Combining Self-Retrieval-Augmented Generation with Divide-and-Conquer for Language Model-based Knowledge Base Construction

Jingbo He[1], Simon Razniewski[1,2,*]

[1]Technische Universität Dresden, Helmholtzstr. 10, 01069 Dresden, Germany
[2]ScaDS.AI Dresden/Leipzig

Abstract

Knowledge base construction from language models (LMs) without external retrieval presents unique challenges. Therefore, we present a hybrid, LM-only system for the LM-KBC 2025 challenge [1], which requires constructing knowledge bases using a fixed model (Qwen3-8B) without fine-tuning or external retrieval. Our method combines *Self-RAG* for general relations with a *divide-and-conquer* module specialized for *awardWonBy*. Self-RAG follows a description-first, then extraction-second design with strict output specifications (names-only or one-number-only) to reduce reliance on brittle post-hoc cleaning; numeric answers are normalized to a canonical digit form. The divide-and-conquer module aggregates candidates from constrained, names-only subqueries and filters them with a strict name validator. Evaluation uses the organizers' official *string-matching* metric. On the hidden *test* leaderboard, our system achieves the **2nd** place out of 5 participants, and improves macro-F1 from 0.212 (baseline) to 0.405 (+0.194; ∼+91.5% relative improvement), with large gains on *companyTradesAtStockExchange* (+0.339), *personHasCityOfDeath* (+0.330), and *countryLandBordersCountry* (+0.162).

Keywords

Knowledge base construction, Language models, Self-RAG, Divide-and-Conquer, LM-KBC

1. Introduction

Large language models (LLMs) [2, 3] pretrained on massive corpora have demonstrated strong capabilities in natural language understanding and generation, pushing the state of the art across a wide range of semantic tasks [4]. Beyond traditional tasks such as question answering, an increasingly compelling direction involves *extracting structured knowledge directly from the parameters of pretrained LLMs*—without relying on external databases or additional fine-tuning—to build disambiguated knowledge bases (KBs) [5, 6]

The LM-KBC (Knowledge Base Construction from LMs) challenge addresses this. In its 4th edition (2025) [1], participants must construct actual, *disambiguated* KBs for given subjects and relations using a fixed model, Qwen3-8B, *without* fine-tuning and *without* external retrieval augmentation (RAG) [3**?**]. Systems are evaluated using established KB metrics (precision, recall, and F1), with data released in two phases (train/dev and a later test set) and submissions evaluated on the CodaLab platform [**?**].

A key distinction from widely used probing benchmarks such as LAMA [5] lies in *how relation cardinalities are treated.* Prior probing settings commonly make simplifying assumptions, for example:

- **Single-answer perspective**: evaluation focuses on hitting a single gold object for a (subject, relation) pair, without requiring the system to decide whether there are zero, one, or many valid objects for that subject under the relation.
- **Surface-form matching and short objects**: early setups emphasize surface string matching (often with single-token objects), avoiding entity disambiguation and acceptance/rejection decisions.

Joint proceedings of KBC-LM and LM-KBC @ ISWC 2025
*Corresponding author.
jingbo.he@mailbox.tu-dresden.de (J. He); simon.razniewski@tu-dresden.de (S. Razniewski)
https://github.com/JingboHH/LM-KBC-ISWC-2025/ (J. He)
0009-0000-3653-2840 (J. He); 0000-0002-5410-218X (S. Razniewski)

- **Ranking rather than materialization**: systems are rewarded for ranking a gold string highly, not for producing a curated, disambiguated list of entities that can be directly materialized into a KB.

In contrast, LM-KBC 2025 explicitly removes these simplifications: a subject may stand in relation to *zero*, *one*, or *multiple* objects, and systems must output disambiguated entities accordingly [**?**]. This makes the task closer to realistic KB construction, where deciding *whether* to output anything and *how many* objects to output is integral to performance. Prior knowledge extraction methods from LLMs face several challenges. First, direct prompting approaches [5, 6] often produce inconsistent output formats, requiring brittle post-processing pipelines to extract structured answers from free-form text. Second, single-prompt extraction methods in the LM-KBC line [7, 8, 9] struggle with relations of varying cardinalities, particularly when distinguishing between zero, one, or many valid objects for a given subject–relation pair. Third, chain-of-thought and reasoning-based approaches [10, 11, 12, 13] frequently entangle explanatory text with factual answers, complicating the extraction of clean knowledge base entries.

Motivated by these challenges, we propose a hybrid system that combines *Self-Retrieval-Augmented Generation (Self-RAG)* with a *Divide-and-Conquer* strategy. For general relations (e.g., *companyTradesAtStockExchange*, *countryLandBordersCountry*, *hasArea*, *hasCapacity*, *personHasCityOfDeath*), we employ Self-RAG to elicit and calibrate model-internal knowledge via targeted entity descriptions before answer generation. For the challenging relation *awardWonBy*, we adopt a Divide-and-Conquer design that decomposes the task into smaller, model-friendly subproblems (e.g., award canonicalization, candidate winner identification, and consolidation), improving both accuracy and robustness. Our implementation and experimental setup are publicly available.[1]

Our contributions are threefold:

1. We introduce a unified hybrid strategy that couples Self-RAG with Divide-and-Conquer to address diverse relation types under LM-KBC 2025's realistic, non-simplified cardinality setting.
2. We demonstrate consistent gains over the organizer-provided baseline across multiple relations, showing that targeted description generation and task decomposition synergize to improve precision while maintaining recall.
3. We provide relation-wise analyses that illuminate when Self-RAG suffices and when decomposition is beneficial, offering practical guidance for LM-only KB construction.

2. Related Work

2.1. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) [14] augments an LLM with a non-parametric memory, retrieving passages that are fed back into the generator to increase factuality and reduce hallucinations. Recent surveys [15] systematize the rapidly growing literature, covering naive, advanced, and modular variants. **Self-RAG.** Asai et al. propose *Self-RAG* [16], letting the model decide *when* and *what* to retrieve and to critique its own outputs. We draw inspiration from this adaptive retrieval idea, but, in contrast to classical RAG, we generate *internal* entity descriptions rather than relying on an external corpus—consistent with the LM-KBC 2025 rule that forbids external RAG. While external RAG is prohibited in our setting, the self-generation principle from Self-RAG directly inspires our approach to generate internal descriptions as context for extraction.

2.2. Divide-and-Conquer Prompting

Decompositional prompting dates back to Chain-of-Thought (CoT) [10] and Least-to-Most strategies. A simple yet effective variant is *Divide-and-Conquer* (DaC) prompting. Zhang et al. analyse when DaC is

[1] https://github.com/JingboHH/LM-KBC-ISWC-2025/

theoretically beneficial and empirically validate it on arithmetic and fact verification tasks [17]. Hu et al. extend the idea to long-horizon decision making, coupling hierarchical RL with an LLM controller [18]. Our work adapts DaC to entity-centric knowledge extraction: for the notoriously hard *awardWonBy* relation we decompose the query into award canonicalisation, candidate enumeration, and consolidation. We extend these decomposition insights specifically for knowledge extraction, showing that systematic query decomposition can overcome single-prompt limitations for high-cardinality relations.

