LLMs for Knowledge Graph Entity Matching: Information from the Graph Structure

Anonymous ACL submission

Abstract

Knowledge Graphs (KGs) encode entities and their relations in a semi-structured form, where nodes are often associated with rich textual descriptions and edges represent semantic relations. In this paper, we study the task of Knowledge Graph Entity Matching, which identifies equivalent entities across two KGs from the same narrow domain. Unlike traditional Entity Alignment, only a small subset of nodes are expected to match, making the task both sparse and semantically nuanced. We formulate KG entity matching as a Retriever-Augmented Generation (RAG) problem and propose four textual entity representations consisting node-level entity descriptions along with local graph structural information. These representations guide candidate retrieval from a huge pool of entities, followed by a large language model (LLM) that selects or rejects matches. Experiments on the OAEI KG Matching Benchmark show that LLMs can effectively exploit semi-structured information and outperform embedding-based baselines, though their advantage is smaller than anticipated. Our analysis reveals that (1) textual representation design significantly impacts retrieval quality, (2) LLMs exhibit non-trivial matching errors despite strong overall performance, and (3) lightweight retrieval-based methods achieve competitive results with substantially lower cost.

1 Introduction

Knowledge Graphs (KGs) are semi-structured knowledge bases in which nodes are typically associated with long textual descriptions, while the textual labels of edges semantically characterize the relationships among nodes. Processing KGs

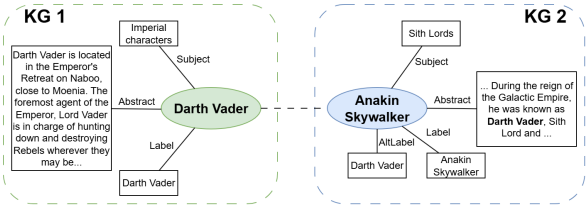

Figure 1: KG Entity Matching example in the Star Wars domain.

therefore requires specialized methodologies: standard graph mining techniques, such as Graph Neural Networks (Wu et al., 2021; Corso et al., 2024), generally ignore textual information, whereas natural language processing (NLP) methods, including Retriever-Augmented Generation (RAG) (Gupta et al., 2024; Edge et al., 2024), are designed for purely textual data.

We study the task of *Knowledge Graph Entity Matching*, where the goal is to identify nodes across two domain-specific KGs that refer to the same real-world entity. These matches enable KG fusion, which is particularly valuable in specialized industrial settings. We note that KG entity matching differs from the well-studied entity alignment problem (Zeng et al., 2021). In KG entity matching, the textual information of both KGs is in the same language, and only a limited number of nodes are expected to match (Li et al., 2021), in contrast to entity alignment settings, where almost all entities are expected to be mapped.

KG Entity matching is challenging: while simple string matching offers a strong baseline, identical names can denote different entities, and equivalent entities often appear under different names. Motivated by advances in Large Language Models (LLMs), we formulate KG entity matching as a RAG-like problem. Our main contribution is the

design of node-level textual representations that combine textual entity descriptions with information from the local graph context. These representations are used to retrieve candidate nodes from the other KG, after which an LLM selects the best match, if any.

Our experiments on the *OAEI KG Matching Benchmark* (Hertling and Paulheim, 2020) show that LLMs effectively leverage semi-structural information, though challenges still remain for them. Compared to traditional embedding-based methods, LLMs achieve higher accuracy, yet the performance gap is smaller than expected. While the OAEI benchmark encompasses many established approaches to KGEM, several of the most recent and relevant LLM-based systems lack publicly available evaluations, preventing a direct empirical comparison with our proposed methods.

In summary, our contributions are threefold:

1. We propose multiple textual representation (context engineering) strategies for encoding semi-structured KG information.
2. We conduct extensive experiments on KG entity matching, showing that LLMs achieve strong accuracy but also make unexpected errors.
3. We demonstrate that lightweight, resource-efficient approaches perform only slightly worse than LLMs.

2 Related work

Knowledge Graphs (KGs) have emerged as a powerful paradigm for representing structured knowledge, enabling semantic search, reasoning, and interoperability (Peng et al., 2023). However, the fusion or linkage of heterogeneous KGs remains a significant challenge due to the differences in scope, granularity, and structure among KGs (Kiourtis et al., 2019; Silva et al., 2022; Mohammadi et al., 2020).

The *Entity Alignment* problem was traditionally designed for Wikipedia/DBpedia thus it is text-focused and it assumes knowledge graphs of comparable size and structure, often derived from the exact same multilingual domains (Zeng et al., 2021). The standard and well-researched dataset for Entity Alignment is the *DBP15K* dataset (Tang et al., 2020) to align DBpedia entities across languages (such as Chinese and English). By contrast, *Knowledge Graph Entity Matching* (KGEM) addresses alignment across heterogeneous, domain-specific graphs where only a small subset of entities share counterparts and the graph structure plays a more important role (Li et al., 2021). This setting introduces additional challenges such as structural imbalance, sparse overlap, and differing terminology, making the task considerably less studied and far from solved in the literature.

Early works on Entity Alignment mostly focused on fine-tuning and optimizing embedding-based approaches (He et al., 2022; Zhu et al., 2024; Zeng et al., 2023). Other solutions include applying methods such as transformers (Li et al., 2023) or Monte Carlo Tree Search (Volkova et al., 2024). However, in recent years, LLMs have also emerged as a prominent direction in this field (Peeters and Bizer, 2023; Peeters et al., 2025), and have achieved better alignment results compared to traditional methods (He et al., 2023; Babaei Giglou et al., 2025; Amini et al., 2025). Various aspects, such as zero-shot prompting (Zhang et al., 2025), ensemble decision strategies (Wang et al., 2025), leveraging background knowledge (Jiang et al., 2024; Portisch et al., 2024) or fine-tuning existing models (Steiner et al., 2025) has been investigated.

The state-of-the-art published solutions for the KGEM task have evaluated Sentence-BERT-style embedding models (such as SORBET (Gosselin and Zouaq, 2023)), and also LLM prompting. For instance, OLaLa (Hertling and Paulheim, 2023) experimented with comparing zero-shot, few-shot, and multiple-choice formulations of KGEM. However, the utilization opportunities of LLMs in KGEM tasks is still under-studied, which is the main goal of this paper.

