Using LLM to Improve Knowledge Graph Entity Matching

Victor Eiti Yamamoto[1,2,*,†], Hideaki Takeda[1,2,†]

[1]National Insitute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
[2]The Graduate University for Advanced Studies, SOKENDAI, Shonan Village, Hayama, Kanagawa 240-0193 Japan

Abstract

Knowledge graphs (KGs) are powerful tools for representing and reasoning over structured information. Entity matching between KGs helps integrate multiple KGs. However, the performance of entity matching tools can be sensitive to parameter settings, such as thresholds. Large language models (LLMs) have emerged as powerful tools for solving reasoning problems and show potential for improving entity alignment. Our approach incorporates two LLM-based steps: filtering and expansion. In the filtering step, an LLM is used to validate entity mappings. The expansion step then uses an LLM to select the correct mapping from a candidate list for any source entity that lacks a corresponding pair after the filtering step. Experiments on the OAEI KG track dataset and matchings with DBpedia datasets show that using an LLM as a filter achieves a low false-negative rate and a favorable false-positive rate, indicating that it can improve precision without significantly lowering recall. However, the expansion step has low precision because the LLM tries to select a corresponding entity even when no correct match exists in the candidate list.

Keywords

Knowledge graph, entity matching, large language models

1. Introduction

Entity matching (EM), the task of identifying equivalent entities across Knowledge Graphs (KGs), is crucial for knowledge integration. Recent Large Language Model (LLM) based methods are often evaluated in unrealistic settings, using hit-rate metrics that neglect precision/recall trade-offs [1, 2]. While traditional statistical methods are competitive [3], they rely on manually tuned similarity thresholds, limiting their robustness.

To address these limitations, we propose a novel two-stage method using an LLM as both a qualitative filter and a correspondence expander. First, we replace the rigid statistical threshold of a baseline matcher by prompting an LLM to verify candidate pairs, creating a high-confidence alignment. Second, for remaining unmatched entities, the LLM selects the correct correspondence from a retrieved candidate list, expanding alignment coverage.

Our evaluation on OAEI [4] and Gollum [5] datasets shows that filtering effectively prunes incorrect candidates with minimal impact on recall, while expansion consistently achieves the highest F-measure against established baselines. These results demonstrate improved precision and recall. However, we note that LLMs tend to force a selection even when no correct match exists, highlighting a key area for future work.

2. Related Works

Existing ontology matching systems employ diverse strategies. For instance, PARIS [6] is a probabilistic matcher that computes equivalence probabilities for predicates and entities by leveraging the concept of functionality and comparing attribute values. LogMap [7], in contrast, uses an anchor-based approach

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
*Corresponding author.
†These authors contributed equally.
eitiyamamoto@gmail.com (V. E. Yamamoto); takeda@nii.ac.jp (H. Takeda)
https://github.com/eitiyamamoto (V. E. Yamamoto)
https://orcid.org/0000-0002-3825-6461 (V. E. Yamamoto); https://orcid.org/0000-0002-2909-7163 (H. Takeda)

that combines lexical indexing with an ontology reasoner to achieve high-precision mappings, though its effectiveness depends on a well-structured class ontology. Another approach, the Full Triple Matcher[8], first maps entities and predicates based on their labels, then uses these mappings to find compatible triples, which are subsequently used to refine the initial entity mappings.

3. Approach

Our approach refines entity alignments produced by an existing matcher, specifically leveraging the output of the Full Triple Matcher. The Full Triple Matcher takes two distinct KGs, G_1 and G_2, as input and outputs a set of aligned entities, M_e, and triples, M_t. We introduce a two-step post-processing pipeline to enhance these initial alignments: Filtering and Expansion.

3.1. Filtering

This step filters the initial set of candidate entity alignments, M_e, to identify pairs corresponding to the same real-world entity. For each source entity e_s within a pair in M_e, we first identify a set of top candidate target entities, denoted as $E_t^{candidate}$, based on the similarity score generated from FTM. Subsequently, each candidate pair, consisting of e_s and a target entity $e_t^{candidate} \in E_t^{candidate}$, is passed to an LLM for verification. To provide context, this input is augmented with the Top-10 most similar triples associated with both e_s and $e_t^{candidate}$ selected based on the scores in M_t. If the LLM confirms that the entities in the pair refer to the same real-world entity, the pair is added to a new filtered set, $M_e^{filtering}$. Otherwise, the pair is discarded.