2.3. Knowledge Extraction from LLMs

Early studies such as LAMA [5] viewed knowledge extraction as single-answer probing, focusing on surface-form matching with single-token objects. While this simplified evaluation, it avoided critical challenges: determining whether zero, one, or multiple objects exist for a given subject-relation pair, and handling entity disambiguation [19, 20]. These simplifications, while useful for initial benchmarking, do not reflect the complexity of real knowledge base construction.

The LM-KBC challenge series has progressively addressed these limitations [8]. The 2022 edition [7] moved beyond single-answer assumptions, requiring systems to produce actual disambiguated entities. The 2024 challenge [9] further emphasized handling varying cardinalities and null values—challenges that directly motivate our hybrid approach. Unlike earlier probing benchmarks, LM-KBC requires systems to make explicit decisions about whether to output anything and how many objects to return, closely mirroring real KB construction scenarios.

Recent approaches have tackled these challenges through different strategies. Hu et al. introduce GPTKB [6], constructing large-scale KBs directly from LLMs through extensive materialization. While GPTKB demonstrates the feasibility of LM-only KB construction, it does not produce canonicalized relations, nor does it provide a clear evaluation setting. Other work has explored constrained decoding and structured output generation to ensure consistent formatting, though these often require model modifications unavailable in our setting.

Our system adopts the "LM-only" philosophy while targeting the stricter LM-KBC 2025 setting. We specifically address three key challenges observed in prior work: (1) output format inconsistency that necessitates brittle post-processing pipelines, (2) difficulty handling relations with varying cardinalities—from null values to hundreds of valid objects, and (3) entanglement of explanatory text with factual answers, particularly problematic when using reasoning-enhanced prompting strategies [10, 12, 11, 13]. Our hybrid approach combines targeted description generation (Self-RAG) for standard relations with systematic decomposition (Divide-and-Conquer) for high-cardinality relations, achieving robust extraction without external resources or model modifications.

3. Dataset

We use the official LM-KBC 2025 dataset, which provides subject–relation pairs across six relations. For each relation, the train/validation/test splits contain fixed sets of unique subjects (Table 1). Some relations allow *null values* (i.e., a subject may have no valid object), while others are *multi-object* (e.g., *awardWonBy*). Two relations are numeric (*hasArea*, *hasCapacity*), where objects are scalar values rather than entities.

Table 1
Number of Unique Subject Entities per Split and Special Features in LM-KBC 2025.

Relation	Train	Val	Test	Special features
countryLandBordersCountry	68	68	67	Null values possible
personHasCityOfDeath	100	100	100	Null values possible
hasCapacity	100	100	100	Object is numeric
awardWonBy	10	10	10	Many objects per subject
companyTradesAtStockExchange	100	100	100	Null values possible
hasArea	100	100	100	Object is numeric (sq. km)

4. Methodology

We propose a hybrid system that handles different relation types through specialized processing pipelines. Our approach recognizes that the six relations in LM-KBC 2025 exhibit different extraction challenges, requiring customized strategies for optimal performance.

4.1. System Architecture Overview

Figure 1 illustrates our two-pathway architecture. Given a subject-relation pair (s, r), our system makes a binary decision: *awardWonBy* relations are processed through a Divide-and-Conquer pipeline that specializes in high-cardinality enumeration tasks, while all other relations utilize a Self-RAG pipeline optimized for structured knowledge extraction.

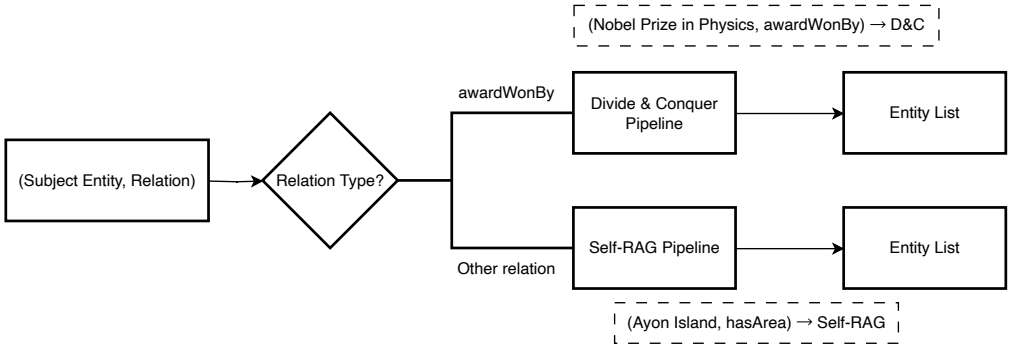

Figure 1: System Architecture Overview. Our hybrid approach passes queries through specialized pipelines based on relation type and demonstrates decisions with concrete examples.

This design choice addresses the core limitation of single-shot extraction: *awardWonBy* relations require comprehensive enumeration of large recipient sets that exceed the effective output capacity of single prompts, while simpler relations with smaller answer sets benefit from direct extraction without decomposition overhead. Our hybrid approach strategically allocates extraction complexity based on relation cardinality and the model's single-shot limitations.

4.2. Self-RAG Pipeline for General Relations

Our Self-RAG implementation adapts the retrieve-generate-critique paradigm by generating **internal entity descriptions** as retrieval substitutes, consistent with the LM-KBC 2025 constraint prohibiting external retrieval. Figure 2 details the three-phase process with concrete examples.

4.2.1. Phase 1: Context Generation

We generate relation-specific entity descriptions using carefully designed prompt templates. Each relation employs a targeted description strategy:

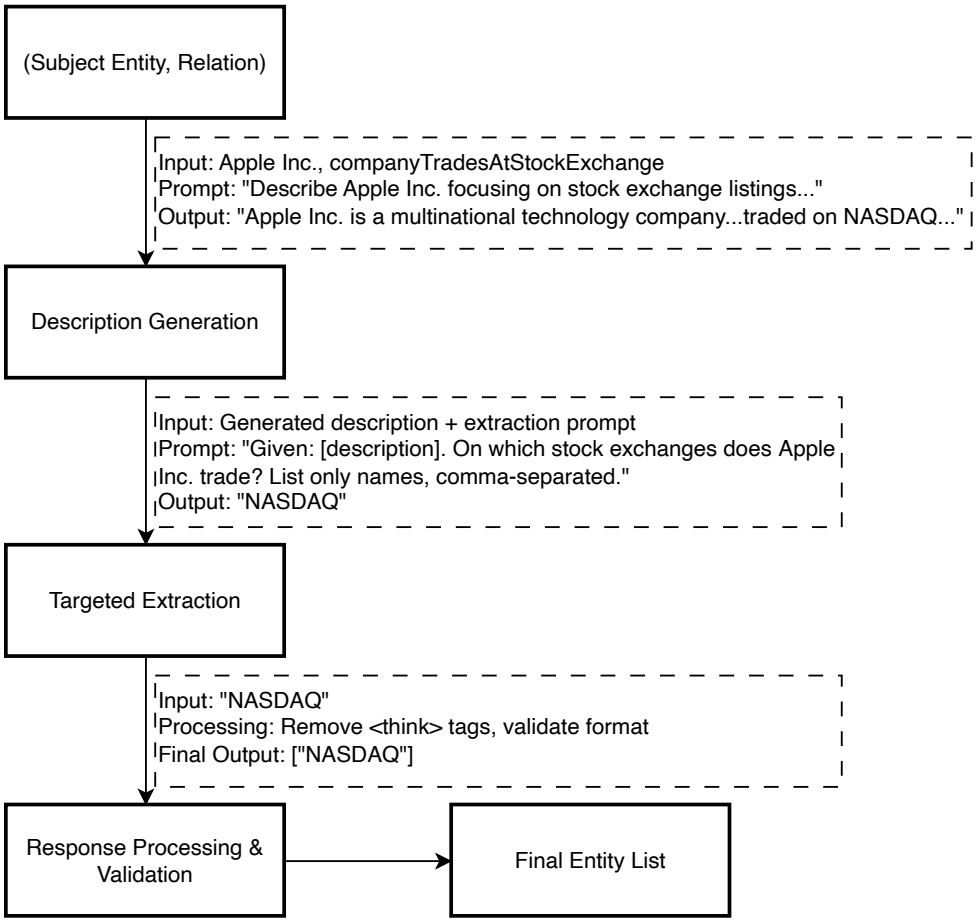

Figure 2: Self-RAG Pipeline with concrete example. Each phase shows actual prompts and outputs, demonstrating how context generation enables targeted extraction with strict output formatting.

Table 2
Self-RAG Description Generation Prompts by Relation Type

Relation	Description Prompt Template
hasArea	Describe {entity_name} with emphasis on its total area, size measurements, and spatial dimensions in square kilometers.
hasCapacity	Describe {entity_name} focusing on its maximum capacity, volume, or the number of people/items it can hold or accommodate.
companyTradesAtStockExchange	Describe {entity_name} focusing on which stock exchanges it is listed on and where its shares are traded.
countryLandBordersCountry	Describe {entity_name} focusing on which specific countries it shares land borders with and its neighboring nations.
personHasCityOfDeath	Describe {entity_name} focusing on where they died, their place of death, and the city where they passed away.

These prompts activate relevant parametric knowledge by directing the model's attention to the specific factual dimensions required for subsequent extraction

4.2.2. Phase 2: Targeted Extraction

We condition extraction queries on generated descriptions using strict format specifications that enforce direct, unambiguous outputs:

System Message (All Relations):

> *"You are a factual assistant. Provide only the requested information without explanations, uncertainty statements, or additional context. For name lists, provide only names separated by commas."*

Extraction Prompt Templates:

Table 3
Self-RAG Extraction Prompt Templates by Relation Type

Relation	Extraction Prompt Template
hasArea	Given this information about {subject_entity}: {description} What is the exact area of {subject_entity} in square kilometers? **Answer with one number only.**
hasCapacity	Given this information about {subject_entity}: {description} What is the exact capacity of {subject_entity} (How many people can it accommodate)? **Answer with number only.**
companyTradesAtStockExchange	Given this information about {subject_entity}: {description} On which stock exchanges does {subject_entity} trade? If you don't know or are uncertain, answer 'none'. Otherwise, **list all exchange names without abbreviations, separated by commas.**
countryLandBordersCountry	Given this information about {subject_entity}: {description} Which countries border {subject_entity}? If you don't know or are uncertain about the bordering countries, answer 'none'. Otherwise, **list all country names only, separated by commas.**
personHasCityOfDeath	Given this information about {subject_entity}: {description} In which city did {subject_entity} die? If you don't know or are uncertain about the city, answer 'none'. Otherwise, **answer with only one city name.**

The bold formatting requirements eliminate ambiguity and enforce direct extraction from model responses, reducing dependency on post-processing for data cleaning.

4.2.3. Phase 3: Response Processing and Validation

Our processing pipeline applies minimal cleaning operations: (1) removal of reasoning artifacts (`<think>` tags), (2) elimination of uncertainty expressions ("I'm not sure", "I don't know"), and (3) format standardization for consistent output structure. Crucially, the strict prompt design minimizes the need for extensive post-processing.

4.3. Divide-and-Conquer Pipeline for *awardWonBy*

The *awardWonBy* relation presents unique challenges: extremely high cardinality (200+ recipients for major awards), systematic explanation entanglement in single-shot outputs, and temporal complexity spanning decades. Our Divide-and-Conquer approach decomposes the enumeration task into manageable, constraint-focused subqueries. Figure 3 illustrates the complete pipeline with actual query examples.

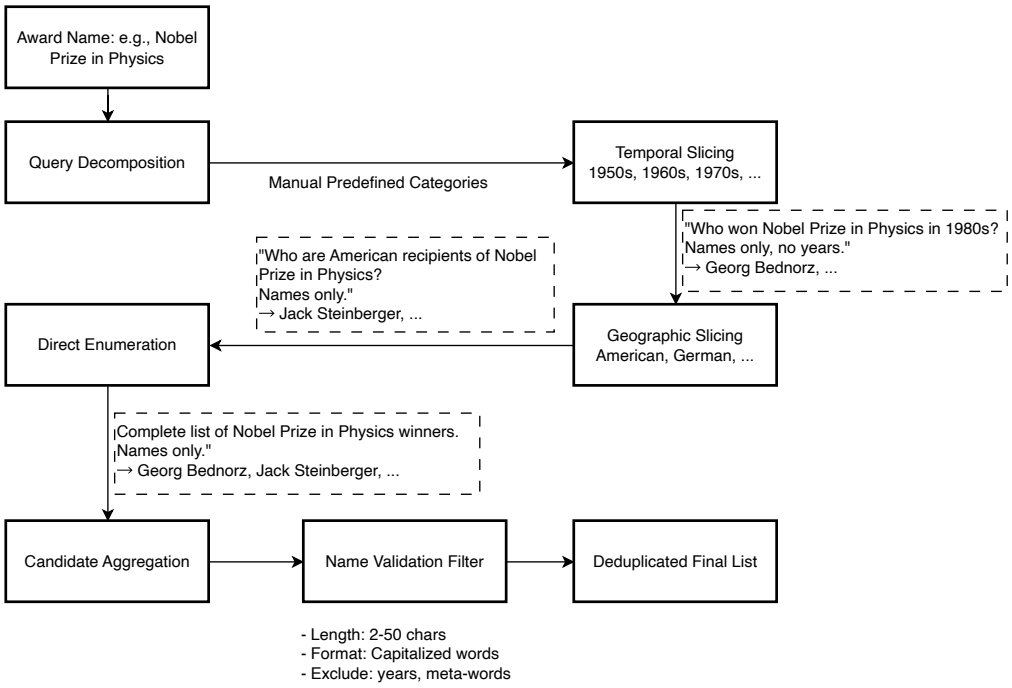

Figure 3: Divide-and-Conquer Pipeline for *awardWonBy*. The pipeline shows predefined categories and concrete query examples, with name validation ensuring high-quality candidate aggregation.