Recent work on *Context Engineering* shows that LLM performance depends strongly on how structured information, like Knowledge Graph (KG), is encoded and organized at inference time. Methods such as StructGPT and KARPA (Jiang et al., 2023; Fang et al., 2025) iteratively construct graph-grounded contexts from KG neighborhoods, while reasoning frameworks (Luo et al., 2024; Sun et al., 2024) alternate between graph traversal and prompting to integrate structured KG evidence. Other studies focus on verbalizing triples into fluent text (Agarwal et al., 2023, 2021) or formatting KG-derived facts into hierarchical or programmatic forms (e.g., JSON, Python) (Liu et al., 2024). The structure and ordering of these KG-informed inputs (Ge et al., 2025) markedly affect reasoning accuracy, underscoring input design as a key determinant of LLM effectiveness over structured

Fandom site	#Instances	#Properties	#Classes
Star Wars Wiki	145,033	700	269
The Old Republic Wiki	4,180	368	101
Star Wars Galaxies Wiki	9,634	148	67
Memory Alpha	45,828	325	181
Star Trek Expanded Universe	13,426	202	283
Memory Beta	51,323	423	240

Table 1: The scraped fandom sites and their graph's sizes

knowledge.

The two most similar papers to our work also follow a RAG-like approach for the Entity Alignment problem. Yang et al. (2024) identify candidate alignments for a given entity by considering name and structure embedding similarities between entities across the KG pairs. Then, an LLM makes iterative decisions for the given candidate set until it converges to a final answer. In a more recent study, Cheng et al. (2025) use LLMs to summarize entity descriptions and select candidates based on the embeddings of these summaries. The final decision in their approach is similarly determined by an LLM. These papers report near-perfect F1 scores (Yang et al., 2024; Wang and Chen, 2024; Tang et al., 2021; Cheng et al., 2025) indicating that the Entity Alignment problem is largely solved in controlled domains.

However, the task of *Knowledge Graph Entity Matching* is inherently different. Building on the recent ideas of LLMs for the Entity Alignment and Context Engineering, we propose a novel approach for the KGEM task. The highly heterogeneous nature of the input KGs require a RAG-like method, where we first retrieve a few candidates from the other graph then we enrich the context of LLM-based selectors based on local graph structures. Moreover, as KGEM targets only a small number of matches, the selectors frequently has to return with *no match found*. We propose various types of textual entity descriptions, and comparatively evaluate embedding-based methods and LLM prompting candidate selectors.

3 Dataset

The *Ontology Alignment Evaluation Initiative (OAEI)* has provided community-driven benchmarks for decades. The *OAEI Knowledge Graph Track*[1] has focused specifically on entity matching across large-scale, highly heterogeneous KG pairs, making it an important reference point for evaluating progress in KGEM.

[1]Data: https://oaei.ontologymatching.org/2025/knowledgegraph

Graph	#Pairs
memoryalpha-memorybeta	9365
stexpanded-memoryalpha	1779
swg-starwars	1121
swtor-starwars	1429

Table 2: Number of gold pairs

The OAEI KG benchmark consists of English fandom wiki sites (see Table 1), provided in *OWL* format, covering popular universes such as *Star Wars* and *Star Trek*. The KGs span a variety of domains (e.g., games, movies, comics) and vary in size from approximately 5,000 to 150,000 nodes. Each KG is derived from a site scrape that includes the infobox information, primarily textual attributes, the first few paragraphs of the main article text, every entity mention in the whole page and links to images.

Locating matching entities in these domains is not always trivial. As illustrated in Figure 1, relying solely on the primary labels of entities is insufficient — effective matching requires examining additional attributes and semantic relationships to uncover equivalences between differently named but conceptually identical entities.

4 Methodology

We propose a RAG-like approach for the KG Entity Matching problem. Our methodology consists of four major stages: (i) composition of textual representations (Dogtags) of entities i.e. KG nodes, (ii) embedding similarity-based and LLM prompted selection algorithms, (iii) deduplication to enforce 1:1 entity matches and (iv) merging the output of multiple methods. The framework diagram, shown in Figure 3, illustrates the overall process of these stages.

4.1 Textual Entity Representations (Dogtags)

Entities in the fandom KGs contain rich textual information along with relational information in the graph. To leverage both of them, we construct multiple forms of textual entity representations, which we refer to as Dogtags:

- **Node - Short:** Includes only the entity's label, alternative labels, type, and first two sentences of the abstract.
- **Node:** Same as *Node - Short*, but the full abstract is kept.
- **Node + Relations:** Contains all available textual and relational information (neighbours of