3.2. Expansion

This step expands the matching set by recovering correct entity pairs that were erroneously removed during the filtering phase. First, we identify a set of source entities, $e_s^{removed}$, that exist in the initial set of pairs, M_e, but lack a corresponding match in the filtered set, $M_e^{filtering}$. For each removed entity $e_s^{removed}$, we retrieve the top-10 most similar candidate entities, $E_t^{candidate}$, from the target knowledge graph, based on the score obtained from FTM. These candidates are then passed to an LLM, which is prompted to perform a multiple-choice selection to identify the single best match that represents the same real-world object. The resulting pair $(e_s^{removed}, e_t)$ is then added to $M_e^{filtering}$ to form the final expanded matching set, $M_e^{matching}$.

4. Result

We compared our methods, Filter and Expansion, against several baselines: BaselineAltLabel, LogMap [7], PARIS [6], and the Full Triple Matcher (FTM) [8]. Our methods extend FTM by adding a filtering step (Filter) or both filtering and expansion (Expansion). We tuned baselines for optimal F-measure using a threshold, a step omitted by our "(w/o threshold)" variants. We used the Gemma 3 [9] with 27 billion parameters as the LLM model. We selected this model because it can be run locally with a single GPU, making it easier to replicate and use. The codes and results are available in the GitHub [1].

On the OAEI 2023 KG track using DBkWik datasets [10], our Expansion (w/o threshold) method achieved the highest recall on four datasets, surpassing LogMap which was highest on three in terms of precision. Furthermore, our standard Expansion method achieved the top F-measure on three datasets (Table 1).

On the large-scale Gollum dataset [5] (MAL/SWW-DBpedia), our Filter method achieved the highest precision, while our Expansion method yielded the best F-measure across both tasks (Table 2). In contrast, LogMap failed to load the data, and PARIS did not terminate on SWW-DBpedia.

[1] https://github.com/eitiyamamoto/llm-kg-matcher.git

Table 1
Result for entity matching dataset (OAEI)

dataset	matcher	Precision	Recall	F-measure	Threshold
SWW-SWG	BaselineAltLabel	0.92	0.62	0.74	0.0
	LogMap	**0.94**	0.69	**0.80**	0.0
	PARIS	0.64	0.67	0.66	0.98
	FTM	0.81	0.77	0.79	0.87
	Filter (w/o threshold)	0.72	0.75	0.74	-
	Filter	0.83	0.73	0.78	**0.84**
	Expansion (w/o threshold)	0.68	**0.81**	0.74	-
	Expansion	0.83	0.77	**0.80**	0.84
SWW-TOR	BaselineAltLabel	0.92	0.90	**0.91**	0.0
	LogMap	**0.94**	0.78	0.86	0.0
	PARIS	0.56	0.93	0.70	0.99
	FTM	0.85	0.90	0.87	0.90
	Filter (w/o threshold)	0.61	0.91	0.73	-
	Filter	0.87	0.87	0.87	0.90
	Expansion (w/o threshold)	0.61	**0.95**	0.74	-
	Expansion	0.87	0.89	0.88	0.90
MCU-MDB	BaselineAltLabel	0.86	0.68	0.76	0.0
	LogMap	0.84	0.46	0.60	0.00
	PARIS	**0.87**	**0.83**	**0.85**	0.60
	FTM	0.81	0.68	0.74	0.90
	Filter (w/o threshold)	0.74	0.68	0.71	-
	Filter	0.82	0.65	0.73	0.72
	Expansion (w/o threshold)	0.61	0.70	0.66	-
	Expansion	0.82	0.66	0.73	0.90
MAL-MBT	BaselineAltLabel	0.88	0.89	0.89	0.0
	LogMap	0.89	0.76	0.82	0.0
	PARIS	0.84	0.92	0.88	0.99
	FTM	0.89	0.89	0.89	0.94
	Filter (w/o threshold)	0.83	0.89	0.86	-
	Filter	**0.92**	0.86	0.89	0.90
	Expansion (w/o threshold)	0.82	**0.93**	0.87	-
	Expansion	**0.92**	0.89	**0.90**	0.90
MAL-STX	BaselineAltLabel	**0.88**	0.93	**0.90**	0.0
	LogMap	**0.88**	0.77	0.82	0.91
	PARIS	0.67	0.93	0.78	0.99
	FTM	0.84	0.92	0.88	0.90
	Filter (w/o threshold)	0.63	0.93	0.75	-
	Filter	0.87	0.91	0.89	0.89
	Expansion (w/o threshold)	0.63	**0.94**	0.75	-
	Expansion	0.87	0.92	**0.90**	**0.90**

On the MCU-MDB dataset, we analyzed our method's performance using confusion matrices (Table 3). The filtering step achieves a low false-negative rate, correctly pruning non-matches while passing most correct pairs to the next step (Table 3a). The expansion step, which relies on a top-10 candidate list for the LLM, achieves a precision of 0.13. In the table 3b, we have a confusion matrix with correct@n to indicate if the correct answer is present in the n-th position. The LLM correctly identifies the entity in 49% of cases when it is in the top-10. This accuracy rises to 60% if the correct entity is ranked first, but drops to 32% for lower-ranked (2-10) entities.