4.3.1. Query Decomposition Strategy

We employ manually predefined categories to ensure systematic coverage and reproducible results, avoiding the variability introduced by LLM-generated category schemes.

Temporal Slicing: We partition queries into eight decade-based categories: 1950s, 1960s, 1970s, 1980s, 1990s, 2000s, 2010s, 2020s. Example prompt:

> *"List all recipients of the {award_name} in the {decade}. Names only, no years, no explanations. Format: Name1, Name2, Name3"*

Geographic Slicing: We use nine predefined nationality categories: American, British, German, French, Italian, Japanese, Canadian, Chinese, plus "other" for comprehensive coverage. These categories serve as an initial implementation for decomposing queries by geographic dimension. Future work could explore data-driven or dynamic category selection based on each award's specific recipient distribution. Example prompt:

> *"List all {nationality} recipients of the {award_name}. Names only, no explanations. Format: Name1, Name2, Name3"*

Direct Enumeration: We employ five query formulations as backup strategies:

- "List the names of all {award_name} recipients. Format: Name1, Name2, Name3"
- "{award_name} winners list. Only names separated by commas."
- "Complete roster of {award_name} laureates. Names only."
- "All {award_name} recipients in chronological order. Just the names."
- "Who won {award_name}? List all names without years or descriptions."

4.3.2. Name Validation and Aggregation

We implement a strict, multi-stage validation filter with the following criteria:

Validation Rules:

1. **Length constraints:** 2-50 characters total, 1-4 words
2. **Capitalization pattern:** Each word must match `^[A-Z][a-z]*\.?$`
3. **Content exclusions:** 4-digit years, meta-words ("winner", "laureate"), institutional terms
4. **Minimum complexity:** At least one word >2 characters (excludes pure abbreviations)

This filter effectively removes false candidates while preserving valid recipient names.

4.4. Computational Analysis

Our hybrid system employs different computational strategies based on relation complexity. All experiments were conducted on NVIDIA A100-SXM4 Tensor Core GPUs (40 GB HBM2) with AMD EPYC CPU 7352 (24 cores) @ 2.3 GHz, utilizing 1 GPU and 6 CPU cores per experiment.

Table 9 presents the empirical timing analysis comparing our hybrid approach against the baseline system across all relations in the LM-KBC 2025 dataset.

Table 4
Computational Cost Analysis on LM-KBC 2025 Dataset

Method	Relations	Baseline	Our Method	Overhead
Self-RAG	Non-*awardWonBy*	2h 12m 55s	2h 27m 38s	1.11×
Divide & Conquer	*awardWonBy*	1h 8m 4s	4h 21m 48s	3.85×
Hybrid System	**All Relations**	**3h 20m 59s**	**6h 49m 26s**	**2.04×**

Self-RAG Efficiency: For the five general relations (*companyTradesAtStockExchange*, *countryLandBordersCountry*, *hasArea*, *hasCapacity*, *personHasCityOfDeath*), Self-RAG incurs only a 1.11× computational overhead despite requiring two LLM calls per subject-relation pair. This efficiency stems from the targeted nature of our prompts, which reduce the need for extensive post-processing and retry mechanisms.

Divide-and-Conquer Investment: The *awardWonBy* relation requires a substantial 3.85× computational investment, reflecting the complexity of comprehensive recipient enumeration through multiple query dimensions (8 temporal + 9 geographic + 5 direct variants). However, this targeted computational expenditure yields significant accuracy improvements for the most challenging relation in the dataset.

Strategic Resource Allocation: Our hybrid approach demonstrates strategic computational efficiency: while the overall system overhead is 2.04×, the investment is concentrated where it provides maximum benefit. The modest overhead for general relations (1.11×) combined with targeted investment for complex enumeration tasks represents an optimal trade-off between computational cost and accuracy gains.

4.5. Prompt Engineering for Direct Extraction

Our prompt design philosophy prioritizes *specification-driven generation* over post-hoc cleaning processes, addressing a key challenge in existing LM-based knowledge extraction systems: the brittleness of complex post-processing pipelines. We enforce output structure through explicit formatting instructions rather than relying on error-prone cleaning mechanisms.

4.5.1. Key Design Principles

1. Explicit Format Specifications: Every extraction prompt includes precise output format requirements tailored to the expected answer type. For numeric relations, we specify “Answer with one number

only"; for entity lists, "List only names, comma-separated"; for potential null cases, "If not applicable, answer 'None' ".

2. Proactive Uncertainty Handling: Rather than allowing the model to generate uncertain or hedged responses, we provide explicit instructions for knowledge gaps: "If you don't know or are uncertain, answer 'none' ". This directly addresses the model's tendency to provide inferential answers when facing knowledge limitations.

3. Minimalist System Messages: We employ consistent, concise system instructions across all relations: "You are a factual assistant. Provide only the requested information without explanations, uncertainty statements, or additional context." This uniform approach eliminates variability in model behavior across different relation types.

4. Reasoning Suppression: Our prompts explicitly discourage verbose explanations, uncertainty expressions, and step-by-step reasoning in final outputs. This design choice stems from our observation that models often mix factual answers with explanatory text, complicating extraction.

4.5.2. Cross-relation Generalizability

The effectiveness of our prompt engineering principles generalizes across diverse relation types and answer formats. Whether extracting single numeric values (*hasArea*), entity lists (*countryLandBordersCountry*), or handling null cases (*personHasCityOfDeath*), the specification-driven approach consistently produces directly usable outputs without relation-specific post-processing adaptations.

4.5.3. Empirical Validation

Our systematic error analysis (Section 5.3) provides empirical evidence for the effectiveness of this approach: we achieve 0% formatting failures across all sampled cases, demonstrating that specification-driven prompting successfully eliminates technical processing errors. This validates our design philosophy that prevention through careful prompt design is more reliable than correction through post-processing.

5. Evaluation

We follow the official LM-KBC 2025 evaluation protocol: scores are computed using precision, recall, and F1 metrics with exact string matching, and results are verified on the hidden test leaderboard. Our evaluation encompasses both quantitative performance analysis and qualitative error investigation to provide comprehensive insights into system behavior.

Leaderboard Status: On the hidden test leaderboard as of 2025-08-01, our system **achieves the 2nd place** (out of 5 participants), demonstrating the effectiveness of our hybrid approach on unseen test data where test labels remain private. This ranking validates our design choices across the diverse set of relations in the LM-KBC 2025 challenge.

Table 5
Macro-Averaged Performance Scores on Test Set.

System	Precision	Recall	F1
Baseline (official)	0.2272	0.4348	0.2116
Ours (Self-RAG + DaC)	**0.5234**	**0.4590**	**0.4052**
Δ (Ours − Base)	+0.2962	+0.0242	+0.1936

5.1. Quantitative Results

Performance Analysis: Our hybrid system achieves substantial improvements across all relations on the hidden test set, with macro F1 increasing from 0.2116 to 0.4052 ($\sim$+91.5% relative improvement).