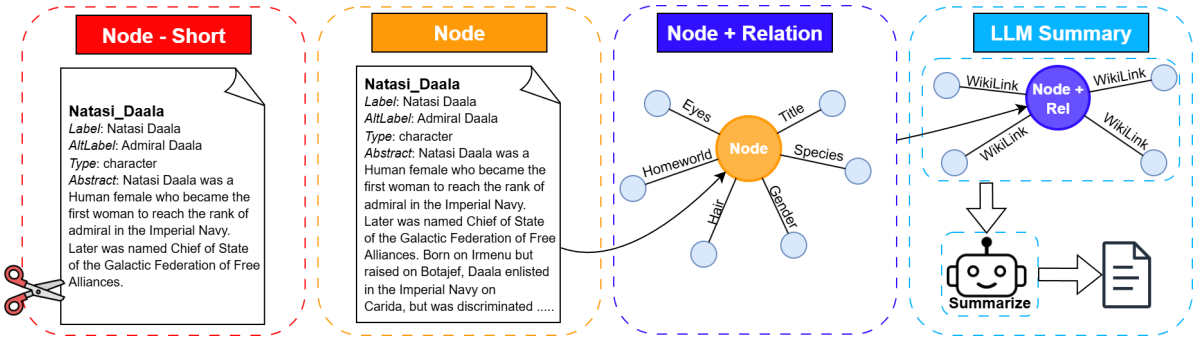

Figure 2: Information used from the KGs to create different representations

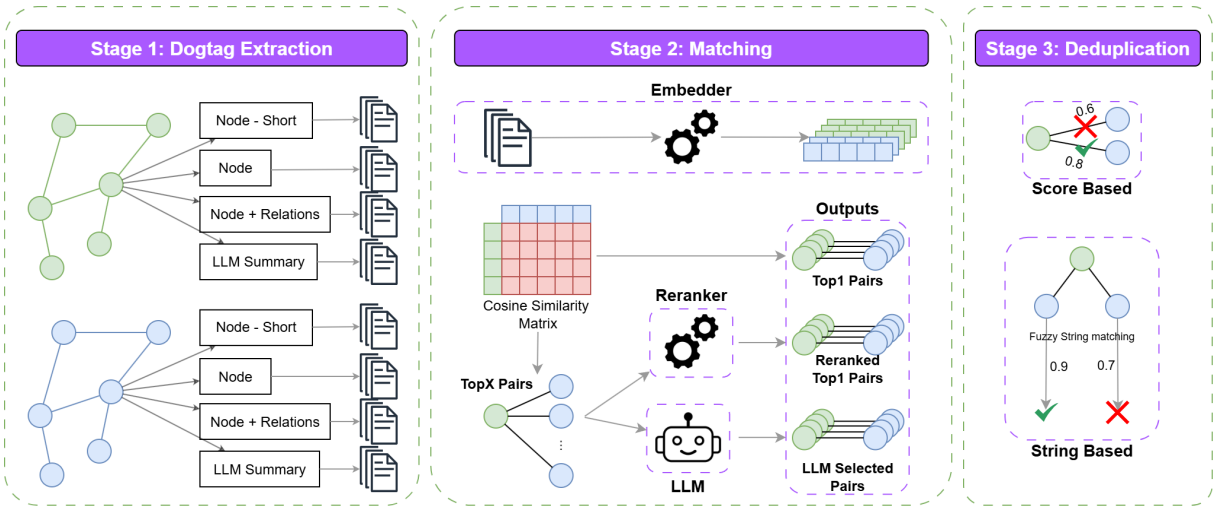

Figure 3: Proposed KG entity matching framework

the node in the graph) for each node, excluding links to other fandom pages, images or irrelevant metadata.

- **LLM Summary:** A concise paragraph generated with an LLM prompted for summarization from all available textual and relational information including links to other fandom pages, images and all metadata. To maintain tractability for LLM input, we limit the number of neighbours used to 20 by sampling.

We have introduced summarization due to the limited prompt capacity of LLM-based selectors. Nodes in the examined KGs often contain rich textual and relational information. Since the matcher must process both the anchor entity and all candidate entities simultaneously, input length becomes a critical bottleneck. Summaries therefore serve as a practical strategy to condense relevant information while retaining salient content.

We visualized the Dogtag extraction process in Figure 2. Each entity in every graph is associated with the four Dogtags, which serve as inputs to the matching algorithms described below. Appendix Table 8 illustrates the different Dogtag variants on an example, showing how textual and relational information are progressively included or summarized.

4.2 Matching Algorithms

All Dogtags are embedded into dense vector representations using document embedder models (Reimers and Gurevych, 2019). We iterate through all nodes (anchor entities) of the smaller KG and retrieve the k most similar entities for it from the other KG using these dense vectors. We propose five strategies to select a matching pair from the Top-k candidates. As only a small ratio of entities have matching pairs, the strategies frequently respond without any matchable entity from the other KG.

1. **Exact Matches:** If the label or an alternative label matches exactly across the two graphs, the corresponding pair is aligned. This method is independent of Dogtags.
2. **Top-1:** The top retrieved candidate is the response.
3. **Reranking:** A retriever reranker model is applied to reorder the retrieved candidates and select the highest reranked candidate.
4. **LLM Selector:** We present the anchor entity

alongside its Top-k retrieved candidates to an LLM, prompting it to select the best match or *None* if no appropriate match exists. We test two prompting styles:

- **Normal:** The task is explained in a straightforward manner, asking the model to choose the most appropriate candidate.
- **Strict:** The instructions emphasize that a candidate should only be selected when the model is completely certain that the two descriptions refer to the exact same entity.

Methods are applied separately for each Dogtag representation, allowing us to analyze the impact of different textual representations on matching performance.

4.3 Deduplication

Iterating over all entities of the smaller KG may yield 1:N or N:M pairs, while only 1:1 matches can be correct. We call these duplicates. To resolve duplicates, we apply two complementary strategies:

Score-based Deduplication (applied to Top-1 and Reranking): When similarity or reranker scores are available, we retain only the pair with the highest score for each entity.

String-based Deduplication (applied to Exact Matches and LLM Selector): When explicit scores are not available, we rely on fuzzy string matching between node names to select the most appropriate pair.

4.4 Merging Alignments

Exact matches is a strong baseline. These pairs are highly reliable but also limited: many true matches differ in surface form, while homonymous labels may cause errors. To improve recall, we merge exact matches with matches from other methods.

Our procedure takes two matching sets: a *primary* set (exact matches) and a *secondary* set. First, we compute a similarity threshold by taking the median similarity score of the exact-match pairs. This acts as a filter for the secondary set.

For each secondary pair, we include it in the merged alignment only if (i) neither entity is already matched in the primary set, and (ii) its similarity score exceeds the threshold. Otherwise, the pair is discarded to preserve 1:1 consistency and avoid noisy additions.

This strategy keeps the strong signal of exact matches intact while selectively incorporating complementary alignments. It thus allows us to analyze how other methods reduce residual errors and improve recall beyond *Exact matches*.

5 Experiments

Models For embeddings we used *BAAI/bge-large-en-v1.5*, for reranking we used *BAAI/bge-reranker-large* without any fine-tuning, and as the LLM we used *Llama-3.1-70B-Instruct*, all obtained from HuggingFace. Sampling is disabled in all LLM runs to ensure deterministic outputs.

Evaluation We follow the evaluation protocols recommended by the OAEI shared task organisers. Manually annotated gold matches are available for four KG pairs (See Table 2). However, due to the limited annotation resources, these alignments are partial: not all correct correspondences are included in the gold standard. When assessing a predicted correspondence, if neither node in the pair appears in the gold standard, the status of the pair remains indeterminate (i.e., it may be either a true match or a false positive). Such pairs are excluded from the evaluation. Precision, recall, and F_1 scores are then computed on the remaining pairs to assess system performance. This evaluation aligns with the dataset creators' method.

Implementation Details We used $k = 10$ for the retriever and leave the investigation of other values as future work. All methods run purely in inference mode; no fine-tuning or additional training is performed. The experiments were executed on 4 × H100 GPUs. Since all models and inference steps are deterministic, results are stable across repeated runs (only single run needed). For fuzzy string matching, we used the `rapidfuzz` python lib.

Reproducibility Our codebase, prompts and data are publicly available on GitHub.[2] We provide a `commands.txt` file that documents the exact commands used for running every experiment end-to-end.