Figure 1 illustrates the comparison between the set of results obtained from the FTM method and the four new methods, "Filtered" and "Expansion", with and without a selection threshold. The Venn diagrams demonstrate a substantial overlap in all four scenarios, indicating a strong agreement between the FTM method and the alternative approaches. Specifically, the application of a threshold to the "Filtered" and "Expansion" methods significantly reduces their number of unique results, making them almost entirely subsets of the FTM results. This high degree of similarity is quantitatively confirmed by the Jaccard coefficient, which was calculated for each comparison and found to vary from 0.91 to 0.98,

Table 2
Result for entity matching dataset (Gollum)

dataset	matcher	Precision	Recall	F-measure	Threshold
MAL-DBpedia	BaselineAltLabel	0.53	0.69	0.60	0.0
	LogMap	-	-	-	-
	PARIS	0.50	0.53	0.51	0.95
	FTM	0.39	**0.73**	0.50	0.90
	Filter (w/o threshold)	**0.74**	0.66	0.70	-
	Filter	**0.74**	0.66	0.70	0.0
	Expansion (w/o threshold)	0.73	0.72	0.72	-
	Expansion	**0.74**	0.72	**0.73**	-
SWW-DBpedia	BaselineAltLabel	0.48	**0.78**	0.59	0.0
	LogMap	-	-	-	-
	PARIS	-	-	-	-
	FTM	0.72	0.71	0.71	0.94
	Filter (w/o threshold)	0.54	0.71	0.62	-
	Filter	**0.91**	0.71	**0.79**	0.90
	Expansion (w/o threshold)	0.54	0.71	0.62	-
	Expansion	**0.91**	0.71	**0.79**	0.90

Table 3
Confusion matrix from MCU-MDB dataset

	Generated answer			
Correct	**Not Tested**	**TRUE**	**FALSE**	**Total**
TRUE	16	1,120	57	1,193
FALSE		393	1,854	2,247
Total	**16**	**1,513**	**1,911**	**3,440**

(a) Filtering

		Correct		Total
Correct@10	**Correct@1**	**TRUE**	**FALSE**	
TRUE		**46**	**47**	**93**
	TRUE	34	22	56
	FALSE	12	25	37
FALSE			**294**	**294**
	FALSE		294	294
Grand Total		**46**	**341**	**387**

(b) Expansion

signifying a very strong correlation between the result sets.

5. Discussion

The FTM with filter and expansion outperformed the baselines because the filter improves precision while the expansion considers non-top-1 candidates to improve recall. However, Table 3b reveals that the expansion step's performance is unexpectedly low. This is because the LLM often fails to select the correct pair from the candidate list and forces a choice even when the correct match is absent. Consequently, the method's effectiveness still relies heavily on a well-tuned similarity threshold to filter out incorrect pairs.

The filter step alone, however, presents a viable alternative to a threshold. As shown in Tables 1 and 2, it achieves comparable F-measure on most datasets, making it advantageous when a labeled dataset for tuning a threshold is unavailable. According to Table 3a, the filter produces few false negatives, thus preserving recall. Its primary limitation is a high false-positive rate, which restricts precision gains. In summary, the filter is highly reliable when rejecting a pair (predicting 'false') but less reliable when accepting one (predicting 'true').

6. Conclusion

In this work, we proposed a two-step (filter and expansion) extension to a statistical method using an LLM. Our approach achieved the best results in most cases when combining both steps with threshold