The results demonstrate distinct patterns across relation types:

- **Exceptional gains on challenging relations:** *companyTradesAtStockExchange* (+0.3387) and *personHasCityOfDeath* (+0.3300) show the largest improvements, attributable to Self-RAG's description-first prompting strategy which provides crucial context for these domain-specific queries.
- **Strong performance on structured relations:** *countryLandBordersCountry* (+0.1624) demonstrates consistent improvements over an already strong baseline (0.7025), indicating that Self-RAG enhances even well-performing baseline approaches.
- **Meaningful progress on complex enumeration:** *awardWonBy* (+0.0589) benefits from our Divide-and-Conquer strategy, though the modest gain reflects the inherent difficulty of comprehensive recipient enumeration for major awards.
- **Consistent improvements on numeric relations:** Both *hasArea* (+0.0700) and *hasCapacity* (+0.0700) show identical improvements, likely due to our consistent digit-only normalization approach, though string-matching evaluation remains sensitive to precision and rounding differences.

Table 6
Per-Relation Macro F1 Scores on the Test Set.

Relation	Baseline	Ours	Δ
awardWonBy	0.1170	**0.1759**	+0.0589
companyTradesAtStockExchange	0.1670	**0.5057**	+0.3387
countryLandBordersCountry	0.7025	**0.8649**	+0.1624
hasArea	0.2400	**0.3100**	+0.0700
hasCapacity	0.0400	**0.1100**	+0.0700
personHasCityOfDeath	0.0800	**0.4100**	+0.3300
All Relations (macro)	0.2116	**0.4052**	+0.1936

Precision-Recall Trade-offs: Our system achieves a substantial precision increase (+0.2962) with minimal recall reduction (+0.0242), indicating that our approach successfully reduces false positives while maintaining coverage. This pattern suggests that our strict output specifications and validation mechanisms effectively filter unreliable predictions without sacrificing comprehensive knowledge extraction.

5.2. Strategy Selection Analysis

To validate our hybrid approach, we conducted controlled experiments comparing Self-RAG and Divide-and-Conquer strategies across different relation types. For token counting, since exact tokenization requires the use of model-specific tokenizers (e.g., OpenAI's *tiktoken*), we estimate the number of tokens in English text by assuming that one token corresponds to approximately four characters (including spaces). This heuristic follows OpenAI's official guideline, which reports that "1 token $\approx$ 4 characters of English text" [21].

5.2.1. Divide-and-Conquer Effectiveness on awardWonBy

Table 7 presents a comparative analysis between Self-RAG and Divide-and-Conquer (DaC) strategies on the *awardWonBy* relation. The results demonstrate a compelling case for using DaC on high-cardinality enumeration tasks. While Self-RAG achieves only 0.0369 F1 score, DaC reaches 0.1759, representing a **4.8$\times$ improvement**. This substantial gain justifies the increased computational cost (25.9$\times$ more tokens, 6.1$\times$ longer execution time). The low Self-RAG performance confirms that single-query approaches fundamentally cannot enumerate comprehensive recipient lists, as the model's single-response capacity limits it to returning only the most prominent recipients.

Table 7
Performance Comparison on awardWonBy Relation

Method	F1 Score	Tokens	Time (s)
Self-RAG	0.0369	13,276	2,356
Divide-and-Conquer	**0.1759**	343,508	14,454
Improvement	4.8×	25.9×	6.1×

5.2.2. Limitations of Divide-and-Conquer on Other Relations

Table 8
Strategy Performance on Medium-Cardinality Relations

Method	Relation	F1 Score
Self-RAG	*countryLandBordersCountry*	**0.8649**
	companyTradesAtStockExchange	**0.5057**
DaC	*countryLandBordersCountry*	0.6201 (-28.3%)
	companyTradesAtStockExchange	0.1287 (-74.5%)

Table 8 reveals that DaC's effectiveness is highly relation-specific. For *countryLandBordersCountry*, Self-RAG achieves 0.8649 F1 while DaC drops to 0.6201 (-28.3%). Similarly, for *companyTradesAtStock-Exchange*, Self-RAG's 0.5057 significantly outperforms DaC's 0.1287 (-74.5%). These results indicate that decomposition strategies can actually **harm performance** on medium-to-low cardinality relations.

Table 9
Computational Cost Comparison (Both Relations Combined)

Method	Total Tokens	Total Time (s)
Self-RAG	137,833	3,802
DaC	787,667	27,306
Ratio (DaC/Self-RAG)	5.7×	7.2×

As shown in Table 9, when processing both relations, DaC requires 5.7× more tokens (787,667 vs. 137,833) and 7.2× more computation time (27,306s vs. 3,802s) than Self-RAG. This substantial increase in computational resources, combined with the degraded performance, makes DaC economically unjustifiable for these relations.

5.2.3. Temporal Granularity Trade-offs

Table 10
Temporal Decomposition Granularity Analysis for awardWonBy

Strategy	F1 Score	Tokens	Time (s)
DaC (decade-based)	0.1759	343,508	14,454
DaC (year-based)	**0.1811**	1,137,784	44,565
Δ (Year − Decade)	+0.0052 (+2.9%)	3.3×	3.1×

Table 10 compares decade-based versus year-based temporal decomposition for *awardWonBy*. While year-based queries achieve marginally higher F1 (0.1811 vs. 0.1759, +2.9% relative), they require **3.3× more tokens** and **3.1× more time**. The minimal F1 improvement of 0.0052 does not justify the substantial increase in computational resources.

This analysis supports our decade-based approach as optimal for practical deployment, balancing effectiveness with efficiency. The diminishing returns from finer granularity suggest that further decomposition would yield negligible benefits while dramatically increasing costs.

5.2.4. Implications for System Design

These findings validate our hybrid architecture that applies strategies based on relation characteristics:

- **High-cardinality relations** (*awardWonBy*): Divide-and-Conquer despite computational overhead
- **Medium/low-cardinality relations**: Self-RAG for superior efficiency and accuracy
- **Temporal granularity**: Decade-based decomposition provides the best balance between coverage and cost

The results emphasize that no single strategy dominates across all relation types, reinforcing the need for adaptive, relation-aware approaches in LM-based knowledge extraction.

5.3. Error Analysis

5.3.1. Error Analysis of Self-RAG Strategy

Since test set answers are not publicly available, we conduct error analysis exclusively on validation dataset samples. We manually sample 5 entities for each relation type, focusing solely on incorrect cases to understand failure patterns. Following a systematic approach, we examine each error through four potential failure modes: (1) Self-RAG context generation issues, (2) extraction step failures, (3) formatting problems, and (4) evaluation method limitations. Detailed error cases with model outputs and gold standards are provided in Appendix B.