Results Table 3 shows the performance of each Dogtag-method combination for our benchmark graph pairs on the KG entity matching tasks. Table 4 similarly presents performance, but after the matches were merged with the exact matches. For the `Top-1` and `Reranking` baselines, we apply the median thresholding strategy described in Section 4.4. In contrast, when merging with the

[2] https://anonymous.4open.science/r/KGEntityMatching-Public-0210/

		memoryalpha-memorybeta			stexpanded-memoryalpha			swg-starwars			swtor-starwars		
Dogtag	**Method**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Node - Short	Top-1	79.32	73.69	76.40	84.34	84.15	84.24	79.53	75.20	77.30	86.83	85.79	86.31
Node - Short	Top-10 Reranked	76.76	71.75	74.17	72.37	72.74	72.55	75.49	71.72	73.56	84.60	8188 83.21	
Node - Short	LLM Selection Normal	95.66	81.19	87.83	93.06	87.46	90.18	90.97	76.36	83.03	94.14	87.75	90.84
Node - Short	LLM Selection Strict	**96.39**	78.45	86.50	92.96	86.06	89.38	91.16	70.83	79.72	93.87	85.65	89.57
Node	Top-1	72.48	64.92	68.49	69.55	70.10	69.82	80.34	75.83	78.02	82.81	79.92	81.34
Node	Top-10 Reranked	75.05	68.25	71.49	69.98	70.49	70.23	75.21	72.26	73.70	84.59	81.81	83.17
Node	LLM Selection Normal	93.26	76.30	83.93	90.81	83.36	86.93	91.70	77.88	84.23	94.05	86.35	90.04
Node	LLM Selection Strict	94.26	74.30	83.10	91.68	83.02	87.14	92.13	73.15	81.55	94.18	84.88	89.29
Node + Relation	Top-1	80.71	75.55	78.05	78.00	77.74	77.87	79.76	76.98	78.35	88.83	86.28	87.54
Node + Relation	Top-10 Reranked	7580	7081	73.22	75.53	74.59	75.06	78.56	74.22	76.33	86.11	83.69	84.88
Node + Relation	LLM Selection Normal	95.59	85.90	90.49	96.69	90.28	93.37	93.31	79.57	85.89	94.43	**90.13**	**92.23**
Node + Relation	LLM Selection Strict	96.15	85.02	90.24	**97.20**	89.60	93.24	93.77	76.54	84.28	**95.04**	88.45	91.63
LLM Summary	Top-1	91.83	**89.47**	90.64	91.84	91.68	91.76	87.91	**87.60**	**87.76**	89.15	88.52	88.83
LLM Summary	Top-10 Reranked	80.49	76.48	78.44	80.37	80.10	80.24	74.59	68.60	71.47	81.36	78.52	79.91
LLM Summary	LLM Selection Normal	95.46	86.90	**90.98**	95.22	**91.74**	**93.44**	93.44	80.11	86.26	90.76	86.56	88.61
LLM Summary	LLM Selection Strict	95.83	82.79	88.84	95.16	89.49	92.24	**94.08**	73.68	82.64	90.32	84.25	87.18

Table 3: Performance comparison across KG pairs, Dogtags and methods

		memoryalpha-memorybeta			stexpanded-memoryalpha			swg-starwars			swtor-starwars		
Dogtag	**Method**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
	ExactMatch	**94.49**	88.23	91.25	**96.59**	90.89	93.66	**93.67**	62.00	74.61	**94.20**	88.66	91.35
Node - Short	Top-1	94.07	89.83	91.90	96.37	92.64	94.47	92.82	71.54	80.81	94.12	90.69	92.37
Node - Short	Top-10 Reranked	92.49	89.35	90.90	90.01	92.19	91.09	86.29	70.74	77.75	92.65	89.99	91.30
Node - Short	LLM Selection Normal	93.51	91.97	92.73	91.97	94.04	93.00	90.00	81.09	85.31	92.67	92.02	92.35
Node - Short	LLM Selection Strict	93.77	91.22	92.48	91.69	92.97	92.32	90.18	76.98	83.06	92.58	91.67	92.12
Node	Top-1	93.81	90.10	91.92	95.66	92.97	94.30	93.00	72.26	81.33	94.17	90.41	92.25
Node	Top-10 Reranked	92.26	89.22	90.72	90.27	92.36	91.30	86.59	69.67	77.21	92.66	90.06	91.34
Node	LLM Selection Normal	92.47	92.24	92.35	91.31	94.49	92.87	90.58	82.34	86.26	92.75	92.16	92.45
Node	LLM Selection Strict	92.88	91.70	92.29	91.59	94.32	92.94	90.98	79.21	84.69	92.85	91.74	92.29
Node + Relation	Top-1	93.99	90.45	92.18	96.11	93.09	94.57	92.82	71.54	80.81	94.03	90.34	92.15
Node + Relation	Top-10 Reranked	91.88	89.64	90.74	93.81	93.65	93.73	90.89	72.08	80.40	93.89	90.34	92.08
Node + Relation	LLM Selection Normal	93.16	93.50	**93.33**	95.36	**95.90**	**95.63**	92.26	82.96	87.36	93.95	**93.42**	**93.68**
Node + Relation	LLM Selection Strict	93.35	93.22	93.28	95.51	95.67	95.59	92.48	81.18	86.46	94.19	93.00	93.59
LLM Summary	Top-1	94.09	90.73	92.38	96.24	93.54	94.87	93.16	71.63	80.99	94.04	90.55	92.26
LLM Summary	Top-10 Reranked	92.57	90.54	91.54	94.30	92.92	93.60	90.71	70.56	79.38	93.49	90.48	91.96
LLM Summary	LLM Selection Normal	92.93	**93.64**	93.28	94.83	94.94	94.89	92.06	**85.82**	**88.83**	93.67	93.21	93.44
LLM Summary	LLM Selection Strict	93.15	93.22	93.18	94.73	95.05	94.89	92.54	83.05	87.54	93.53	93.00	93.26

Table 4: Merged results comparison across KG pairs, Dogtags and methods

Step	**Time / 5000 nodes**
LLM Summary	$\sim$ 2 hours
Embedding	$\sim$ 1 minute
Reranking	$\sim$ 6 minutes
LLM Selection	$\sim$ 1 hour

Table 5: Runtime per processing step (for 5000 nodes).

`LLM Selector`, no thresholding is applied since the model does not produce scores; however, the LLM may abstain from selecting a pair.

