T., Mesiti, M., Montanelli, S., and Cavalleri, E. (2024). REAL: A retrieval-augmented entity linking approach for biomedical concept recognition. In Demner-Fushman, D., Ananiadou, S., Miwa, M., Roberts, K., and Tsujii, J., editors, *Proceedings of the 23rd*

Workshop on Biomedical Natural Language Processing, pages 380–389, Bangkok, Thailand. Association for Computational Linguistics.

Tedeschi, S., Conia, S., Cecconi, F., and Navigli, R. (2021). Named Entity Recognition for Entity Linking: What works and what's next. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2584–2596, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

Vollmers, D., Zahera, H., Moussallem, D., and Ngonga Ngomo, A.-C. (2025). Contextual augmentation for entity linking using large language models. In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S., editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8535–8545, Abu Dhabi, UAE. Association for Computational Linguistics.

Wang, F., Tao, Z., Wang, M., Hu, M., and Bai, X. (2025). AELC: Adaptive entity linking with LLM-driven contextualization. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V., editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4313–4327, Suzhou, China. Association for Computational Linguistics.

Xin, A., Qi, Y., Yao, Z., Zhu, F., Zeng, K., Xu, B., Hou, L., and Li, J. (2025). Llmael: Large language models are good context augmenters for entity linking. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 3550–3559, New York, NY, USA. Association for Computing Machinery.

Ye, C. and Mitchell, C. S. (2025). LLM as entity disambiguator for biomedical entity-linking. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–312, Vienna, Austria. Association for Computational Linguistics.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., and Chi, E. (2023). Least-to-most prompting enables complex reasoning in large language models.

Zhou, K., Li, Y., Wang, Q., Qiao, Q., and Li, Q. (2024). GenDecider: Integrating "none of the candidates" judgments in zero-shot entity linking re-ranking. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 239–245, Mexico City, Mexico. Association for Computational Linguistics.