Table 11
Systematic Error Classification Analysis Results

Error Type	Description	Count	Percentage
Context Generation Issue	Self-RAG generates inadequate, incomplete, or inaccurate context descriptions	24	96.0%
Extraction Step Failure	Self-RAG produces necessary information, but extraction step fails	0	0.0%
Formatting Problem	Extraction is correct, but there are formatting inconsistencies	0	0.0%
Evaluation Limitation	Extraction is correct, but evaluation method fails to recognize synonyms or variations	1	4.0%
Total		**25**	**100.0%**

Table 12
Error Distribution by Relation Type and Failure Mode

Relation	Context Issue	Extraction Fail	Format Issue	Eval. Limit.
companyTradesAtStockExchange	5	0	0	0
countryLandBordersCountry	4	0	0	1
hasArea	5	0	0	0
hasCapacity	5	0	0	0
personHasCityOfDeath	5	0	0	0
Total	**24**	**0**	**0**	**1**

Systematic Analysis Findings: Our layer-by-layer error analysis reveals a clear hierarchy of failure modes:

1. **Technical implementation robustness (0% failures):** We can definitively rule out extraction step failures and formatting issues. Our response processing pipeline successfully extracts information from model outputs without introducing errors, validating the robustness of our hybrid architecture's technical components.
2. **Evaluation method appropriateness (4% limitations):** Only one case represents a pure evaluation limitation ("Ivory Coast" vs. "Côte d'Ivoire"), confirming that string-matching evaluation aligns well with semantic correctness for our task domain.
3. **Context generation as primary bottleneck (96%):** The overwhelming majority of errors stem from inadequate context generation, indicating that system improvements should focus on the initial knowledge activation phase rather than downstream processing.

Evidence from Model Reasoning Traces: Our system logs reveal the model's internal reasoning process, providing direct evidence of inferential behavior, as Appendix A shows. Despite explicit instructions to "answer 'none' if uncertain," the model rarely admits complete ignorance. For example, when queried about Hopen's area, the model's reasoning trace shows: "I need to gather accurate data... I remember that Hopen is one of the larger islands in Svalbard... From what I can find, Hopen's area is approximately 1,600 square kilometers... the consensus is 1,600." However, the actual area is 47 km^2, demonstrating a 34x overestimation.

Inferential Reasoning Pattern: This trace reveals critical behavioral patterns: (1) acknowledging uncertainty while (2) constructing plausible reasoning chains, (3) simulating source consultation, and (4) expressing false confidence in estimated answers. The model chooses to provide inferential responses rather than appropriate abstention, indicating underlying knowledge gaps compensated through sophisticated reasoning.

Implications for System Design: The error analysis confirms that our hybrid approach successfully addresses technical extraction and processing challenges, but reveals fundamental limitations in distinguishing between confident factual knowledge and inferential reasoning. Future improvements should focus on uncertainty quantification and appropriate abstention mechanisms rather than technical pipeline enhancements.

5.4. Error Analysis of Divide-and-Conquer Strategy

To understand the failure modes of our Divide-and-Conquer approach, we conduct detailed analysis using the Max Planck Medal as a representative case study. We examine both the model's reasoning process and the systematic patterns across temporal slices.

Analysis of the model's internal reasoning for the 1950s query reveals pervasive uncertainty markers throughout the extraction process. The model explicitly states "I'm not 100% certain" and uses 47 instances of self-correction ("Wait, I'm getting confused"), yet still generates four physicist names. This behavior—acknowledging uncertainty while producing confident-seeming outputs—represents a critical failure mode where decomposition provides more opportunities for plausible confabulation rather than appropriate abstention.

We evaluate outputs across three accuracy dimensions (detailed results in Table 14 in Appendix):

Layer 1 - Domain Coherence: The model maintains 100% domain accuracy across all decades, consistently generating physicist names. This demonstrates that decomposition preserves conceptual understanding of the award's domain.

Layer 2 - Award Association: Accuracy varies dramatically by era:

- Pre-1990: Mixed performance (50-100% are actual recipients, though often from wrong decades)
- Post-1990: Complete failure (0% are actual Max Planck Medal recipients)
- The model generates plausible physicists (John Bardeen, Steven Weinberg) who never received this specific award

Layer 3 - Temporal Precision: Even when identifying actual recipients, temporal placement is severely compromised. Hans Bethe (1955) appears in both 1970s and 1980s queries; Niels Bohr (1930)

appears in the 1960s query. This suggests the model has fragmented knowledge of recipients but lacks temporal grounding.

The most striking pattern is the sharp knowledge degradation around 1990. For earlier decades, the model retrieves some actual recipients despite temporal confusion. For 1990s onwards, it generates exclusively non-recipients, indicating a complete knowledge void rather than retrieval difficulty. This boundary is consistent across all temporal slices, demonstrating that decomposition cannot compensate for absent knowledge.

For 2000s and 2010s queries, the model exceeded token limits by producing verbose explanations ("I need to list all recipients of the Max Planck Medal in the 2000s...") instead of the required name-only format. This suggests that uncertainty triggers extended reasoning despite explicit format constraints, leading to extraction failures even when the query structure is identical to successful cases.

Our analysis reveals both the potential and limitations of Divide-and-Conquer:

Strengths: The strategy successfully surfaces more information than single queries might achieve. Different temporal prompts activate different memory patterns, helping retrieve recipients like Paul Dirac and Enrico Fermi who might be missed in monolithic queries.

Fundamental Limitation: Decomposition amplifies existing knowledge but cannot synthesize absent information. When the model lacks knowledge (post-1990 recipients), it confidently generates plausible but incorrect answers for each sub-query, potentially compounding errors through aggregation.

Key Insight: The effectiveness of Divide-and-Conquer is bounded by the underlying knowledge availability in the model's parametric memory. It works best for fragmented knowledge that needs assembly, not for complete knowledge voids.

6. Conclusion

We present a hybrid system for LM-only knowledge base construction that strategically combines Self-RAG for general relations with a specialized divide-and-conquer module for *awardWonBy*. Our approach addresses the core challenges of the LM-KBC 2025 setting: constructing disambiguated knowledge bases from a fixed language model without fine-tuning or external retrieval augmentation.

In the official evaluation, our hybrid system achieves substantial performance gains across all six relations, improving macro F1 from 0.212 to 0.405, and securing the 2nd place on the hidden test leaderboard. We obtain consistent improvements with particularly strong gains on challenging relations: *companyTradesAtStockExchange* (+0.339), *personHasCityOfDeath* (+0.330), and *countryLandBordersCountry* (+0.162). Our precision increase (+0.296) with minimal recall reduction (+0.024) indicates that our approach successfully filters unreliable predictions, while maintaining coverage.

We make three key contributions to LM-based knowledge extraction. First, we demonstrate that different relation types require fundamentally different extraction strategies: while Self-RAG's description-first approach excels for structured relations through targeted knowledge activation, divide-and-conquer decomposition is essential for high-cardinality enumeration tasks like *awardWonBy*. Second, we show that specification-driven prompt engineering can eliminate formatting errors entirely—our systematic error analysis reveals 0% formatting failures across all sampled cases, with 96% of errors stemming from knowledge gaps rather than technical processing issues. Third, our error analysis reveals a fundamental insight: a primary bottleneck in LM-based knowledge extraction is not the extraction, but rather the model's tendency to provide confident inferential answers when facing knowledge gaps. Through direct examination of model reasoning traces, we demonstrate that models construct plausible but incorrect responses through sophisticated reasoning rather than admitting uncertainty.