6 Discussion

Table 3 provides a comparison of the four Dogtag variants. The performance ranking is consistent across datasets: **Node** $<$ **Node-Short** $<$ **Node+Relations** $<$ **LLM Summary**. The gains from truncating the abstract highlight that the most informative content is typically concentrated in the opening sentences, while later text may introduce peripheral details that dilute the entity signal. Long abstracts can therefore overshadow the label itself, which often provides the strongest matching cue. In contrast, the LLM-generated summaries achieve the best overall performance, suggesting that compressing information into concise text improves both recall and precision.

Turning to the selectors, we observe a consistent ranking from lowest to highest performance: **Top-10 Reranked** $<$ **Top-1** $<$ **LLM Strict** $<$ **LLM Normal**. There is a high degree of lexical overlap over all graph pairs, making Exact matching particularly effective on these datasets, on MemoryAlpha–MemoryBeta it even dominates other methods. The strict LLM prompt delivers high precisions but at the cost of reduced recall, leading to a lower F1 score than the more permissive prompt. Consequently, the Normal prompt strikes the best balance between reliability and coverage. By contrast, the reranker consistently underperforms. Expanding the candidate pool from 10 to 100 candidates further degrades performance, suggesting that a larger candidate set introduces noise that outweighs any potential gains.

When combined with Exact Matches, the rela-

tive ordering of methods remains unchanged (Table 4). Our goal of improving recall was successful. The strongest configuration is therefore the hybrid approach of **Exact Matches + LLM Summary Dogtag + LLM Selection (Normal)**, which consistently yields the highest F1 scores.

Lightweight embedding-based models remain surprisingly competitive with large-scale LLMs. This is especially important when considering that for the smaller KG, one would need to make an LLM call for each node (tens of thousands in total) which is several orders of magnitude worse in both runtime and cost compared to simply retrieving the Top-1 result.

How consistent is the LLM selector? To assess the stability of the `LLM Selector`, we investigated how sensitive its choices are to the ordering of the same set of candidate entities. Specifically, we varied the order of the Top-10 retrieved candidates (based on cosine similarity) presented to the LLM and compared the resulting matches. We considered three settings:

- **Original:** candidates sorted in decreasing cosine similarity,
- **Reverse:** candidates sorted in the opposite order,
- **Random:** candidates presented in a randomly permuted order.

Figure 4 shows the distribution of the positions chosen by the LLM. In the original ordering, most selections concentrate on the first candidate, as expected. Reversing the order flips this distribution, with the majority now drawn from the end of the original ranking. Interestingly, the random ordering reveals a non-uniform bias: the LLM strongly prefers earlier positions, with selection probability decreasing as candidates appear later in the list. This effect persisted across three independent random shuffles, confirming its robustness. Notably, even in the reversed ordering, we observe a small but systematic preference for the first position, suggesting a positional bias beyond semantic content.

Regarding performance (Table 6), altering the retrieved candidates' order either by reversal or randomization causes the F1 score to drop sharply, from 90.18 to the 73–76 range. Under random orderings, we also observe substantial variance in recall. These results highlight that the effectiveness of the LLM selection mechanism is highly sensitive to the ordering of candidates, rather than being determined solely by their semantic content. Since this experiment used the *Node-Short* Dogtags ($\approx$ 2,000 characters on average), this sensitivity cannot be attributed to input truncation or mid-context attention loss.

	stexpanded-memoryalpha		
Order	**P**	**R**	**F1**
Retriever	93.06	87.46	90.18
Reverse	65.09	89.83	75.48
Random	64.18 $\pm$ 0.24	86.9 $\pm$ 5.31	73.77 $\pm$ 1.96

Table 6: LLM Selection different ordering performance.

Why does the LLM summary perform better? To understand the advantage of LLM summaries, we compared the recall of gold-standard alignments within the Top-10 retrieved candidates across all four Dogtag configurations (Hits@10). As shown in Figure 5, the LLM summaries consistently contain a higher proportion of gold pairs than other representations, which explains why downstream matchers achieve stronger performance when operating on this textual form.

Two main factors likely contribute to this effect:
Inclusion of Wiki Links: Summaries integrate mentions of related entities via wiki links, providing additional context. While these relations extend coverage, our manual inspection suggests their contribution is limited as the core identifying information is already present in standard attributes and relations. Examples in Appendix A illustrate that information introduced solely by wiki links is rarely decisive.
Natural language representation: The summaries transform structured graph content into coherent, dense paragraphs. This yields two benefits. First, redundancy is reduced: multiple relations expressing the same fact are collapsed into a single mention, making the representation more compact. Second, continuous natural language text aligns better with the training distribution of sentence embedding models such as `bge-large-v1.5`, which are typically optimized for free-form text rather than structured triples. Both factors likely improve the effectiveness of the retrieval. Table 9 in the Appendix shows an example where a gold pair was found by the *LLM Summary* method, but the *Node + Relation* method couldn't find it.

Given the same retrieved candidates, do semi-structured or summarized representations work better? LLMs process text differently from embedding-based models: they are pre-trained on vast amounts of natural language, along with other

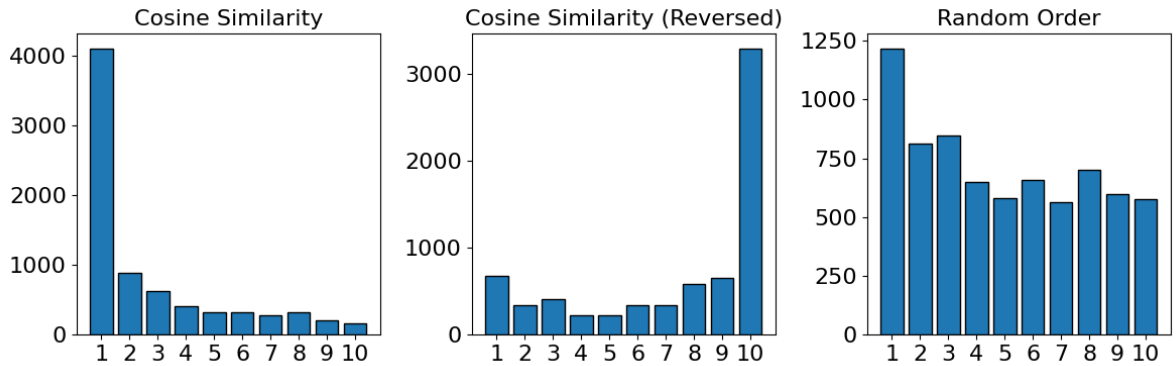

Figure 4: LLM pick index frequency based on different orderings of the Top-10 entities.

		memoryalpha-memorybeta			stexpanded-memoryalpha			swg-starwars			swtor-starwars		
Candidates	**Decision**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
LLM Summary	LLM Summary	**95.46**	86.90	90.98	**95.22**	91.74	**93.44**	**93.44**	80.11	86.26	**90.76**	86.56	88.61
LLM Summary	Node - Short	90.86	88.85	89.84	87.29	95.00	90.98	90.43	83.50	86.83	90.42	92.44	**91.42**
LLM Summary	Node	89.73	89.70	89.72	86.19	95.39	90.55	90.28	**83.68**	**86.85**	89.93	92.51	91.20
LLM Summary	Node + Relation	89.50	**93.55**	**91.48**	85.74	**95.67**	90.44	87.75	**83.68**	85.66	88.70	91.74	90.20

Table 7: Performance on LLM Summary Top-10, but different Dogtags for LLM Selection.