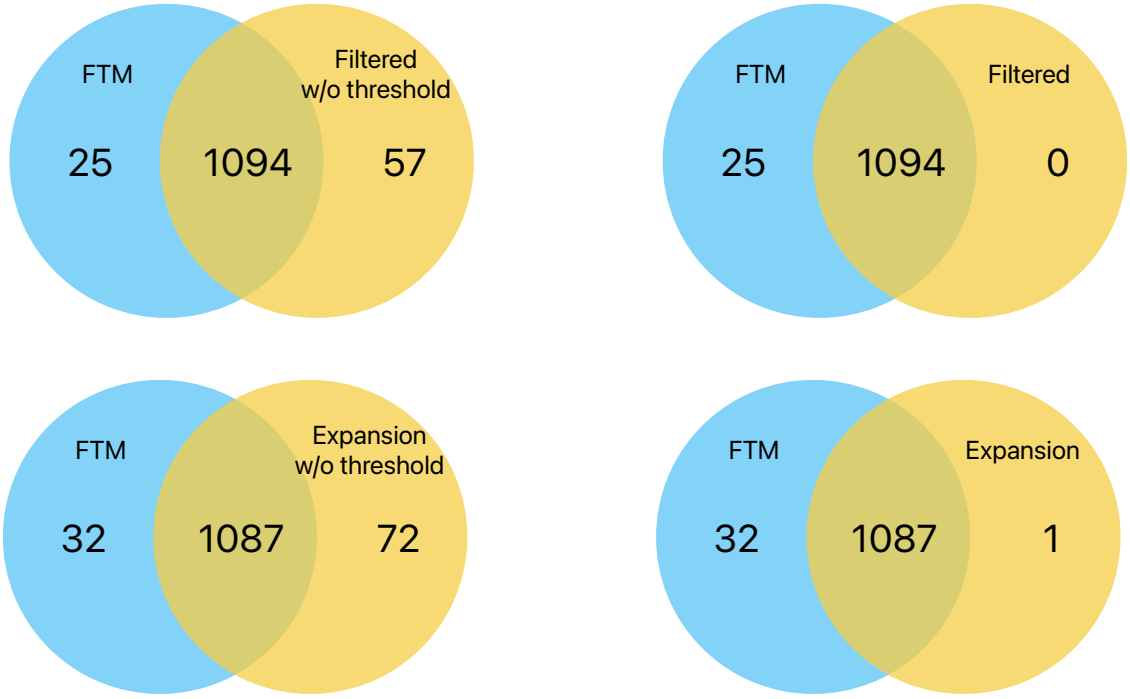

Figure 1: Comparison of the FTM, "Filtered" (top) and "Expansion" (bottom) methods for MCU-MDB dataset

filtering. We also found that the filter step alone yields comparable results, suggesting it can replace threshold filtering when an optimal threshold is unknown. However, the filter step maintains high recall at the cost of lower precision. On the other hand, the expansion step helps find pairs beyond the top-1 candidate, but its tendency to always select a pair—even when none is correct—results in low precision. As future work, we will refine these steps to learn from the selected pairs, test how the choice of LLM impacts performance and how to reduce the use of LLM to reduce computation effort, such as finding corner cases and entity pairs with close similarities.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini and Grammarly in order to: paraphrase and reword, improve writing style, and grammar and spelling checking. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] L. Yang, H. Chen, X. Wang, J. Yang, F.-Y. Wang, H. Liu, Two heads are better than one: Integrating knowledge from knowledge graphs and large language models for entity alignment, arXiv preprint arXiv:2401.16960 (2024).

[2] X. Chen, T. Lu, Z. Wang, Llm-align: Utilizing large language models for entity alignment in knowledge graphs, arXiv preprint arXiv:2412.04690 (2024).

[3] M. Leone, S. Huber, A. Arora, A. García-Durán, R. West, A critical re-evaluation of neural methods for entity alignment, Proc. VLDB Endow. 15 (2022) 1712–1725. URL: https://doi.org/10.14778/3529337.3529355. doi:10.14778/3529337.3529355.

[4] S. Hertling, H. Paulheim, The knowledge graph track at oaei: Gold standards, baselines, and the

golden hammer bias, in: The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17, Springer, 2020, pp. 343–359.
[5] S. Hertling, H. Paulheim, Gollum: A gold standard for large scale multi source knowledge graph matching, arXiv preprint arXiv:2209.07479 (2022).
[6] F. M. Suchanek, S. Abiteboul, P. Senellart, Paris: Probabilistic alignment of relations, instances, and schema, Proc. VLDB Endow. 5 (2011) 157–168. URL: https://doi.org/10.14778/2078331.2078332. doi:10.14778/2078331.2078332.
[7] E. Jiménez-Ruiz, B. Cuenca Grau, Logmap: Logic-based and scalable ontology matching, in: The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I 10, Springer, 2011, pp. 273–288.
[8] V. E. Yamamoto, H. Takeda, Full triple matcher: Integrating all triple elements between heterogeneous knowledge graphs, ACM Trans. Web (2025). URL: https://doi.org/10.1145/3754338. doi:10.1145/3754338, just Accepted.
[9] A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, L. Rouillard, et al., Gemma 3 technical report, CoRR (2025).
[10] S. Hertling, H. Paulheim, Dbkwik: extracting and integrating knowledge from thousands of wikis, Knowledge and Information Systems 62 (2020) 2169–2190.