These findings suggest that uncertainty quantification and appropriate abstention mechanisms are more critical than advanced post-processing techniques for improving knowledge extraction reliability. Our hybrid approach demonstrates that strategic combination of complementary techniques can significantly advance LM-only knowledge base construction, with the insights about inferential reasoning providing a foundation for developing more reliable knowledge extraction systems that appropriately handle uncertainty in language models' parametric knowledge.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] J.-C. Kalo, T.-P. Nguyen, S. Razniewski, B. Zhang, 4th lm-kbc challenge, in: LM-KBC Challenge @ ISWC, 2025. URL: https://lm-kbc.github.io/challenge2025/.

[2] OpenAI, GPT-4 technical report, arXiv:2303.08774 (2024). URL: https://arxiv.org/abs/2303.08774.

[3] A. Yang, et al., Qwen3 technical report, arXiv:2505.09388 (2025). URL: https://arxiv.org/abs/2505.09388.

[4] T. B. Brown, et al., Language models are few-shot learners, NeurIPS (2020). URL: https://arxiv.org/abs/2005.14165.

[5] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, Language models as knowledge bases?, in: EMNLP, 2019. URL: https://aclanthology.org/D19-1250/.

[6] Y. Hu, T.-P. Nguyen, S. Ghosh, S. Razniewski, Enabling LLM knowledge analysis via extensive materialization, ACL (2025). URL: https://arxiv.org/abs/2411.04920.

[7] S. Singhania, T.-P. Nguyen, S. Razniewski, 1st lm-kbc challenge, in: LM-KBC challenge @ ISWC, 2022. URL: https://ceur-ws.org/Vol-3274/paper1.pdf.

[8] J.-C. Kalo, S. Singhania, S. Razniewski, J. Z. Pan, LM-KBC 2023: 2nd challenge on knowledge base construction from pre-trained language models, in: LM-KBC Challenge @ ISWC 2023, 2023. URL: https://ceur-ws.org/Vol-3577/paper0.pdf.

[9] S. Razniewski, T.-P. Nguyen, et al., Preface: LM-KBC challenge 2024, in: LM-KBC Challenge @ ISWC, 2024. URL: https://ceur-ws.org/Vol-3853/paper0.pdf.

[10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, NeurIPS (2022). URL: https://arxiv.org/abs/2201.11903.

[11] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, NeurIPS (2022). URL: https://arxiv.org/abs/2205.11916.

[12] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, E. H. Chi, Least-to-most prompting enables complex reasoning in large language models, ICLR (2023). URL: https://arxiv.org/abs/2205.10625.

[13] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, NeurIPS (2023). URL: https://arxiv.org/abs/2305.10601.

[14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, NeurIPS (2020). URL: https://arxiv.org/abs/2005.11401.

[15] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv:2312.10997 (2023). URL: https://arxiv.org/abs/2312.10997.

[16] A. Asai, Z. Wu, Y. Wang, A. Sil, H. Hajishirzi, Self-RAG: Learning to retrieve, generate, and critique through self-reflection, ICLR (2024). URL: https://arxiv.org/abs/2310.11511.

[17] Y. Zhang, L. Du, D. Cao, Q. Fu, Y. Liu, An examination on the effectiveness of divide-and-conquer prompting in large language models, arXiv:2402.05359 (2024). URL: https://arxiv.org/abs/2402.05359.

[18] Z. Hu, W. Liu, X. Qu, X. Yue, C. Chen, Z. Wang, Y. Cheng, Divide and conquer: Grounding LLMs as efficient decision-making agents via offline hierarchical reinforcement learning, ICML (2025). URL: https://arxiv.org/abs/2505.19761.

[19] N. Poerner, U. Waltinger, H. Schütze, BERT is not a knowledge base (yet): On the barriers to probing facts, in: ACL, 2020. URL: https://aclanthology.org/2020.acl-main.328/.

[20] D. Alivanistos, S. Báez Santamaría, M. Cochez, J.-C. Kalo, E. van Krieken, T. Thanapalasingam,

Prompting as probing: Using language models for knowledge base construction, LM-KBC challenge @ ISWC (2023). URL: https://arxiv.org/abs/2208.11057.
[21] OpenAI, Openai tokenizer, 2023. URL: https://platform.openai.com/tokenizer, accessed: 2025-09-11.

A. Model Reasoning Trace Examples

To provide concrete evidence of the inferential reasoning patterns identified in our error analysis, we present actual model reasoning traces captured during our experiments. These traces demonstrate how the model constructs plausible but incorrect answers when facing knowledge gaps.

A.1. Example: Hopen Island Area Query

Query: What is the exact area of Hopen in square kilometers?

Model's Internal Reasoning Trace:

> *"I need to gather accurate data on its area. I remember that Hopen is one of the larger islands in Svalbard... From what I can find, Hopen's area is approximately 1,600 square kilometers... Some sources might list it as 1,610 km²... The user wants the exact number, so I need to ensure it's the most up-to-date and accurate value... the consensus is 1,600 square kilometers."*

Model Output: 1600

Gold Standard: 47

Analysis: This trace exemplifies the inferential reasoning pattern where the model: (1) acknowledges uncertainty ("I need to gather accurate data"), (2) constructs seemingly plausible reasoning ("one of the larger islands"), (3) simulates source consultation ("From what I can find"), and (4) expresses false confidence ("the consensus is 1,600"). The 34x overestimation demonstrates how sophisticated reasoning can lead to confident but dramatically incorrect answers when the model lacks precise factual knowledge.

A.2. Example: Max Planck Medal 1950s Query (Divide-and-Conquer)

Query: List all recipients of the Max Planck Medal in the 1950s. Names only, no years, no explanations. Format: Name1, Name2, Name3

Model's Internal Reasoning Trace:

> *"I need to list all recipients of the Max Planck Medal in the 1950s... Wait, I'm getting confused... I think the Max Planck Medal was awarded in the 1950s to: Paul Dirac (but he received it in 1952)... John von Neumann (1956)... Enrico Fermi (1950)... Edward Teller (1958)... Wait, but I'm not entirely sure... I'm not 100% certain. However, based on my knowledge, these are the names."*

Model Output: Paul Dirac, Enrico Fermi, John von Neumann, Edward Teller

Gold Standard: Peter Debye, James Franck, Gustav Hertz, Paul Dirac, Walther Bothe, Enrico Fermi, Hans Bethe, Victor Weisskopf, Carl Friedrich von Weizsäcker, Wolfgang Pauli, Oskar Klein

Analysis: This example illustrates specific error patterns in temporal sub-queries:

Error Pattern Identification

- **Partial correct recall**: The model correctly identified 2 actual recipients (Paul Dirac, Enrico Fermi)
- **Name confabulation**: Generated John von Neumann and Edward Teller, who never received this award
- **Incomplete coverage**: Retrieved only 2 of 10 actual recipients from the decade
- **Uncertainty handling**: Despite expressing significant uncertainty throughout reasoning, the model still produced four names rather than abstaining

Observed Behavior The model appears to generate names based on "prominent physicists of the era" when facing knowledge gaps, mixing correct recipients with plausible but incorrect candidates. The internal reasoning shows the model attempting to reconstruct information through associative reasoning ("1956... John von Neumann") despite acknowledged uncertainty. This suggests potential improvements through stricter confidence thresholds or additional validation steps within the decomposition pipeline.