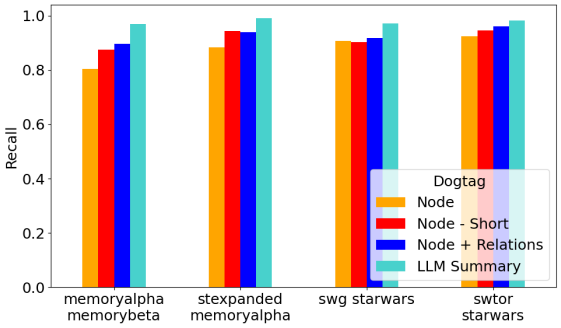

Figure 5: Top-10 pairs' Gold Recall for each Dogtag version.

modalities including structured datasets and software codes. We hypothesized that LLMs might in fact benefit more from structured inputs (even if redundant) than from dense summaries. To test this, we fixed the retrieval stage by using Top-10 candidates retrieved based on LLM summaries, and then applied the `LLM Selector` with different Dogtag representations (Node, Node - Short, Node + Relations) of the candidates. As shown in Table 7, different representations perform best for each KG pair. This finding suggests that, while the retriever favors compact textual summaries, LLMs can exploit richer and structured descriptions more effectively in some cases, corroborating prior observations by Liu et al. (2024); Wu and Tsioutsiouliklis (2024).

Did the LLM summaries contain hallucinations? To ensure faithfulness, we manually inspected 20 LLM-generated summaries from each KG. In all cases, we found that the summaries were faithful to the input: no information was introduced beyond what was provided in the prompt. This confirms that retriever performance improvements are attributable to improved representation and compression of available knowledge, rather than hallucination of external facts from LLM pretraining.

Conclusion

In our work, we addressed the task of *Knowledge Graph Entity Matching*, proposing multiple strategies for constructing textual representations from textual and graph structure information about entities. After a retriever step both embedding-based and LLM-based selection methods were evaluated. Experiments on the OAEI Knowledge Graph Track revealed four main insights: (1) textual representation design has a strong impact on retrieval and selection quality, (2) LLM-based approaches can outperform traditional methods but incur higher computational cost, and (3) LLM performance is highly sensitive to context engineering and candidate ordering.

Here, we involved only local graph information (neighbours) into the textual descriptions. In the future, we plan to experiment with global graph information, like communities, influencer paths and so on.

Limitations

We observed that some of the gold-standard pairs provided for evaluation were incorrect, which may have affected performance estimates.

Moreover, the `Top-1` and `Reranking` workflows

were not explicitly optimized for precision or recall, and all reported results correspond to $k = 10$. Exploring different k values and incorporating precision/recall-oriented optimization could further improve performance, which we leave for future work.

LLM-based solutions, although often more accurate, require substantially more computational resources, which limits their accessibility to many use cases.

The datasets employed in our experiments are drawn from well-known fandom domains, which are present in the pre-training dataset of any LLM. It remains uncertain how well these methods would transfer to less popular or more obscure domains.

Additionally, entities with sparse or noisy descriptions/relations are harder to align reliably. Embedding-based methods may conflate entities with highly similar contexts (e.g., multiple characters sharing titles or roles), leading to false positives.

Finally, our deduplication heuristics can discard valid alignments if similarity scores fail to reflect the true underlying connections.

Acknowledgments

Anonymous ACL submission

References

Ankush Agarwal, Sakharam Gawade, Amar Prakash Azad, and Pushpak Bhattacharyya. 2023. Kitlm: Domain-specific knowledge integration into language models for question answering. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 202–294, Goa University, Goa, India. NLP Association of India (NLPAI).

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Reihaneh Amini, Sanaz Saki Norouzi, Pascal Hitzler, and Reza Amini. 2025. Towards complex ontology alignment using large language models. In *Knowledge Graphs and Semantic Web*.

Hamed Babaei Giglou, Jennifer D'Souza, Felix Engel, and Sören Auer. 2025. Llms4om: Matching ontologies with large language models. In *The Semantic Web: ESWC 2024 Satellite Events*, pages 25–35, Cham. Springer Nature Switzerland.

Jingwei Cheng, Chenglong Lu, Linyan Yang, Guoqing Chen, and Fu Zhang. 2025. Easyea: Large language model is all you need in entity alignment between knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20981–20995, Vienna, Austria. Association for Computational Linguistics.

Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. 2024. Graph neural networks. *Nature Reviews Methods Primers*, 4(1):17.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization.

Siyuan Fang, Kaijing Ma, Tianyu Zheng, Xeron Du, Ningxuan Lu, Ge Zhang, and Qingkun Tang. 2025. Karpa: A training-free method of adapting knowledge graph as references for large language model's reasoning path aggregation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24724–24746, Vienna, Austria. Association for Computational Linguistics.

Yuyao Ge, Shenghua Liu, Baolong Bi, Yiwei Wang, Lingrui Mei, Wenjie Feng, Lizhe Chen, and Xueqi Cheng. 2025. Can graph descriptive order affect solving graph problems with LLMs? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6404–6420, Vienna, Austria. Association for Computational Linguistics.

Francis Gosselin and Amal Zouaq. 2023. Sorbetmatcher results for oaei 2023. In *OM@ ISWC*, pages 184–190.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions.

Yuan He, Jiaoyan Chen, Denvar Antonyrajah, and Ian Horrocks. 2022. Bertmap: A bert-based ontology alignment system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5684–5691.

Yuan He, Jiaoyan Chen, Hang Dong, and Ian Horrocks. 2023. Exploring large language models for ontology alignment. In *Proceedings of the ISWC 2023 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with the 22nd International Semantic Web Conference (ISWC 2023)*. CEUR Workshop Proceedings.

Sven Hertling and Heiko Paulheim. 2020. The knowledge graph track at oaei. In *The Semantic Web*, pages 343–359, Cham. Springer International Publishing.

Sven Hertling and Heiko Paulheim. 2023. Olala: Ontology matching with large language models. In *Proceedings of the 12th Knowledge Capture Conference 2023*, K-CAP '23, page 131–139, New York, NY, USA. Association for Computing Machinery.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, Singapore. Association for Computational Linguistics.

Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, and Yuanzhuo Wang. 2024. Unlocking the power of large language models for entity alignment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7566–7583, Bangkok, Thailand. Association for Computational Linguistics.

Athanasios Kiourtis, Argyro Mavrogiorgou, Andreas Menychtas, Ilias Maglogiannis, and Dimosthenis Kyriazis. 2019. Structurally mapping healthcare data to hl7 fhir through ontology alignment. *Journal of Medical Systems*, 43(3):62.

Yuliang Li, Jinfeng Li, Yoshi Suhara, AnHai Doan, and Wang-Chiew Tan. 2023. Effective entity matching with transformers. *The VLDB Journal*, 32(6):1215–1235.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, Jin Wang, Wataru Hirota, and Wang-Chiew Tan. 2021. Deep entity matching: Challenges and opportunities. *J. Data and Information Quality*, 13(1):1–17.

Kai Liu, Zhihang Fu, Chao Chen, Wei Zhang, Rongxin Jiang, Fan Zhou, Yaowu Chen, Yue Wu, and Jieping Ye. 2024. Enhancing llm's cognition via structurization. In *Advances in Neural Information Processing Systems*, volume 37, pages 133710–133742. Curran Associates, Inc.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Majid Mohammadi, Wout Hofman, and Yao-Hua Tan. 2020. Seamless interoperability in logistics by ontology alignment. *Journal of Supply Chain Management Science*, 1(3-4):104–117.

Ralph Peeters and Christian Bizer. 2023. Using chatgpt for entity matching. In *New Trends in Database and Information Systems*, pages 221–230, Cham. Springer Nature Switzerland.

Ralph Peeters, Aaron Steiner, and Christian Bizer. 2025. Entity matching using large language models. In *Proceedings 28th International Conference on Extending Database Technology (EDBT 2025), Barcelona, Spain, March 25–March 28*, volume 2, Experiments & Analyses Track, pages 529–541, Konstanz. OpenProceedings.org.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102.

Jan Portisch, Michael Hladik, and Heiko Paulheim. 2024. Background knowledge in ontology matching: A survey. *Semantic Web*, 15(6):2639–2693.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Marta Contreiras Silva, Daniel Faria, and Catia Pesquita. 2022. Matching multiple ontologies to build a knowledge graph for personalized medicine. In *The Semantic Web*, pages 461–477, Cham. Springer International Publishing.

Aaron Steiner, Ralph Peeters, and Christian Bizer. 2025. Fine-tuning large language models for entity matching. In *2025 IEEE 41st International Conference on Data Engineering Workshops (ICDEW)*, pages 9–17.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2020. Bert-int:a bert-based interaction model for knowledge graph alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3174–3180. International Joint Conferences on Artificial Intelligence Organization. Main track.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2021. Bert-int: a bert-based interaction model for knowledge graph alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20)*, pages 439–445, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.

Teon Volkova, Evander Delacruz, and Thaddeus Cavanaugh. 2024. A novel approach to optimize large language models for named entity matching with monte carlo tree search. *Wiley*.

Tianshu Wang, Xiaoyang Chen, Hongyu Lin, Xuanang Chen, Xianpei Han, Le Sun, Hao Wang, and Zhenyu Zeng. 2025. Match, compare, or select? an investigation of large language models for entity matching. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 96–109, Abu Dhabi, UAE. Association for Computational Linguistics.

Zhichun Wang and Xuan Chen. 2024. Dera: Dense entity retrieval for entity alignment in knowledge graphs. *arXiv preprint arXiv:2408.01154*.

Xue Wu and Kostas Tsioutsiouliklis. 2024. Thinking with knowledge graphs: Enhancing llm reasoning through structured data.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Linyao Yang, Hongyang Chen, Xiao Wang, Jing Yang, Fei-Yue Wang, and Han Liu. 2024. Two heads are better than one: Integrating knowledge from knowledge graphs and large language models for entity alignment.

Kaisheng Zeng, Chengjiang Li, Lei Hou, Juanzi Li, and Ling Feng. 2021. A comprehensive survey of entity alignment for knowledge graphs. *AI Open*, 2:1–13.

Weixin Zeng, Xiang Zhao, Zhen Tan, Jiuyang Tang, and Xueqi Cheng. 2023. Matching knowledge graphs in entity embedding spaces: An experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12770–12784.

Zeyu Zhang, Paul Groth, Iacer Calixto, and Sebastian Schelter. 2025. Anymatch – efficient zero-shot entity matching with a small language model. In *Workshop on Preparing Good Data for Generative AI: Challenges and Approaches*.

Beibei Zhu, Ruolin Wang, Junyi Wang, Fei Shao, and Kerun Wang. 2024. A survey: knowledge graph entity alignment research based on graph embedding. *Artificial Intelligence Review*, 57(9):229.

A Examples

We illustrate different Dogtag representations for the entity: `Black_Sun_AEG-77_Vigo_Gunship` from the `starwars` graph in Table 8.