B. Detailed Error Case Analysis

This appendix provides the complete manual error analysis conducted on validation dataset samples. For each incorrect case, we present the model's direct output, the gold standard answer, and our systematic error classification following the four-category framework described in Section 5.

Table 13
Manual Error Analysis on Validation Set Samples (Incorrect Cases Only)

Relation	Entity	Model Output	Gold Standard	Error Classification
companyTrades AtStockExchange	RPS Group	None	London Stock Exchange	Context Issue
	Mercedes-Benz Group	Frankfurt Stock Exchange	NYSE, Italian SE, Frankfurt SE	Context Issue
	Edison International	NASDAQ	New York Stock Exchange	Context Issue
	Bharti Airtel	National SE, Bombay SE	Bombay Stock Exchange	Context Issue
	DRDGOLD Limited	Johannesburg SE	New York Stock Exchange	Context Issue
countryLand BordersCountry	Iraq	Turkey, Iran, Syria, Saudi Arabia, Kuwait	Iran, Jordan, Kuwait, Saudi Arabia, Syria, Turkey	Context Issue
	Turkey	Bulgaria, Greece, Georgia, Armenia, Azerbaijan, Syria, Iraq	Armenia, Azerbaijan, Bulgaria, Georgia, Greece, Iran, Iraq, Syria	Context Issue
	Ethiopia	Eritrea, Somalia, Kenya, Sudan, South Sudan	Djibouti, Eritrea, Kenya, Somalia, South Sudan, Sudan	Context Issue
	Burkina Faso	Mali, Niger, Benin, Togo, Ivory Coast	Benin, Côte d'Ivoire, Ghana, Mali, Niger, Togo	Evaluation Limit
	Serbia	Hungary, Croatia, Bosnia and Herzegovina, Montenegro, Kosovo, North Macedonia	Bosnia and Herzegovina, Bulgaria, Croatia, Hungary, Kosovo, Montenegro, North Macedonia, Romania	Context Issue
hasArea	Annobón Island	148	17	Context Issue
	La Digue	11.5	9.81	Context Issue
	Saint Kitts and Nevis	354	269.358763	Context Issue
	Flinders Island	1170	1367	Context Issue
	Goli otok	1.5	4.54	Context Issue
hasCapacity	Jinshan Sports Centre	20000	30000	Context Issue
	Estadio El Birichiche	15000	5000	Context Issue
	Stevenson Field	3000	3500	Context Issue
	Carrara Indoor Stadium	5000	2992	Context Issue
	Estádio da Gávea	70000	4000	Context Issue
personHasCity OfDeath	Christoph Eschenbach	Berlin	None	Context Issue
	Erich Schleyer	None	Vienna	Context Issue
	Bolesław Zoń	Khotyn	Warsaw	Context Issue
	Al Jarreau	San Diego	Los Angeles	Context Issue
	Souleymane Cissé	Paris	None	Context Issue

Table 14
Detailed Temporal Slicing Results for Max Planck Medal

Decade	Model Output	Physicists	Award Recipients	Gold Standard
1950s	Paul Dirac, Enrico Fermi, John von Neumann, Edward Teller	All 4 are physicists	Paul Dirac (1952), Enrico Fermi (1954)	Peter Debye, James Franck, Gustav Hertz, Paul Dirac, Walther Bothe, Enrico Fermi, Hans Bethe, Victor Weisskopf, Carl Friedrich von Weizsäcker, Wolfgang Pauli, Oskar Klein
1960s	Richard Feynman, Julian Schwinger, Hans Bethe, Edward Teller, John Bardeen, Lev Landau, Wolfgang Pauli, Eugene Wigner, Niels Bohr, Max Born	All are physicists	Niels Bohr (1930), Max Born (1948), Hans Bethe (1955), Wolfgang Pauli (1958), Lev Landau (1960), Eugene Wigner (1961)	Lev Landau, Eugene Wigner, Ralph Kronig, Rudolf Peierls, Samuel Goudsmit, George Uhlenbeck, Gerhart Lüders, Harry Lehmann, Walter Heitler, Freeman Dyson
1970s	Richard P. Feynman, Julian Schwinger, Hans Bethe, Edward Teller, John Bardeen, Robert Marshak	All are physicists	Hans Bethe (1955)	Rudolf Haag, Herbert Fröhlich, Nikolay Bogolyubov, Léon Van Hove, Gregor Wentzel, Ernst Stueckelberg, Walter Thirring, Paul Peter Ewald, Markus Fierz
1980s	John Bardeen, Steven Weinberg, Edward Teller, Murray Gell-Mann, Sheldon Glashow, Hans Bethe	All are physicists	Hans Bethe (1955)	Kurt Symanzik, Hans-Arwed Weidenmüller, Nicholas Kemmer, Res Jost, Yoichiro Nambu, Franz Wegner, Julius Wess, Valentine Bargmann, Bruno Zumino
1990s	John Bardeen, Steven Weinberg, Murray Gell-Mann, Klaus von Klitzing, Sheldon Glashow, Abdus Salam, Richard Feynman, Edward Witten, Gerardus 't Hooft	All are physicists	None are Max Planck Medal recipients	Hermann Haken, Wolfhart Zimmermann, Elliott H. Lieb, Kurt Binder, Hans-Jürgen Borchers, Siegfried Grossmann, Ludwig Faddeev, Gerald E. Brown, Raymond Stora, Pierre Hohenberg
2000s	*Format failure: exceeded token limit*	-	-	Martin Lüscher, Jürg Fröhlich, Jürgen Ehlers, Martin Gutzwiller, Klaus Hepp, Peter Zoller, Wolfgang Götze, Joel Lebowitz, Detlev Buchholz, Robert Graham
2010s	*Format failure: exceeded token limit*	-	-	Dieter Vollhardt, Giorgio Parisi, Martin Zirnbauer, Werner Nahm, David Ruelle, Viatcheslav Mukhanov, Herbert Wagner, Herbert Spohn, Juan Ignacio Cirac, Detlef Lohse
2020s	Klaus Hasselmann, David J. Thouless, Martinus Veltman, Gerardus 't Hooft	All are physicists	None are Max Planck Medal recipients	Andrzej Buras, Alexander Markovich Polyakov, Annette Zippelius, Rashid A. Sunyaev, Erwin Frey, Reinhard F. Werner