Node
label: Black Sun AEG-77 Vigo Gunship **altLabel:** Black Sun AEG-77 Vigo Gunship **type:** class/starship **abstract:** The Black Sun's transports became a common sight in both the underworld and legitimate shipping with the advent of Xizor Transport Systems. This particular transport class became well known as lightly defended. Pirates from a variety of bands began picking off the freighters, particularly those smuggling illegal cargo for Black Sun. Instead of dedicating fighters to escort transports carrying valuable cargo, Black Sun began modifying some of their transports to carry heavy firepower and act as escorts for the rest of the convoys. Named after Black Sun's lieutenants, the Vigo, this once-merchant class vessel has been elevated to the status of "gunship." This gunship comes equipped with two forward mounted guns, four top mounted turrets, and two bottom mounted turrets. The gunship, originally a transport ship, is still large enough to hold a number of items and passengers inside. Black Sun AEG-77 Vigo Gunship Chassis Blueprints.
Node - Short
label: Black Sun AEG-77 Vigo Gunship **altLabel:** Black Sun AEG-77 Vigo Gunship **type:** class/starship **abstract:** The Black Sun's transports became a common sight in both the underworld and legitimate shipping with the advent of Xizor Transport Systems. This particular transport class became well known as lightly defended.
Node + Relation
abstract: The Black Sun's transports became a common sight in both the underworld and legitimate shipping with the advent of Xizor Transport Systems. This particular transport class became well known as lightly defended. Pirates from a variety of bands began picking off the freighters, particularly those smuggling illegal cargo for Black Sun. Instead of dedicating fighters to escort transports carrying valuable cargo, Black Sun began modifying some of their transports to carry heavy firepower and act as escorts for the rest of the convoys. Named after Black Sun's lieutenants, the Vigo, this once-merchant class vessel has been elevated to the status of "gunship." This gunship comes equipped with two forward mounted guns, four top mounted turrets, and two bottom mounted turrets. The gunship, originally a transport ship, is still large enough to hold a number of items and passengers inside. Black Sun AEG-77 Vigo Gunship Chassis Blueprints. **accel:** 10 **altLabel:** ['Black Sun AEG-77 Vigo Gunship', 'Blacksun AEG-77 Vigo Gunship'] **astromech:** Flight computer **certification:** Flight_Artistry_(Skill) **crew:** Pilot, co-pilot/operations officer, Engineer, gunner x6 **decel:** 10 **label:** Black Sun AEG-77 Vigo Gunship **mass:** ∼3 million **name:** AEG-77 **pitch:** 10 **role:** Multipassenger ship, Transporter, Battleship, Escort **roll:** 10 **speed:** 0.6 **subject:** Category:Freelance_ships **type:** starship **weapons:** 2 **yaw:** 10
LLM Summary
The Black Sun AEG-77 Vigo Gunship is a modified transport vessel that has been converted into a gunship to escort convoys carrying valuable cargo. Originally a lightly defended freighter, it has been equipped with heavy firepower, including two forward-mounted guns and six turrets. The gunship can hold multiple passengers and items, and is crewed by a pilot, co-pilot, engineer, and six gunners. With a mass of approximately 3 million units and a speed of 0.6, the AEG-77 is a formidable escort vessel in the Black Sun fleet.

Table 8: Dogtag variants of the entity: Black Sun AEG-77 Vigo Gunship from starwars

B Prompts Used for Top-10 Selection

All prompts formatted the candidate entities between "<EXAMPLE></EXAMPLE>" XML tags, while adding an "ID:{}" line to each candidate that the LLM can refer to.

```
<EXAMPLE>
ID:1
abstract: The Mandellian Gambit was a
    starship active during the Great
    Galactic War between the Galactic
    Republic and the Sith Empire.
altLabel: Mandellian Gambit
era: Old_Republic_era
label: Mandellian Gambit
subject: Category:Starships
type: individual_ship
</EXAMPLE>
```

B.1 Normal

```
TASK: You will be given a description of
     an anchor entity and a list of
    candidate entities, all formatted
    with <EXAMPLE> tags. Your task is to
    :
- Identify the candidate entity that is
    the same as the anchor entity.
- Return the ID number of the matching
    candidate entity.
- If none of the candidates match the
    anchor entity, return -1.
- Answer with the ID (or -1) only, no
    explanations
###
<ANCHOR>
{anchor}
</ANCHOR>
###
{candidates strings}
###
Answer:
```

B.2 Strict

```
TASK: You will be given a description of
     an anchor entity and a list of
    candidate entities, all formatted
    with <EXAMPLE> tags. Your task is to
    :
- Identify the candidate entity that is
    the same as the anchor entity.
- Return the ID number of the matching
    candidate entity. The selected
    Candidate MUST BE the same as the
    Anchor entity.
- If none of the candidates match the
    anchor entity, return -1.
- Answer with the ID (or -1) only, no
    explanations
###
<ANCHOR>
{anchor}
</ANCHOR>
###
{candidates strings}
###
Answer:
```

http://swg.wikia.com/A_Demon_Doll	http://starwars.wikia.com/Desert_Demon_Doll
LLM Summary: ✓ A Demon Doll is a reward item in the game, specifically obtained by completing the Tusken Valuables collection. It is categorized as a collection reward and has a corresponding wiki page with additional information and images, including a picture of the Tusken doll. The item is also linked to the Tusken Valuables collection, providing context for its significance in the game. Overall, A Demon Doll serves as a notable achievement for players who complete the specified collection.	**LLM Summary:** ✓ The Desert Demon Doll is a toy created by the Tusken Raiders on the planet Tatooine. It is a notable part of Tusken Raider culture and is also classified as a toy and merchandise in the Star Wars universe.
Node + Relation: ✗ A_Demon_Doll abstract: This is the reward for completing the Tusken Valuables collection. altLabel: A Demon Doll label: A Demon Doll subject: Category:Collection_rewards type: Thing	**Node + Relation:** ✗ Desert_Demon_Doll abstract: A Desert Demon Doll was a toy made by Tusken Raiders on Tatooine. altLabel: [Desert demon doll, Desert Demon Doll] label: Desert Demon Doll subject: [Category:Tusken_Raider_culture, Category:Toys_and_merchandise] type: swgwiki

Table 9: Comparison of gold pairs found by LLM summary but not by Node + Relation representations using LLM Selection.

	memoryalpha-memorybeta			stexpanded-memoryalpha			swg-starwars			swtor-starwars		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
BaselineAltLabel	0.88	0.89	0.89	0.88	0.92	0.90	0.92	0.63	0.74	0.92	0.90	0.91
BaselineLabel	0.95	0.77	0.85	0.98	0.83	0.90	0.96	0.53	0.68	0.95	0.84	0.89
LogMap	0.89	0.76	0.82	0.88	0.75	0.81	0.94	0.68	0.79	0.94	0.75	0.84
LogMapLt	0.90	0.77	0.83	0.94	0.82	0.88	0.91	0.62	0.74	0.91	0.83	0.87
Matcha	0.58	0.90	0.71	0.68	0.93	0.78	0.71	0.80	0.75	0.73	0.90	0.81
LLM Summary + LLM Selection	0.95	0.87	0.91	0.95	0.92	0.93	0.93	0.80	0.86	0.91	0.87	0.89

Table 10: Comparison with SOTA systems on OAEI Knowledge Graph Track from 2024

C Comparison with SOTA systems on OAEI Knowledge Graph Track

In Table 10 we compare our results to the latest available systems on these datasets.